



Xpert.press

Stephan Verclas  
Claudia Linnhoff-Popien (Hrsg.)

# Smart Mobile Apps

Mit Business-Apps ins Zeitalter  
mobiler Geschäftsprozesse



Springer

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals  
in den Bereichen Softwareentwicklung,  
Internettechnologie und IT-Management aktuell  
und kompetent relevantes Fachwissen über  
Technologien und Produkte zur Entwicklung  
und Anwendung moderner Informationstechnologien.

Stephan Verclas · Claudia Linnhoff-Popien

# Smart Mobile Apps

Mit Business-Apps ins Zeitalter  
mobiler Geschäftsprozesse



Dr. Stephan Verclas  
T-Systems International GmbH  
Innovation Center München  
Elisabeth-Selbert-Str. 1  
80939 München  
Germany  
*stephan.verclas@t-systems.com*

Prof. Dr. Claudia Linnhoff-Popien  
Lehrstuhl für Mobile und Verteilte Systeme  
Ludwig-Maximilians-Universität München  
Oettingenstr. 67  
80538 München  
Germany  
*linnhoff@ifi.lmu.de*

ISSN 1439-5428  
ISBN 978-3-642-22258-0 e-ISBN 978-3-642-22259-7  
DOI 10.1007/978-3-642-22259-7  
Springer Heidelberg Dordrecht London New York

Bibliografische Information der Deutschen Nationalbibliothek  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;  
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer-Verlag Berlin Heidelberg 2012  
Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

*Einbandgestaltung:* KünkelLopka GmbH, Heidelberg

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media ([www.springer.de](http://www.springer.de))

---

# Vorwort

Die Informations- und Kommunikationstechnik verändert die Gesellschaft immer wieder aufs Neue. Haben vor ein paar Jahren noch einfache Handys das Bild geprägt, sind es jetzt Smartphones und andere mobile Endgeräte. Jeder zweite Mensch auf der Welt hat heute bereits eines. Die Anwender können damit nicht nur telefonieren, sondern auch E-Mails senden, Fotos und Filme machen, ins Internet gehen, soziale Medien nutzen oder navigieren. Es ist eine neue digitale Welt der intelligenten Vernetzung entstanden. Die Inhalte kommen aus der Cloud, und der Datenzugriff ist von überall möglich.

Mobile Apps sind Softwareprogramme, die auf mobilen Endgeräten wie Smartphones oder Tablets ausgeführt werden. Sie haben meist nur wenige Funktionen, damit die Anwender diese intuitiv, schnell und umfassend bedienen kann. Ein weiterer Vorteil: Smartphone sind immer sofort betriebsbereit, ein langwieriges Booten wie beim PC oder Notebook gibt es nicht.

Das vorliegende Buch „Smart Mobile Apps“ beleuchtet diese Entwicklung aus drei Perspektiven – aus der Anwendersicht, aus der technologischen Sicht und aus der marktwirtschaftlichen Sicht:

**Teil I: Einführung**

**Teil II–IV: Anwendersicht**

Business-Apps für Endkunden

Business-Apps für Mitarbeiter

Herausforderungen der Anwendungsbranchen

**Teil V–VIII: Technologische Sicht**

Sicherheit, Datenschutz & Privacy

Kontext, Navigation & Personalisierung

Workflow & Prozessintegration

Plattformen & Betriebssysteme

**Teil IX–X: Marktwirtschaftliche Sicht**

Geschäftsmodelle & Marktforschung

Wandel in der Gesellschaft

Dies soll eine ganzheitliche Betrachtung des noch jungen Themas sicherstellen. Die Beiträge stammen von 116 Experten aus Wissenschaft, Wirtschaft und Verbänden, und sind jeweils von bis zu fünf Experten begutachtet worden, bevor eine Aufnahme in das Buch erfolgte.

Ausgangspunkt für das Werk ist eine Kooperation zwischen dem T-Systems Innovation Center München und dem Lehrstuhl für Mobile und Verteilte Systeme der Ludwig-Maximilians-Universität München. Diese hat das Ziel, das Entstehen und Gestalten von Innovationen auf dem Gebiet der Mobile Business Applications zu erforschen. Kunden, Partner und Mitstreiter haben dazu beigetragen, dass die Kooperation heute sowohl national als auch international Beachtung findet. Dieses Know-how wurde in dem Buch dokumentiert, und es spiegelt den aktuellen Stand von Wissenschaft, Wirtschaft und Technik in Deutschland wider.

Die zentrale Botschaft des vorliegenden Handbuchs lautet: Aktuell sind Consumer Apps bereits äußerst erfolgreich, Business Apps haben sich hingegen noch nicht durchgesetzt. Hierzu fehlen derzeit noch Konzepte, wie diese sich erfolgreich am Markt etablieren können. Auch wenn die Produktivitätsvorteile von Business Apps offensichtlich sind, hemmen die technischen und wirtschaftlichen Hürden eine größere Verbreitung über die derzeitigen Insellösungen oder den Testbetrieb. Dieses Dilemma gilt es künftig zu überwinden. Dazu liefert das Buch erste Ansätze. In diesem Sinne wünschen wir viel Spaß und Erfolg bei der Weiterentwicklung von Smart Mobile Apps.

Stephan Verclas, Claudia Linnhoff-Popien

---

# Inhaltsverzeichnis

## Teil I Einführung

- 1 Mit Business-Apps ins Zeitalter mobiler Geschäftsprozesse ..... 3  
C. Linnhoff-Popien, S. Verclas

## Teil II Anwendersicht: Business-Apps für Endkunden

- 2 Zum Einfluss mobiler Endgeräte  
auf das individuelle Gesundheitsverhalten von PKV-Versicherten ... 19  
T. Nguyen, J. Lindenmeier, T. Fritsch, F. Kerssenfischer
- 3 Smart Automotive Apps für kundenorientierte Dienstleistungen  
im Fahrzeug ..... 27  
T. Schlachtbauer, G. Stanzl, M. Pühler, M. Schermann, H. Krcmar
- 4 Smart Apps in einem vernetzten (auto)mobilen Umfeld:  
IT-Security und Privacy ..... 43  
B. Weyl, M. Graf, A. Bouard

## Teil III Anwendersicht: Business-Apps für Mitarbeiter

- 5 Mobile Apps für industrielle Anwendungen  
am Beispiel von Siemens ..... 61  
A. Zeidler, R. Eckl, W. Trumler, M. Franz
- 6 Neue Wege in das Internet der Dinge durch Smart Mobile Apps ..... 81  
M. Wagner, P. Holleis, H. Berndt
- 7 Eine neue Generation von Geschäftsanwendungen ..... 95  
M. Kern

<b>8 Herausforderungen bei der Mobilisierung von Business Applikationen und erste Lösungsansätze</b>	107
M. Euler, M. Hacke, C. Hartherz, S. Steiner, S. Verclas	

#### **Teil IV Anwendersicht: Herausforderungen der Anwendungsbranche**

<b>9 Anwendungssicht mobiler Geschäftsanwendungen</b>	125
L. Kuassi, M. Bischel	
<b>10 Der Browser als mobile Plattform der Zukunft – Die Möglichkeiten von HTML5-Apps</b>	147
K. Albert, M. Stiller	
<b>11 Die Grenzen des Browsers durchbrechen</b>	161
A.R.S. Gerlicher	
<b>12 Technologische und marktseitige Unsicherheit bei der Neuentwicklung von Mobile Enterprise Services</b>	177
F. Weiß, M. Söllner	

#### **Teil V Technologische Sicht: Sicherheit, Datenschutz & Privacy**

<b>13 Smart Mobile Apps: Enabler oder Risiko?</b>	193
C. Eckert, C. Schneider	
<b>14 Das Smartphone als sichere Burg</b>	209
P. Wildt, R. Meister	
<b>15 Hochsichere Smartphones mit L4Android</b>	225
S. Liebergeld, M. Lange, A. Lackorzynski, A. Warg	
<b>16 Zertifizierte Apps: mehr Funktionalität, Sicherheit und Bedienungsfreundlichkeit</b>	241
W.-R. Heidemann, I. Zumbruch	
<b>17 Mobile Apps in Enterprise-Anwendungen unter Berücksichtigung von Sicherheitsaspekten</b>	253
J. Bertram, C. Kleiner	
<b>18 Ein interoperables Bezahlverfahren für In-App- und Online-Käufe</b>	269
T. Goeller, T. Hin Yeap	

#### **Teil VI Technologische Sicht: Kontext, Navigation & Personalisierung**

<b>19 Technologische Herausforderungen für kontextsensitive Geschäftsanwendungen</b>	285
M. Werner, M. Kessel, F. Gschwandtner, M. Dürr, K. Wiesner, T. Mair	

---

<b>20</b>	<b>Integration von Kontextinformationen in Smart Applications und Smart Workflows</b>	301
	D. Bade, W. Lamersdorf	
<b>21</b>	<b>Die Anwendung denkt mit auf Schritt und Tritt</b>	319
	K. Geihs	

## Teil VII Technologische Sicht: Workflow & Prozessintegration

<b>22</b>	<b>Agile Geschäftsprozesse durch integrierte mobile Kommunikation</b>	333
	J. Totzke, K. Klug	
<b>23</b>	<b>Der Cloud-Broker: dynamische Orchestrierung von Cloud-Diensten zu Smart Mobile Apps</b>	351
	V.A. Danciu, N. gentschen Felde, D. Kranzlmüller, M. Schiffers, J.R. Watzl	
<b>24</b>	<b>Smart Apps aus der Wolke</b>	369
	H. Sirtl, F. Koch	
<b>25</b>	<b>Business Web: Cloud-basierte Flexibilisierung und Mobilisierung von Geschäftsprozessen</b>	385
	U. Kubach, R. Ackermann, M. Ameling, V. Lotz, D. Oberle, V.A. Schmidt	

## Teil VIII Technologische Sicht: Plattformen & Betriebssysteme

<b>26</b>	<b>Architekturen mobiler Multiplattform-Apps</b>	403
	F. Willnecker, D. Ismailović, W. Maisan	
<b>27</b>	<b>Programmierung von Smart Mobile Apps</b>	419
	T. Strang, M. Lichtenstern	
<b>28</b>	<b>Eine Dienstplattform zur Entwicklung mobiler sozialer Anwendungen auf Basis von XMPP</b>	431
	R. Lübke, D. Schuster, T. Springer, N. Jansen, A. Schill	
<b>29</b>	<b>Mobile kollaborative Apps</b>	447
	J. Gäbler, R. Klauck, R. Kopsch, F. Liu, M. Pink, S. Schöpke, H. König	
<b>30</b>	<b>Entwicklung mobiler Betriebssysteme im Lichte neuer Apps, veränderter Herausforderungen und der Virtualisierung</b>	465
	U. Baumgarten, A. Bernhofer, R. Dörfl	
<b>31</b>	<b>Mit Java ins Internet der Dinge aufbrechen</b>	475
	R. Eschrich, G. Burger	

**Teil IX Marktwirtschaftliche Sicht: Geschäftsmodelle & Marktforschung**

- 32 Plattformoffenheit – Implikationen für Anbieter von Mobile Business Apps . . . . .** 495  
D. Hilkert, C. Burkard, T. Widjaja, T. Hess, P. Buxmann
- 33 Smarter Apps – Motor für Geschäftsmodellinnovationen . . . . .** 507  
T. Goetz, N. Feldmann, S. Schmidt

**Teil X Marktwirtschaftliche Sicht: Wandel in der Gesellschaft**

- 34 Mobile Kommunikation der nächsten Dekade . . . . .** 521  
A. Picot, C. Janello
- 35 Mobil – interaktiv – sozial: Der digitale Mensch von morgen – „always in touch“? . . . . .** 531  
M. Wolf, K. Dirtheuer, S. Sagl, G. Herrmann
- 36 Die mobile „Generation plus“ – Anforderungen und Potenziale . . . . .** 545  
H. Klaus, M.C. Balasch, C. Nedopil, S. Glende, M. Gerneth
- 37 Smart Mobile Apps in der Lehre . . . . .** 559  
M. Wagner, B. Brügge
- 38 Paul und Paula oder der *Spramokom®* . . . . .** 571  
W. von Reden

---

# **Autoren**

Ralf Ackermann  
SAP AG, Walldorf, E-mail: ralf.ackermann@sap.com

Kristin Albert  
Fraunhofer ESK, München, E-mail: kristin.albert@esk.fraunhofer.de

Michael Ameling  
SAP AG, Walldorf, E-mail: michael.ameling@sap.com

Dirk Bade  
Universität Hamburg, Verteilte Systeme und Informationssysteme,  
E-mail: bade@informatik.uni-hamburg.de

Michael C. Balasch  
Deutsche Telekom AG, Laboratories, Berlin,  
E-mail: Michael.Balasch@telekom.de

Uwe Baumgarten  
Technische Universität München, Fakultät für Informatik,  
E-mail: baumgaru@in.tum.de

Hendrik Berndt  
DOCOMO Communications Laboratories Europe GmbH, München,  
E-mail: berndt@docomolab-euro.com

Andreas Bernhofer  
Technische Universität München, Fakultät für Informatik,  
E-mail: andreas.bernhofer@tum.de

Jens Bertram  
Fachhochschule Hannover, Fakultät IV, Abt. Informatik,  
E-mail: jens.bertram1@stud.fh-hannover.de

Michael Bischel  
LeoSoft GmbH, Haar b. München, E-mail: info@leosoftsys.com

Alexandre Bouard  
BMW Forschung und Technik GmbH, München,  
E-mail: alexandre.bouard@bmw.de

Bernd Brügge  
Technische Universität München, Lehrstuhl für Angewandte Softwaretechnik,  
Institut für Informatik, E-mail: bruegge@in.tum.de

Guido Burger  
Oracle, Stuttgart, E-mail: guido.burger@oracle.com

Christoph Burkard  
Technische Universität Darmstadt, Fachgebiet Wirtschaftsinformatik | Software  
Business & Information Management, E-mail: burkard@is.tu-darmstadt.de

Peter Buxmann  
Technische Universität Darmstadt, Fachgebiet Wirtschaftsinformatik | Software  
Business & Information Management, E-mail: buxmann@is.tu-darmstadt.de

Vitalian A. Danciu  
Münchner Netz-Management Team (MNM-Team), Ludwig-Maximilians-  
Universität München, E-mail: danciu@mnm-team.org

Kerstin Dirtheuer  
TNS Infratest GmbH, München, E-mail: kerstin.dirtheuer@tns-infratest.com

Robert Dörfel  
Technische Universität München, Fakultät für Informatik,  
E-mail: doerfel@in.tum.de

Michael Dürr  
Ludwig-Maximilians-Universität München, Institut für Informatik,  
E-mail: michael.duerr@ifi.lmu.de

Claudia Eckert  
Fraunhofer AISEC, Garching bei München,  
E-mail: claudia.eckert@aisec.fraunhofer.de  
Technische Universität München, Lehrstuhl für Sicherheit in der Informatik,  
Garching bei München, E-mail: eckert@sec.in.tum.de

Roland Eckl  
Siemens AG, Corporate Research and Technologies, München,  
E-mail: eckl.roland@siemens.com

Rainer Eschrich  
Oracle Inc., Java Embedded Global Sales Unit, München,  
E-mail: rainer.eschrich@oracle.com

Matthias Euler  
Deutsche Telekom AG, Darmstadt, E-mail: meuler@telekom.de

Nils gentschen Felde

Münchner Netz-Management Team (MNM-Team), Ludwig-Maximilians-Universität München, E-mail: felde@mnm-team.org

Niels Feldmann

IBM Deutschland GmbH, Frankfurt, E-mail: Niels.Feldmann@de.ibm.com

Marquart Franz

Siemens AG, Corporate Research and Technologies, München,  
E-mail: marquart.franz@siemens.com

Tobias Fritsch

Allianz Managed Operations & Services SE, Unterföhring,  
E-mail: t.fritsch@allianz.de

Jan Gäbler

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze und Kommunikationssysteme, E-mail: jgaebler@informatik.tu-cottbus.de

Kurt Geihs

Universität Kassel, Fachbereich Elektrotechnik/Informatik,  
E-mail: geihs@uni-kassel.de

Ansgar R. S. Gerlicher

Hochschule der Medien Stuttgart, Studiengang Mobile Medien, Professur Mobile Applications, E-mail: gerlicher@hdm-stuttgart.de

Marlene Gerneth

Deutsche Telekom AG, Laboratories, Berlin,  
E-mail: Marlene.Gerneth@telekom.de

Sebastian Glende

YOUSE GmbH, Berlin, E-mail: Sebastian.Glende@telekom.de

Toni Goeller

Toposis Corp., Ottawa, Ontario, Canada, E-mail: toni.goeller@toposis.com

Thomas Goetz

IBM Deutschland GmbH, Köln, E-mail: Thomas.goetz@de.ibm.com

Maximilian Graf

BMW Forschung und Technik GmbH, München,  
E-mail: maximilian.graf@bmw.de

Florian Gschwandtner

Ludwig-Maximilians-Universität München, Institut für Informatik,  
E-mail: florian.gschwandtner@ifi.lmu.de

Marcus Hacke

T-Systems International GmbH, Darmstadt, E-mail: marcus.hacke@t-systems.com

Claus Hartherz

T-Systems International GmbH, Bonn, E-mail: claus.hartherz@t-systems.com

Wolf-Rüdiger Heidemann

TÜV SÜD Product Service GmbH, Software-Qualität und Escrow-Services,  
München, E-mail: wolf.heidemann@tuev-sued.de

Gabriele Herrmann

TNS Infratest GmbH, Hamburg, E-mail: gabriele.herrmann@tns-infratest.com

Thomas Hess

Ludwig-Maximilians-Universität München, Institut für Wirtschaftsinformatik und  
Neue Medien, E-mail: thess@bwl.lmu.de

Daniel Hilkert

Ludwig-Maximilians-Universität München, Institut für Wirtschaftsinformatik und  
Neue Medien, E-mail: hilkert@bwl.lmu.de

Paul Holleis

DOCOMO Communications Laboratories Europe GmbH, München,  
E-mail: holleis@docomolab-euro.com

Damir Ismailović

Technische Universität München, Institut für Informatik, Lehrstuhl für Angewandte  
Softwaretechnik, E-mail: damir.ismailovic@in.tum.de

Christoph Janello

Ludwig-Maximilians-Universität München, Institut für Information, Organisation  
und Management, E-mail: janello@lmu.de

Nikolas Jansen

Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: nikojansen@gmail.com

Michael Kern

sovanta AG, X-House, Heidelberg,  
E-mail: michael.kern@sovanta.com

Frederik Kerssenfischer

Allianz Managed Operations & Services SE, Unterföhring,  
E-mail: frederik.kerssenfischer@allianz.de

Moritz Kessel

Ludwig-Maximilians-Universität München, Institut für Informatik,  
E-mail: moritz.kessel@ifi.lmu.de

Ronny Klauck

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze und  
Kommunikationssysteme, E-mail: rk@informatik.tu-cottbus.de

Harald Klaus

Deutsche Telekom AG, Laboratories, Berlin, E-mail: Harald.Klaus@telekom.de

**Carsten Kleiner**

Fachhochschule Hannover, Fakultät IV, Abt. Informatik, E-mail: ckleiner@acm.org

**Karl Klug**

Siemens Enterprise Communications GmbH & Co. KG, München,  
E-mail: karl.klug@siemens-enterprise.com

**Frank Koch**

Microsoft Deutschland GmbH, Unterschleißheim,  
E-mail: frankoch@microsoft.com

**Hartmut König**

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze und  
Kommunikationssysteme, E-mail: koenig@informatik.tu-cottbus.de

**Ralf Kopsch**

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze und  
Kommunikationssysteme, E-mail: rkopsch@informatik.tu-cottbus.de

**Dieter Kranzlmüller**

Münchner Netz-Management Team (MNM-Team), Ludwig-Maximilians-  
Universität München, E-mail: kranzlm@mnm-team.org

**Helmut Krcmar**

Technische Universität München, Lehrstuhl für Wirtschaftsinformatik, Garching b.  
München, E-mail: krcmar@in.tum.de

**Léopold Kuassi**

LeoSoft GmbH, Haar b. München, E-mail: info@leosoftsys.com

**Uwe Kubach**

SAP AG, Walldorf, E-mail: uwe.kubach@sap.com

**Adam Lackorzynski**

Technische Universität Dresden, Professur für Betriebssystem,  
E-mail: adam@os.inf.tu-dresden.de

**Winfried Lamersdorf**

Universität Hamburg, Verteilte Systeme und Informationssysteme,  
E-mail: lamersd@informatik.uni-hamburg.de

**Matthias Lange**

Technische Universität Berlin und Deutsche Telekom Laboratories, Security in  
Telecommunications, Berlin, E-mail: mlange@sec.t-labs.tu-berlin.de

**Michael Lichtenstern**

Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Kommunikation  
und Navigation, Wessling/Oberpfaffenhofen, E-mail: m.lichtenstein@dlr.de

**Steffen Liebergeld**

Technische Universität Berlin und Deutsche Telekom Laboratories, Security in  
Telecommunications, Berlin, E-mail: steffen@sec.t-labs.tu-berlin.de

**Jörg Lindenmeier**  
WHL Wissenschaftliche Hochschule Lahr, E-mail: joerg.lindenmeier@whl-lahr.de

**Claudia Linnhoff-Popien**  
Ludwig-Maximilians-Universität München, Institut für Informatik,  
E-mail: linnhoff@ifi.lmu.de

**Fuwen Liu**  
Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze und  
Kommunikationssysteme, E-mail: lfw@informatik.tu-cottbus.de

**Volkmar Lotz**  
SAP Labs France SAS, Sophia-Antipolis, Frankreich,  
E-mail: volkmar.lotz@sap.com

**Robert Lübke**  
Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: robert.luebke@tu-dresden.de

**Thomas Mair**  
Ludwig-Maximilians-Universität München, Institut für Informatik,  
E-mail: thomas.mair@ifi.lmu.de

**Wolfgang Maison**  
Weptun GmbH München, E-mail: w.maison@weptun.de

**Ralf Meister**  
GeNUA mbH, Kirchheim bei München, E-mail: Ralf\_Meister@genua.de

**Christoph Nedopil**  
YOUSE GmbH, Berlin, E-mail: Christoph.Nedopil@youse.de

**Tristan Nguyen**  
WHL Wissenschaftliche Hochschule Lahr, E-mail: tristan.nguyen@whl-lahr.de

**Daniel Oberle**  
SAP AG, Walldorf, E-mail: d.oberle@sap.com

**Arnold Picot**  
Ludwig-Maximilians-Universität München, Institut für Information, Organisation  
und Management, E-mail: picot@lmu.de

**Mario Pink**  
Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze und  
Kommunikationssysteme, E-mail: pink@informatik.tu-cottbus.de

**Maximilian Pühler**  
Audi AG, Audi IT Open Innovation, Ingolstadt,  
E-mail: Maximilian.Puehler@audi.de

Wolf von Reden

Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut, Berlin,  
E-mail: wolf.von.reden@hhi.fraunhofer.de

Stefanie Sagl

TNS Infratest GmbH, München, E-mail: stefanie.sagl@tns-infratest.com

Michael Schermann

Technische Universität München, Lehrstuhl für Wirtschaftsinformatik, Garching b.  
München

Michael Schiffers

Münchener Netz-Management Team (MNM-Team), Ludwig-Maximilians-  
Universität München, E-mail: schiffer@mnm-team.org

Alexander Schill

Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: alexander.schill@tu-dresden.de

Tobias Schlachtbauer

Technische Universität München, Lehrstuhl für Wirtschaftsinformatik, Garching b.  
München, E-mail: Tobias.Schlachtbauer@in.tum.de

Sebastian Schmidt

IBM Deutschland GmbH, München, E-mail: Sebastian.Schmidt@de.ibm.com

Vasco Alexander Schmidt

SAP AG, Walldorf, E-mail: vasco.alexander.schmidt@sap.com

Christian Schneider

TU München, Fakultät für Informatik, Lehrstuhl für Sicherheit in der Informatik,  
Garching bei München, E-mail: chrschn@sec.in.tum.de

Sebastian Schöpke

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze und  
Kommunikationssysteme, E-mail: seb@informatik.tu-cottbus.de

Daniel Schuster

Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: daniel.schuster@tu-dresden.de

Holger Sirtl

Microsoft Deutschland GmbH, Unterschleißheim,  
E-mail: holger.sirtl@microsoft.com

Matthias Söllner

Universität Kassel, Fachgebiet Wirtschaftsinformatik,  
E-mail: soellner@uni-kassel.de

Thomas Springer

Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: thomas.springer@tu-dresden.de

**Gerhard Stanzl**

Audi AG, Geschäftsentwicklung IT, Ingolstadt, E-mail: Gerhard.Stanzl@audi.de

**Sascha Steiner**

T-Systems International GmbH, Darmstadt, E-mail: sascha.steiner@t-systems.com

**Michael Stiller**

Fraunhofer ESK, München, E-mail: michael.stiller@esk.fraunhofer.de

**Thomas Strang**

Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Kommunikation und Navigation, Wessling/Oberpfaffenhofen, E-mail: thomas.strang@dlr.de

**Jürgen Totzke**

Siemens Enterprise Communications GmbH & Co. KG, München,

E-mail: juergen.totzke@siemens-enterprise.com

**Wolfgang Trumler**

Siemens AG, Corporate Research and Technologies, Erlangen,

E-mail: wolfgang.trumler@siemens.com

**Stephan Verclas**

T-Systems International GmbH, München,

E-mail: stephan.verclas@t-systems.com

**Martin Wagner**

Technische Universität München, Lehrstuhl für Angewandte Softwaretechnik,

Institut für Informatik, E-mail: wagmarti@in.tum.de

**Matthias Wagner**

DOCOMO Communications Laboratories Europe GmbH, München,

E-mail: wagner@docomolab-euro.com

**Alexander Warg**

Technische Universität Dresden, Professur für Betriebssystem,

E-mail: warg@os.inf.tu-dresden.de

**Johannes R. Watzl**

Münchner Netz-Management Team (MNM-Team), Ludwig-Maximilians-

Universität München, E-mail: watzl@mnm-team.org

**Frank Weiß**

Universität Kassel/Detecon International, München, E-mail: weiss@wi-kassel.de

**Martin Werner**

Ludwig-Maximilians-Universität München, Institut für Informatik,

E-mail: martin.werner@ifi.lmu.de

**Benjamin Weyl**

BMW Forschung und Technik GmbH, München, E-mail: benjamin.weyl@bmw.de

Thomas Widjaja

Technische Universität Darmstadt, Fachgebiet Wirtschaftsinformatik | Software  
Business & Information Management, E-mail: widjaja@is.tu-darmstadt.de

Kevin Wiesner

Ludwig-Maximilians-Universität München, Institut für Informatik,  
E-mail: kevin.wiesner@ifi.lmu.de

Patrick Wildt

GeNUA mbH, Kirchheim bei München, E-mail: Patrick\_Wildt@genua.de

Felix Willnecker

Weptun GmbH München, Technische Universität München, Institut für Informatik,  
Lehrstuhl für Angewandte Softwaretechnik, E-mail: f.willnecker@weptun.de

Malthe Wolf

TNS Infratest GmbH, München, E-mail: malthe.wolf@tns-infratest.com

Tet Hin Yeap

University of Ottawa, School of Electrical Engineering and Computer Science,  
Ottawa, Ontario, Canada, E-mail: yeap@site.uottawa.ca

Andreas Zeidler

Siemens AG, Corporate Research and Technologies, München,  
E-mail: a.zeidler@siemens.com

Ina Zumbruch

TÜV SÜD Product Service GmbH, Software-Qualität und Escrow-Services,  
München, E-mail: ina.zumbruch@tuev-sued.de

# **Teil I**

## **Einführung**

---

# Kapitel 1

## Mit Business-Apps ins Zeitalter mobiler Geschäftsprozesse

Claudia Linnhoff-Popien und Stephan Verclas

**Zusammenfassung** Mobile Apps sind Programme, die auf modernen Handys, so genannten Smartphones, ausgeführt werden. Wurden nach einer Studie des Analytikhauses Gartner [1] im Jahr 2010 schon 8 Milliarden Apps auf die Smartphones aus dem Netz geladen, so steht der eigentliche Boom noch bevor: Bis Ende 2014 sollen 185 Milliarden App-Downloads erfolgen. Allerdings adressieren die derzeitigen Apps fast ausschließlich den Nutzer als Konsumenten, während es bisher wenige erfolgreiche Apps im Business-Bereich gibt, was u. a. am Fehlen eines schlüssigen App-Konzeptes im Business-Markt liegt.

Hier liegen ungeahnte Möglichkeiten. Mit Business-Apps können Unternehmen schnell und unkompliziert ihre Geschäftsprozesse mobilisieren. Leider gibt es noch ein paar Herausforderungen wie z. B. die fehlende Interoperabilität der verschiedenen mobilen Plattformen. Diese gilt es anzugehen, und Chancen dafür gibt es viele: Smartphones, Pads und Co. müssen sich nur in unser Denken integrieren. So können Business-Apps Situationen intuitiv begegnen – am Flughafen nach dem Check-in, wenn noch eine Auslandsversicherung benötigt wird, oder wenn direkt an einem Unfallort der Schaden aufgenommen werden soll. Mit neuesten Endgeräten und zu gestaltenden Business-Apps schafft solch ein „Instant Business“ neue Märkte und Umsatzchancen.

### 1.1 Motivation

Aktuell vollzieht sich ein grundlegender Wandel: die Business-IT wird mobil. Vorbilder sind dabei weniger Nokia, Siemens Mobile oder Microsoft – sondern Player

---

Claudia Linnhoff-Popien  
Ludwig-Maximilians-Universität München, Institut für Informatik,  
E-mail: linnhoff@ifi.lmu.de

Stephan Verclas  
T-Systems International GmbH, München,  
E-mail: stephan.verclas@t-systems.com

wie Google, Facebook und Apple. In kürzester Zeit rollen sie bestehende Märkte aus und besetzen neue. Wirtschaftspolitisch gilt es, den amerikanisch geprägten Schwergewichten deutsche Lösungen entgegenzusetzen – neue Innovationen zu wagen, um nicht von der mobilen Applikationswelt überrollt zu werden.

Das Potential der Business-Apps nimmt dieses Potenzial in Angriff: Intelligente mobile Apps werden Geschäftsprozesse im wahrsten Sinne des Wortes „mobilisieren“. Es entsteht ein mobiles Internet, dass den Geschäftsprozessen in allen Branchen neue Möglichkeiten gibt: Von der Unterstützung des Außendienstmitarbeiters bis hin zu Meldungen wie Unwetter- und Infektionswarnung oder sicherheitsrelevanten Hinweisen auf Großveranstaltungen werden dem Kunden Informationen individualisiert und zeitnah auf dem Smartphone zur Verfügung gestellt.

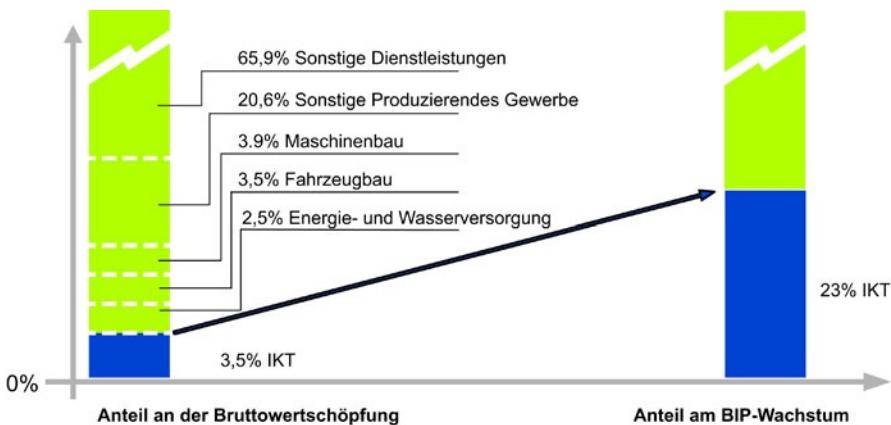
Das Thema der mobilen Business-Apps ist im Kern ein IT-Thema, es spielt jedoch in unterschiedlichsten Wirtschafts- und Industriebranchen eine elementare Rolle für den Geschäftserfolg. Dazu gehören die Automobil- und Elektroindustrie gleichermaßen wie der Maschinenbau, die Energieversorgung, die Finanz- und Medienbranche, aber auch der Logistikbereich, um nur einige zu nennen. Nicht nur die internen Geschäftsprozesse der Unternehmen sind zu mobilisieren, auch und vor allem die Geschäftsabläufe zwischen Unternehmen und Kunden ändern sich sehr stark. Kunden und Geschäftspartner werden direkter und unmittelbarer in die Geschäftsabläufe einbezogen, die Kundenkontakte finden situations- und ortsabhängiger statt. Durch die vielfältigen, mobilen Endgeräte können die Informationen sowohl von Kunden und Mitarbeitern als auch von Geschäftspartnern direkt in die eigenen IT-Systeme einfließen: ein Paradigmenwechsel, der die Business-IT von Grund auf umwälzen wird.

## 1.2 Marktsituation

Bevor das mobile Internet und mobile Geschäftsprozesse betrachtet werden, soll zunächst auf die Bedeutung der Informations- und Kommunikationsindustrie (IKT) im Allgemeinen eingegangen werden. Dies ist deshalb so entscheidend, da der volkswirtschaftliche Nutzen der IKT in letzter Instanz erst durch die Anwendungsbranchen zur Geltung kommt.

In den letzten Jahren hat die Bedeutung von IKT in Unternehmen stetig zugenommen. Laut einer aktuellen Studie des Zentrums für Europäische Wirtschaftsforschung (ZEW) führen inzwischen 40 % aller innovativen Unternehmen in Deutschland Neuerungen ein, die auf IKT beruhen [2]. Der volkswirtschaftliche Nutzen wird durch die hohe Innovationskraft der IKT-Branche deutlich und steigt zunehmend: Obwohl heute „nur“ 3,5 % des BIP von IKT-Dienstleistungen erbracht werden, geht knapp ein Viertel des BIP-Wachstums auf den Einsatz von IKT zurück [3].

Im Jahr 2010 haben 81 % der Anbieter von Software und IKT-Services sowie 78 % der Hardwarehersteller neue Produkte oder Dienste auf den Markt gebracht. Die IKT-Branche liegt damit noch vor Elektrotechnik (70 %), Chemie- und Pharma-



**Abb. 1.1** IKT als Wachstumsmotor nach Statistischem Bundesamt und Zentrum für Europäische Wirtschaftsforschung

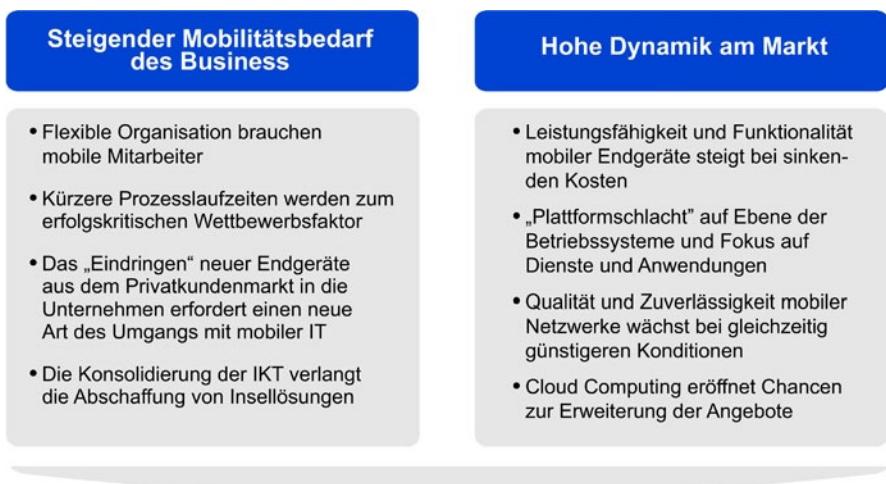
Industrie (69 %) und Maschinenbau (64 %), so die Ausführungen der ZEW in „ITK als Wegbereiter von Innovationen“ von 2010 [2].

Mit einem erwarteten Marktvolumen von über 6 Mrd. Euro in 2011 und einer Wachstumsrate von über 8 % bleibt der mobile Datendienstleistungsbereich nach einer Studie der Germany Trade and Invest Gesellschaft für Außenwirtschaft und Standort-Marketing das am schnellsten wachsende Segment im deutschen Telekommunikationsmarkt [4].

Die zunehmende Flexibilisierung und Mobilität von Arbeitsumgebungen und Prozessen lässt sich nach IDC [5] an mehreren Indikatoren messen: Die Zahl mobil arbeitender Mitarbeiter in den Unternehmen steigt ständig. Bis 2011 werden bis zu 1 Mrd. Arbeitnehmer weltweit „mobil“ sein. Ermöglicht wird dieses neue Paradigma durch die rasant wachsende Verbreitung von Smartphones. Bereits 2008 bis 2010 ist die Zahl der Smartphones von 3,1 Mio. auf 8,2 Mio. Stück gestiegen.

Die Kombination dieser Entwicklungen mit dem gleichzeitigen Wachstum der mobilen Internetnutzung sowie des mobilen Datenverkehrs legt die Schlussfolgerung nahe, dass auch der Bedarf für mobile Unternehmensapplikationen exponentiell wachsen wird. Smart Mobile Apps werden dabei zunehmend zu einem geschäfts-kritischen Baustein in der IKT-Infrastruktur und erlangen entscheidende Bedeutung als Wachstumstreiber sowohl für die IKT-Branche als auch als produktivitätssteigerndes Element für andere Wirtschaftszweige.

Darüber hinaus gibt es weitere treibende Faktoren für mobile Geschäftsanwendungen: Der intensive Wettbewerb erfordert von den Unternehmen immer kürzere Prozesslaufzeiten, wie sie nur noch mit mobilen Lösungen zu erreichen sind. Innovative Endgeräte aus dem privaten Umfeld wie z. B. iPhones kommen zunehmend auch in den Unternehmen zum Einsatz und fordern von den Sicherheitsverantwortlichen neue Konzepte und Lösungen. Zudem lassen sich mobile Insellösungen im Umfeld einer stetigen Konsolidierung der IKT nur noch schwer aufrechterhalten. Der Markt wird indes getrieben durch leistungsfähige mobile Endgeräte und mehr



### Eine neue Generation mobiler Lösungen reift heran

**Abb. 1.2** Treibende Faktoren von Enterprise Mobility, nach Berlecon 2010

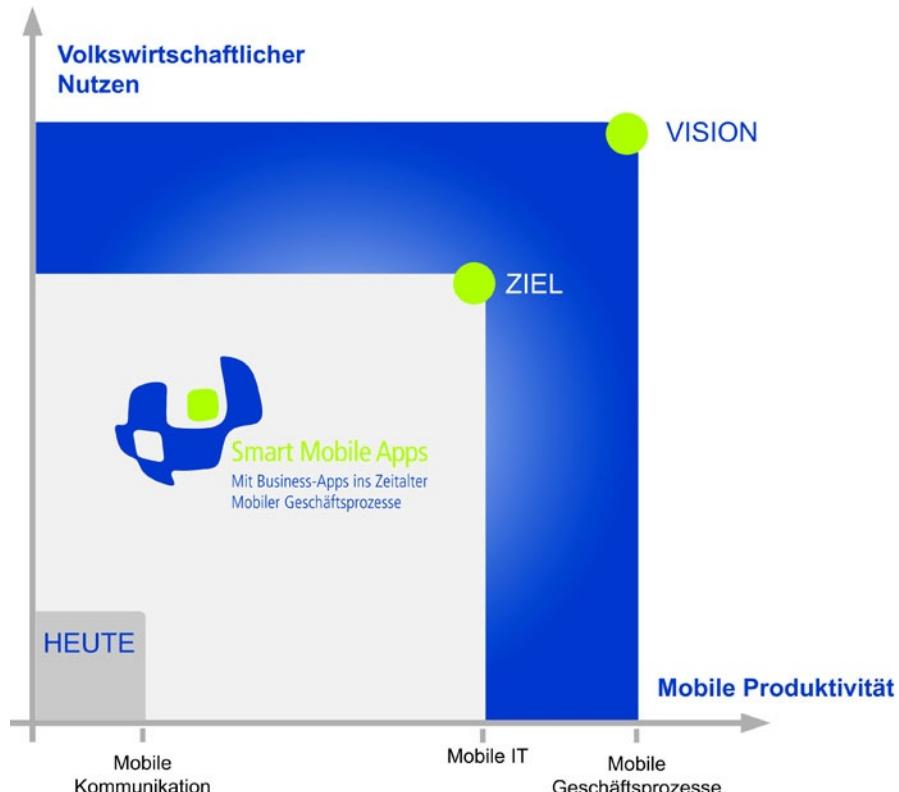
Zuverlässigkeit und Qualität bei den Netzen bei gleichzeitigem Preisverfall. Eine Differenzierung bei den Endgeräten erfolgt inzwischen nicht mehr nur über die Hardware, sondern immer mehr über Betriebssysteme und die dafür verfügbaren Dienste und Anwendungen (siehe Abb. 1.2).

Die Voraussetzungen für eine neue Generation mobiler Applikationen sind somit gegeben.

Bei der Anzahl der in App Stores verfügbaren Apps liegt Apple mit dem iPhone und 350.000 Apps an der Spitze, Stand Januar 2011 [6]. Im April 2011 gab es zudem 340.000 Android-Apps. Für die Kunden von RIM mit Blackberry, die eher auf Geschäftsanwendungen abzielen, stehen nur 7800 Apps zur Verfügung. Gartner geht davon aus, dass beispielsweise Apple sich auch weiterhin auf den Konsumtenmarkt konzentriert und nur taktische Anstrengungen im Bereich der Entwicklung von Enterprise Applikationen zeigt. Zudem ist Apple nach Gartner nur für diejenigen Unternehmen eine geeignete Wahl, für die Sicherheit eine untergeordnete Rolle spielt. Das Marktpotenzial und der Nachholbedarf sind aus diesem Grunde im Bereich Geschäftsanwendungen enorm.

## 1.3 Mobile Geschäftsprozesse

Produktivitätssteigerung und Prozessbeschleunigung sind seit jeher Treiber für Wachstum und Wohlstand. Business-Apps schaffen durch die Mobilisierung der Geschäftswelt die Voraussetzungen für bessere und schnellere Abläufe und eine ef-



**Abb. 1.3** Erhöhung des volkswirtschaftlichen Nutzens mobiler Geschäftsprozesse

fektivere Nutzung der Arbeitszeit. So leisten sie einen wichtigen Beitrag zur volkswirtschaftlichen Entwicklung.

Heute beschränkt sich die Nutzung von mobilen Endgeräten im Wesentlichen auf die Kommunikation und Interaktion zwischen Menschen. Sprache, SMS, E-Mail und Instant Messaging bieten eine große Bandbreite an Möglichkeiten. Mit neuen Smart Mobile Apps insbesondere im Geschäftsumfeld können auch die klassischen IT-Systeme mobilisiert werden, die in modernen Unternehmen die Arbeitsabläufe unterstützen. Damit wird es den Mitarbeitern möglich, von unterwegs aus produktiv zu sein und Reisezeit als Arbeitszeit zu nutzen. Die Effizienz des Einzelnen steigt und aus volkswirtschaftlicher Sicht kann in der gleichen Zeit mehr Wert geschaffen werden.

Der volle volkswirtschaftliche Nutzen kann durch die Umsetzung der Vision komplett mobilisierter Geschäftsprozesse erreicht werden. Langfristig können technische Voraussetzungen geschaffen werden und somit ein maximaler Beitrag zu Wachstum und Beschäftigung geleistet werden (siehe Abb. 1.3).

An drei konkreten Beispielen soll der Einsatz von Business-Apps im Folgenden veranschaulicht werden.

### Apps for Cars – Aktive und passive Verkehrssicherheit

Gerade im Auto erleben Fahrer und Beifahrer eine hohe Ausprägung von Mobilität. Die Infrastruktur künftiger Fahrzeuge bietet zudem eine optimale Ausstattung mit Displays, Speicher- und Prozessorkapazität sowie Verbindung zum Netz. In Zusammenarbeit mit der Automobilindustrie und den Automobilverbänden kann dieses Potenzial genutzt werden, um die aktive und passive Verkehrssicherheit zu erhöhen und Informationsdienste anzubieten. Erste Apps finden hier schon Verbreitung, jedoch sind diese plattformspezifisch und haben eher informativen Charakter.

### Apps for Finance – Claim Notification

Der Versicherungs- und Bankenbereich ist geprägt von vielfältigsten Transaktionen, die in Echtzeit mit einem hohen Grad an Sicherheit und Privacy ausgeführt werden sollen. Bei einer Migration von Finanztransaktionen in das mobile Internet muss höchsten Formen von Sicherheit und Privacy Rechnung getragen werden. Beim Nachhalten von Schadensfällen kommt ferner der Wunsch nach Authentifizierung sowie nach Aufbereitung von Kontext wie z. B. der Ort, die Zeit, die involvierten Personen oder der Situation der Schadensentstehung hinzu. Die Schadensaufnahme vor Ort z. B. eines Verkehrsunfalls (aber auch von Einbruch-, Wasser-, Sturm- oder Vandalismusschäden) hat den Vorteil, dass der Kunde über seine mobile App direkt mit dem Versicherungsunternehmen kommunizieren kann und in der Lage ist, situativ auf Wünsche des Agenten zu reagieren. Beweisfotos können mit der Kamera des Smartphones aufgenommen und mit GPS-Koordinaten sowie der aktuellen Uhrzeit versehen werden.

### Apps for Media – Video on demand

Smartphones und Handys kommt besonders dann eine große Bedeutung zu, wenn der Nutzer keine Alternative hinsichtlich Infrastruktur und Endgeräten hat. In der Regel ist er dann unterwegs, er reist von einem Ort zum anderen. Klassische Informationsdienste stehen bereits zur Verfügung. Diese sollen jedoch erweitert werden um High-End-Services, die mit Premium-Inhalten und höchsten Bandbreiten hohe Qualität bieten. Der Ausbau von Netzen (LTE) wird hier gleichermaßen von Nutzen sein wie der Preisverfall bei Speicher, Hardware und Displays. So kann die hohe Bandbreite genutzt werden, um kommerzielle Videos herunterzuladen und dem Nutzer gegen Entgelt anzubieten. Dabei sind Inhalteanbieter in der Lage, ihre Produkte in Video Stores anzubieten. Der Store-Betreiber sorgt dann mit einem



**Abb. 1.4** Wertschöpfungskette klassischer und mobiler Unternehmensanwendungen

fairen Geschäftsmodell dafür, dass Zahlungen auf sicherem Wege stattfinden und Netzbetreiber, Inhalteanbieter sowie weitere Beteiligte in fairer Weise entlohnt werden.

## 1.4 Die Wertschöpfungskette für Business-Apps

Aus dem Prozess des Auseinandersetzens mit Trends und Entwicklungen unserer Gesellschaft resultieren zahlreiche Anforderungen an die Informations- und Kommunikationstechnologie. Einer der größten Megatrends, der sich in zahlreichen Unternehmensstrategien widerspiegelt, ist die Mobilisierung der Geschäftswelt. Es gilt Innovationen für Mitarbeiter, Partner und Kunden zu entwickeln, um unabhängig von Ort, Zeit und Endgerät auf Daten und Anwendungen zuzugreifen.

Der noch sehr junge Markt für mobile Geschäftsanwendungen bringt dabei eine einschneidende Änderung der Marktstrukturen mit sich – praktisch kein Markttakteur kann die gesamte Wertschöpfungskette abdecken. Im Gegensatz zur klassischen Bereitstellung von Unternehmensanwendungen gestaltet sich die Einführung von mobilen Geschäftsanwendungen wesentlich komplexer.

Die Wertschöpfungskette für Business-Apps ist im Gegensatz zu der für klassische Anwendungen extrem fragmentiert und komplex. Eine ganze Reihe von Unternehmen wie Hersteller, Service Provider oder System-Integratoren leisten jeweils einen kleinen Beitrag zur Implementierung mobiler Unternehmensanwendungen (siehe Abb. 1.4).

Entscheidend für den Wirtschaftsstandort Deutschland ist die branchenübergreifende Kooperation für die verschiedenen Stufen der Wertschöpfungskette. Die Zusammenarbeit mit etablierten Unternehmen und wissenschaftlichen Partnern kann gewährleisten, dass die Kompetenzen der IKT-Branche auf andere Branchen übertragen werden, wodurch ein breites Wissensnetz über verschiedene Wirtschaftsbereiche hinweg aufgebaut werden kann. Das Thema Business-Apps ist dabei ein sehr junges Thema.

## 1.5 Technische Herausforderungen

Betrachtet und analysiert man das Thema der mobilen Business-Apps, so finden sich die folgenden technischen Barrieren, die derzeit das exponentielle Wachstum bezüglich Anzahl, Download und Umsatz bei Business-Apps noch bremsen.

### Heterogenität der mobilen Plattformen

Die Tatsache, dass derzeit mehrere mobile Betriebssysteme parallel am Markt existieren wie z. B. Blackberry, Apple, Android, Windows Phone 7 und derzeit nicht abzusehen ist, dass sich eines alleine mittelfristig durchsetzen wird, schafft Probleme bei der Entwicklung von Business-Apps und dem Management der verschiedenen Endgeräte. In der Regel muss für jedes Betriebssystem eine eigene Management- und Entwicklungsinfrastruktur vorgehalten werden, was zu hohen Kosten und langen Entwicklungszeiten führt.

### Entkopplung der Lebenszyklen

Besonders im Bereich von Business-Apps in Fahrzeugen, Maschinen und Industrieanlagen stellen die unterschiedlichen Lebenszyklen der Applikationen und der Industriegüter, in die diese eingebaut sind, eine große Herausforderung dar. Offensichtlich wird das Problem im Auto, dessen Nutzungsdauer etwa 10 Jahre beträgt, während das eingebaute Navigationssystem schon nach 1–2 Jahren nicht mehr dem technischen Standard entspricht.

### Skalierbarkeit

Individuelle und spezialisierte mobile Geschäftsanwendungen lassen sich für eine überschaubare Anzahl von Nutzern relativ leicht ausrollen und auch auf dem aktuellen Stand halten. Werden die Anwendungen jedoch in großer Zahl und mit vielen Varianten ausgerollt, so stellt sich schnell die Frage nach der Skalierbarkeit. Wie können Tausende von Endgeräten mit neuen Versionen einer Software betankt werden und wie kann im Einzelnen sichergestellt werden, dass die neue Softwareversion sich mit dem Betriebssystem des Endgerätes verträgt?

### Sicherheit

Gerade für Business-Apps stellt sich das Problem, dass mobile Endgeräte häufig von der gleichen Person in unterschiedlichen Rollen genutzt werden. Eine Trennung von privaten und geschäftlichen Daten sowie die Implementierung unterschiedlicher

Sicherheitsstufen für die beiden Bereiche ist eine wesentliche Herausforderung im Bereich Sicherheit.

Integration in Geschäftsabläufe und die vorhandene IT-Landschaft

Apps für Endkunden sind oft als sogenannte Stand-alone-Apps ausgelegt, das heißt, sie benötigen zum Funktionieren keine Verbindung zu IT-Systemen im Hintergrund. Ganz anders bei mobilen Geschäftsanwendungen. Hier ist eine Anbindung an die Unternehmens-IT unerlässlich und führt bei der heutigen Vielfalt von Schnittstellen und technischen Standards zu erheblichen Schwierigkeiten. Zudem sind die Prozesse in der Regel nicht für die mobile Nutzung ausgelegt und müssen deren Anforderungen angepasst werden.

## 1.6 Gestaltungskomponenten von Business-Apps

Business-Apps sind aus unterschiedlichen Sichtweisen zu betrachten und haben vielerlei Facetten. Bei der Anzahl von möglichen Betrachtungsweisen soll im Folgenden zu einer Darstellung übergegangen werden, welche die reine IT zwischen eine Anwendungs- und eine Marktsicht einbettet.

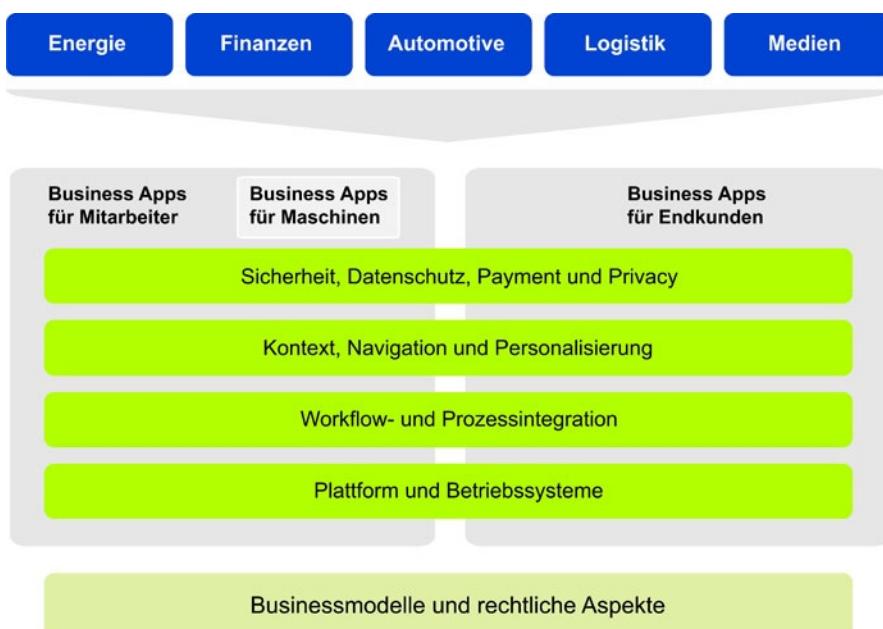


Abb. 1.5 Thematische Bausteine zur Gestaltung von Business-Apps

Dabei gibt es zunächst – wie in Abb. 1.5 dargestellt – vertikale Ebenen der unterschiedlichen Arten von Apps, und dann horizontale Ebenen der verschiedenen Bereiche, für die Apps von Bedeutung sind.

### Business-Apps für Mitarbeiter und Maschinen

Business-Apps für Maschinen kommen beispielsweise in der Logistik zur Anwendung, in Systemen der Fahrerassistenz, der Industrie oder in Krankenhäusern – vom LKW bis zum medizinischen Gerät. Dabei ist robuste Hardware erforderlich, die Extremforderungen bezüglich Temperatur, Erschütterung und Feuchtigkeit standhält. Eine weitere Herausforderung sind die langen Lebenszyklen der Maschinen – im Gegensatz zu einer zweijährigen Lebenszeit des klassischen Handys hat ein LKW eine Lebensdauer von 10 bis 15 Jahren, eine Industrieanlage wird über einen noch längeren Zeitraum betrieben.

Apps für Mitarbeiter und Apps für Maschinen müssen in die IT der Unternehmen eingebunden und gemanagt werden. Im alltäglichen Betrieb verschwimmen zunehmend die Grenzen zwischen privater und geschäftlicher Nutzung der Geräte. Es ist deshalb notwendig, technische Maßnahmen zu ergreifen, um die private und die unternehmenskritische Anwendung auf demselben Gerät nebeneinander zu betreiben. Es müssen neue Virtualisierungs- und Authentifizierungsmaßnahmen implementiert, und die Nutzeridentität muss stärker von der mobilen Hardware getrennt werden.

Unternehmenskritische Anwendungen und die zunehmende Verlagerung von Diensten und Inhalten in die Cloud verlangen ferner einen durchgängigen Netzzugang. Deshalb gewinnen breitbandige Verbindungen über Mobilfunktechniken immer mehr an Bedeutung.

### Business-Apps für Endkunden

Während klassische Apps aus dem Consumer-Umfeld heraus gewachsen sind, bedingt eine Durchdringung von Geschäftsprozessen des mobilen Internets eine völlig neue Akzeptanz.

Zu den grundlegenden Herausforderungen gehören dabei zunächst eine einfache, zielgruppenspezifische Nutzeroberfläche und Bedienbarkeit. Die durch mobile Endgeräte entstehende Komplexität soll über die Produktlaufzeit hinweg vom Nutzer nicht wahrnehmbar sein. Apps müssen plattformunabhängig realisiert werden und auf möglichst vielen Endgeräten am Markt funktionieren. Hierzu ist es notwendig, alle gängigen Plattformen und Betriebssysteme abzudecken. Apps für sensible Interaktionen, wie sie beispielsweise in der Finanzbranche üblich sind, müssen höchsten Anforderungen an die Datensicherheit genügen. Mit steigender Komplexität der Vernetzung entstehen neue Arten von Gefahren und dadurch neue Anforderungen an die Sicherheit. Zum Abrechnen ist schließlich ein einfaches und sicheres Bezahl- system erforderlich, das für mobile Anwendungen geeignet ist.

## Sicherheit, Datenschutz, Payment und Privacy

Mit der Entwicklung der Mobiltelefone weg von reinen Lösungen für Gespräche und hin zu einem reichhaltigen Angebot an Services auf leistungsfähigen und ergonomisch gestalteten Smartphones findet auch eine zunehmende Nutzung von mobilen Anwendungen statt.

Daneben ist eine Konvergenz von Rollen auf dem mobilen Endgerät auszumachen: Anstatt für jede Rolle ein eigenes Handy mit sich herumzutragen, benutzen viele Leute ihr Gerät in mehreren Funktionen – als Privatperson, Konzernmitarbeiter oder in der Kommunikation mit Geschäftspartnern und Kunden. Diese Konvergenz birgt unter Sicherheits- und Datenschutzgesichtspunkten einige Herausforderungen, begonnen mit dem unbeabsichtigten oder von Dritten böswillig herbeigeführten Abfluss von Informationen zwischen Privat- und Geschäftsnutzung.

Weil große Player auf dem Markt wie z. B. Apple mit dem iPhone sich nicht durch Offenheit auszeichnen, ist ein natürliches Misstrauen gegenüber dieser Plattform vorhanden. Ziel ist es, durch die Schaffung von Vertrauen und das Ausräumen von Bedenken gegenüber dieser Technologie nachweisbar Sicherheit und Transparenz garantieren zu können, um der Plattform zum Erfolg zu verhelfen. Durch die Einführung von Gütesiegeln und Zertifizierungen sollen Business-Apps hinsichtlich ihrer Sicherheit klassifiziert werden. Ferner sollen die Spannungsfelder zwischen Sicherheit und Benutzbarkeit oder Privacy im Kontext der Smart Mobile Apps betrachtet werden.

## Kontext, Navigation und Personalisierung

Das Thema Kontext, Navigation und Personalisierung ist ein grundlegendes Thema mobiler Diensterbringung, da mobile Endgeräte ihren Ort ständig wechseln und darauf in geeigneter Weise reagieren sollen. Die Ein- und Ausgabe ist auf einem mobilen Endgerät extrem eingeschränkt, und die Sensorik der Geräte liefert wertvolle Informationen. Bei einer ortsbasierten Websuche wird beispielsweise additiv zum Suchbegriff auch der aktuelle Aufenthaltsort des Endgerätes berücksichtigt und schon durch die Eingabe von wenigen Buchstaben eine sinnvolle Suche ermöglicht.

Die grundlegenden Fragestellungen bestehen im Typ der Nutzerinteraktion und der Entscheidung, welche Informationen proaktiv bzw. reaktiv angeboten werden. Allerdings wird die Ausbringung vielfältiger Informationen über ein reaktives System schnell zu einem Engpass für die Skalierbarkeit. Empfehlungssysteme sollen – und das ist der fundamentale Unterschied zwischen dem Markt für Privatkunden und dem Business-Markt – nicht nur für die Diensterbringung irrelevante Information wie z. B. Werbung personalisieren, sondern auch fundamental relevante Daten wie Kundendaten auf den Bedarf abstimmen.

Eine große Herausforderung besteht ferner in der Erstellung eines Dienstmodells, welches neben der Dienstauffindung, -komposition und -empfehlung auch den Austausch von Personalisierungsinformationen umfasst. So muss – entgegen der Situation im Privatkundenmarkt – nicht auf eine Person hin die Personalisierung erfolgen,

sondern auf verschiedene, sehr komplexe und im Verhältnis zueinander stehende Einheiten bzw. Rollen.

### Workflow- und Prozessintegration

Um die zunehmende Verbreitung mobiler Endgeräte im Umfeld prozessgetriebener Abläufe optimal zu nutzen, gilt es zunächst, die speziellen Fähigkeiten dieser Gerätekasse zu unterstützen. Multimodale Bedienkonzepte sind zu entwickeln, die in Abhängigkeit von Kontextparametern eine angepasste Steuerung der Applikationen beispielsweise in lauten Umgebungen oder am Steuer eines Fahrzeugs ermöglichen. Wechsel zwischen verschiedenen Anwendungen und Bedienkonzepten wirken insbesondere auf Mobilgeräten disruptiv, so dass eine einheitliche Schnittstelle für die Integration verschiedener Prozesslösungen in eine gemeinsame Arbeitsumgebung für den Endanwender benötigt wird.

Als Antwort auf diese Herausforderungen sollen Ansätze für die Integration mobiler Anwender in Workflows und Prozesse erarbeitet werden, die nicht nur eine optimale Ansteuerung der Endgeräte, sondern auch die Vereinheitlichung des Zugriffs auf die zugrundeliegenden Verfahren und die Optimierung der Prozessmodelle umfassen. Dies betrifft Anwendungssoftware, Middleware und Entwicklungswerkzeuge, die Qualität und Effizienz der Projektrealisierung maximieren. Die Vielfalt der Beiträge der verschiedenen Unternehmen in diesem Buch, deren Tätigkeitsfelder alle Schichten der zu betrachtenden Technologiestapel in jeweils verschiedenen Ausprägungen umfassen, begünstigt die Entwicklung generischer und umfassend einsetzbarer Software- und Methodiklösungen.

### Plattformen und Betriebssysteme

Plattformen und Betriebssysteme sind eine tragende Säule für die Umsetzung von Geschäftsanwendungen. Gängige Endgeräte unterscheiden sich in der Hardware, dem mobilen Betriebssystem, Protokollarchitekturen, Middleware und Anwendungen. Die wichtigsten Herausforderungen hierbei sind die Heterogenität, die mit ihrer Überwindung einhergehende Interoperabilität, die Skalierbarkeit, die 100 %-Konnektivität, die Sicherheit und die Komplexitätsreduzierung.

Dabei stellt sich zunächst die Frage, wie viel Intelligenz in ein Gerät soll. Das Spektrum reicht dabei von einer eigenständigen Business-App auf dem Endgerät über Thin Clients bis hin zu einer browsergestützten Business-App, die nur noch Ein- und Ausgabe behandelt, aber keine ernsthafte Geschäftslogik mehr enthält.

## Businessmodelle und rechtliche Aspekte

Die Entwicklung profitabler Geschäftsmodelle ist von zentraler Bedeutung für die erfolgreiche Kommerzialisierung von Business-Apps – insbesondere im Geschäftskundensegment. Eine große Herausforderung für diese Geschäftsmodelle ist es, einen Weg zu zeichnen vom App-Produktgeschäft, d. h. Einzeltransaktionen wie dem Verkauf von Spielen im B2C-Markt, zu einem B2B-Systemgeschäft mit inhaltlichen und zeitlichen Verbundeffekten und entsprechenden Preissystemen. Es gibt bereits erfolgreiche Einzelbeispiele von „B2B-Apps“, etwa die Programme von Salesforce.com, um Kundenbeziehungen über das iPhone, Blackberry oder über Tablet-PCs zu führen. Zunehmend können Mitarbeiter mit Smartphones Arbeiten mobil erledigen, die früher nur vom Laptop oder Desktop-Rechner aus möglich waren.

Ein weiteres zentrales Thema bei der Entwicklung von Geschäftsmodellen mit mobilen Apps ist die Analyse der Kostenfaktoren, da viele Produkte reine IT-Lösungen darstellen, die einen bedeutenden Einfluss auf die Kostensituation des Unternehmens haben. So kommt der Steuerung von Innovationsprozessen bei der Entwicklung von Business Apps und der Bereitstellung dieser Apps eine besondere Bedeutung zu. Insbesondere spielen nach J. Howe die Konzepte Open Innovation, Kundenintegration und Crowdsourcing bei der Entwicklung von Apps eine wichtige Rolle, um die Einbindung von Kunden zu gewährleisten.

## 1.7 Bewertung und Ausblick

Das Analystenhaus Gartner [7] geht nach einer Einordnung im „Magic Quadrant for Mobile Enterprise Application Platforms“ seit Dezember 2009 davon aus, dass Apple als Marktführer bei verfügbaren Apps sich auf den Consumer-Markt konzentriert und eher geringe Anstrengungen im Bereich der Enterprise- Applikationen unternimmt. Diese sind zudem auch vornehmlich taktisch geprägt.

Dennoch ist es so, dass sich nach dem Berlecon-Report „Mobile enterprise solutions“ aus dem Jahr 2010 [8] der Markt für mobile Unternehmenslösungen neu justiert: Immer vielschichtigere und zunehmend ausgereifte Technologieangebote treffen auf einen steigenden Bedarf an mobilen Lösungen sowohl zur Verbesserung der Unternehmenskommunikation als auch zur Geschäftsprozessoptimierung.

Einem großen Bedarf und einer großen Nachfrage steht demnach nichts an Lösungen gegenüber. – Eine herausfordernde Situation, die es zu gestalten gilt!

## Literaturverzeichnis

1. Gartner (2011) Gartner Says Worldwide Mobile Application Store Revenue Forecast to Surpass \$15 Billion in 2011, 26.01.2011

2. Zentrum für Europäische Wirtschaftsforschung im Auftrag von BITKOM (2010) Informations- und Telekommunikationstechnologien als Wegbereiter für Innovationen
3. BITKOM (2010) ITK ist Motor für Wachstum und Innovation
4. Germany Trade and Invest – Gesellschaft für Außenwirtschaft und Standortmarketing mbH (2010) Web 2.0: Mobile Datendienste und IT-Dienstleistungen beschleunigen Wachstum der deutschen IKT-Branche
5. IDC (2010) More Than One Billion Mobile Workers Worldwide by Year's End
6. Apple (2011) Über 10 Milliarden Apps aus dem App Store von Apple heruntergeladen
7. Gartner (2009) Magic Quadrant for Mobile Enterprise Application Platforms
8. Berlecom Research (2010) Mobile Enterprise Solutions 2010

**Teil II**

**Anwendersicht: Business-Apps für  
Endkunden**

---

## Kapitel 2

# Zum Einfluss mobiler Endgeräte auf das individuelle Gesundheitsverhalten von PKV-Versicherten

Tristan Nguyen, Jörg Lindenmeier, Tobias Fritsch und Frederik Kerssenfischer

**Zusammenfassung** Gesundheitsmärkte sind entscheidend durch das Phänomen der asymmetrischen Informationsverteilung charakterisiert. Der Begriff der asymmetrischen Informationsverteilung umschreibt, dass zwischen den Akteuren des Gesundheitswesens vor und/oder nach Abschluss von Kontrakten ungleich verteilte Informationsbestände vorliegen. Als Akteure auf dem PKV-Markt treten die privaten Krankenversicherer als Kostenträger, die Versicherten als Nachfrager von Gesundheitsleistungen sowie ambulante, stationäre und integrierte Versorgungseinrichtungen als Leistungserbringer auf. Neben dem Problem der „Adversen Selektion“, welches durch verschiedene Formen des „Signalings“ und „Screenings“ gelöst werden kann, stellt das Phänomen des „Moral Hazard“ die wesentliche Konsequenz bestehender Informationsasymmetrien dar. Das Phänomen des „Moral Hazard“ kann u. a. sowohl im Kontext von Vertragsbeziehungen zwischen den privaten Krankenversicherungen und Privatversicherten als auch zwischen den Privatversicherten und Ärzten auftreten. Mit der Einführung des iPhone durch Apple und die spätestens dadurch eingeläutete Ära der „Smartphones“ ergeben sich neue Möglichkeiten zu versuchen, die auf den oben beschriebenen Feldern bestehenden Probleme zu lösen bzw. abzumildern. Konkret gesprochen, kann mit dem Einsatz von mobilen Endgeräten (z. B. Smartphones oder Tablet-PCs) am Kern der beschriebenen Probleme angesetzt werden. Dies bedeutet, dass man durch den Einsatz von mobilen Endgeräten Informationsdefizite verringern kann, die auf Seiten der Krankenversicherungen bzw. der Krankenversicherten bestehen. Es kann vermutet werden, dass

---

Tristan Nguyen

WHL Wissenschaftliche Hochschule Lahr, E-mail: tristan.nguyen@whl-lahr.de

Jörg Lindenmeier

WHL Wissenschaftliche Hochschule Lahr, E-mail: joerg.lindenmeier@whl-lahr.de

Tobias Fritsch

Allianz Managed Operations & Services SE, E-mail: t.fritsch@allianz.de

Frederik Kerssenfischer

Allianz Managed Operations & Services SE, E-mail: frederik.kerssenfischer@allianz.de

die durch diesen Abbau von Informationsasymmetrien bedingte Verringerung der Moral-Hazard-Problematik in einer effizienteren Allokation knapper Ressourcen münden wird. Dies wird Krankenversicherungen Spielräume eröffnen, mit einer attraktiveren Prämiengestaltung am Markt zu konkurrieren.

## 2.1 Problemstellung

Gesundheitsmärkte und Versicherungsmärkte sind entscheidend durch das Phänomen der asymmetrischen Informationsverteilung charakterisiert. Der Begriff umschreibt, dass zwischen den Akteuren (Kunde/Versicherung) vor und/oder nach Abschluss von Kontrakten ungleich verteilte Informationsbestände vorliegen. Als Akteure auf dem PKV-Markt treten die privaten Krankenversicherer als Kostenträger, die Versicherten als Nachfrager von Gesundheitsleistungen sowie ambulante, stationäre und integrierte Versorgungseinrichtungen als Leistungserbringer auf. Neben dem Problem der „Adversen Selektion“ [1], welches durch verschiedene Formen des „Signaling“ und „Screening“ gelöst werden kann, stellt das Phänomen des „Moral Hazard“ die wesentliche Konsequenz bestehender Informationsasymmetrien dar [2].

In Bezug auf Versicherungsmärkte wird von *Moral Hazard* gesprochen, wenn das Individuum, weil es versichert ist, eigene Maßnahmen zur Reduzierung seines Risikos für einen Schadensfall vernachlässigt.<sup>1</sup> Arrow [4] bezeichnet Moral Hazard als die Tatsache, dass der Versicherungsvertrag selbst die Anreize und damit die Schadenwahrscheinlichkeit ändert, auf deren Basis der Versicherungsvertrag zustande kommt.<sup>2</sup> Das Phänomen des „Moral Hazard“ kann sowohl im Kontext von Vertragsbeziehungen zwischen den privaten Krankenversicherungen (Kostenträger) und Privatversicherten (Leistungsnachfrager) als auch zwischen den Privatversicherten (Leistungsnachfrager) und Ärzten (Leistungserbringer) auftreten (zur theoretischen Begründung und empirischen Evidenz vgl. hierzu [5]). Eine Ausprägung des Moral-Hazard-Verhaltens stellt etwa das Phänomen der angebotsinduzierten Nachfrage dar. In der Vertragsbeziehung zwischen Versicherten und Ärzten besteht zunächst das Problem, dass die Nachfrager behandlungsorientiert denken. Das heißt, dass die Privatversicherten – unabhängig von den Behandlungskosten – die angeblich für sie beste Behandlungsleistung in Anspruch nehmen wollen. Dazu kommt, dass die Leistungserbringer einkommensorientiert denken und somit präferieren, die Leistungen zu verordnen, die ihr Einkommen maximieren. Zudem können die Versicherten – aufgrund des massiven Informationsvorsprungs des Arztes über die angemessene Behandlung – nur sehr eingeschränkt beurteilen, ob die ärztliche Behandlungsempfehlung medizinisch sinnvoll ist. Dies führt dazu, dass die Leistungserbringer die Möglichkeit haben, die Nachfrage nach Gesundheitsleistungen künstlich aufzublähen [6].

---

<sup>1</sup> Vgl. [3], S. 4.

<sup>2</sup> Vgl. [4], S. 142.

Im Folgenden wird diskutiert, wie die asymmetrische Informationsverteilung zwischen Versicherten und Versicherungsunternehmen und das daraus resultierende Moral-Hazard-Verhalten mit Hilfe des Ansatzes mobiler Endgeräte abgemildert werden kann. Darauf aufbauend werden wesentliche Ansatzpunkte für die zukünftige Forschung zum Einsatz mobiler Endgeräte in dem umrissenen Anwendungsbereit abgeleitet.

## **2.2 Mobile Endgeräte und Moral-Hazard-Verhalten**

Im Rahmen der nachfolgenden Ausführung wird der Zusammenhang zwischen den Themenkreisen des Einsatzes mobiler Endgeräte in (kranken-)versicherungswirtschaftlichen Kontexten und dem Phänomen des Moral-Hazard-Verhaltens zunächst allgemein beleuchtet. Im Anschluss daran werden Anknüpfungspunkte für die zukünftige Forschung entwickelt.

### ***2.2.1 Der Einsatz mobiler Endgeräte zur Reduzierung Moral-Hazard-induzierter Probleme***

In der vorangegangenen Problemstellung wurde geschildert, dass die Versicherten aufgrund der eingeschränkten Möglichkeiten des Versicherungsgebers, das Verhalten der Versicherten zu beobachten, und der daraus resultierenden ex-post asymmetrischen Informationsverteilung einen Anreiz haben, ihre Anstrengungen zur Schadensverhinderung zu minimieren. Dies resultiert in einer Situation, in der Ressourcen durch grundsätzlich vermeidbare Schäden verschwendet werden. Da diese Ressourcenverschwendungen zulasten der Gesamtheit der Versicherungsnehmer geht, stellt diese Situation ein soziales Dilemma dar [7], in welchem individuelle Rationalität zu gesellschaftlicher Irrationalität führt [8].

Mit der Einführung des iPhones durch Apple im Jahre 2007 und die spätestens dadurch eingeläutete Ära der Smartphones ergeben sich neue Möglichkeiten, diese Moral-Hazard-induzierten Probleme auf Krankenversicherungsmärkten zu lösen.<sup>3</sup> Mit dem Einsatz von mobilen Endgeräten (z. B. Smartphones oder Tablet-PCs) kann versucht werden, an der asymmetrischen Informationsverteilung zwischen Versicherungsgeber und Versicherungsnehmer und somit am Kern der beschriebenen Probleme anzusetzen (vgl. [10] für eine vergleichbare Idee im Finanzsektor). Mit Hilfe von mobilen Endgeräten besteht die Möglichkeit für Versicherte, Informatio-

---

<sup>3</sup> Krankenversicherer können zusätzlich auch Gesundheitsinformationen zur Beeinflussung des „Compliance“-Verhaltens über mobile Endgeräte an Versicherte versenden (vgl. [9]). So könnten z. B. Erinnerungsnachrichten in Bezug auf die Medikamenteneinnahme über spezielle Anwendungsprogramme („Apps“) versendet werden. Da hier keine sensiblen Informationen vonseiten der Versicherten geliefert werden müssen, ist zu erwarten, dass die Teilnahmebereitschaft grundsätzlich hoch ausgeprägt sein müsste.

nen über ihr Gesundheitsverhalten an ihre private Krankenversicherung zu übermitteln und damit den Informationsstand sowie die Kontrollmöglichkeiten und Einflussnahme der Krankenversicherer zu erhöhen. Diese Informationen können sich sowohl auf das allgemeine Präventionsverhalten (vgl. hierzu [11]) als auch das „Compliance“-Verhalten (vgl. hierzu z. B. [12]) beziehen. So ist etwa vorstellbar, dass Versicherte – unter Berücksichtigung von vorgegebenen Trainingsplänen – gegenüber ihrer Krankenversicherung ihre Ausdauersportaktivitäten nachweisen können (Behebung von ex-ante-Moral-Hazard). Im Versicherungsmarkt sind hier bereits Lösungen implementiert. So können Sportler etwa mit Applikation „runtastic“ ihre Sportaktivitäten aufzeichnen und sich über „Social Media“-Plattformen mit anderen Sportlern vernetzen und vergleichen. Im Zusammenhang mit dem Phänomen des ex-post-Moral-Hazard, wie z. B. im Themenkreis „Compliance“ bei der Medikamenteneinnahme, könnte es sinnvoll sein, dass Versicherte Informationen über ihr Gesundheitsverhalten an ihren Krankenversicherer liefern. Zu berücksichtigen ist dabei der aktuelle technische Stand. Eine Datenübertragung muss eindeutig einem Kunden zuweisbar sein sowie ein Mindestmaß an Betrugssicherheit besitzen, um einem Missbrauch vorzubeugen.

Vor diesem Hintergrund kann festgehalten werden, dass grundsätzlich die Möglichkeit besteht, durch den Einsatz mobiler Endgeräte Informationsdefizite aufseiten der Krankenversicherungen zu verringern. Es kann ferner unterstellt werden, dass die durch diesen Abbau von Informationsasymmetrien bedingte Verringerung der Moral-Hazard-Problematik in einer effizienteren Allokation knapper Ressourcen münden wird. Dies liegt darin begründet, dass durch den Einsatz mobiler Endgeräte die Kontrollmöglichkeiten der Krankenversicherungen erhöht werden und Druck auf die Versicherten, sich gesundheitsbewusst zu verhalten, ausgeübt werden kann. Die effizientere Ressourcenallokation wird Krankenversicherungen auch Spielräume eröffnen, mit einer attraktiveren Prämiengestaltung am Markt zu konkurrieren. Seitens der Versicherten muss in diesem Zusammenhang aber die Bereitschaft bestehen, die entsprechenden Informationen an ihren Krankenversicherer zu liefern. Motivationsfördernd können hierbei Anreize pekuniärer (konkret: Übernahme von Fitnessclubgebühren oder Tarifgestaltung im Sinne von „Self-selecting tariffs“) [13]. oder sachlich-materieller Natur (konkret: Bereitstellung mobiler Endgeräte) sein.

## **2.2.2 Themenfelder der zukünftigen Forschung**

Vor dem Hintergrund der aufgezeigten Problematik und des beschriebenen Potentials des Einsatzes mobiler Endgeräte ergeben sich auf drei Feldern zentrale Fragestellungen für die zukünftige Forschung:<sup>4</sup> Während der erste Themenkreis, die „*Analyse der individuellen Akzeptanz des Einsatzes mobiler Endgeräte*“, primär einen verhaltenswissenschaftlichen bzw. soziopsychologischen Fokus hat, können die beiden

<sup>4</sup> An dieser Stelle sei erwähnt, dass Aspekte der technischen Umsetzung, die Relevanz der individuellen Privatheit von Gesundheitsinformationen sowie rechtliche Fragestellung im Zusammenhang mit der Datenschutzproblematik im vorliegenden Beitrag nicht diskutiert werden.

anderen zentralen Fragestellungen (*Ausgestaltung des Einsatzes von mobilen Endgeräten bei Privatversicherten* und *Effekte des Einsatzes von mobilen Endgeräten bei Privatversicherten*) dem Bereich der versicherungswirtschaftlichen Marketing-Implementierung bzw. der gesundheitsökonomischen Evaluation zugeordnet werden.

### 2.2.2.1 Analyse der individuellen Akzeptanz des Einsatzes mobiler Endgeräte

Der Erfolg des Einsatzes mobiler Endgeräte hängt entscheidend von der Bereitschaft der Privatversicherten ab, an entsprechenden Programmen teilzunehmen. In der verhaltenswissenschaftlich oder sozialpsychologisch orientierten Forschung wurden in angrenzenden Themenbereichen bereits Studien zur Akzeptanz neuer Technologien durchgeführt. Demzufolge können bestehende Modelle der individuellen Technologie-Akzeptanz (z. B. [14]) unter Berücksichtigung notwendiger Modifikationen auf den Bereich der Akzeptanz des Einsatzes mobiler Endgeräte bei Privatversicherten übertragen werden. So müssen bei der Analyse der kundenseitigen Akzeptanz von „Mobile services“ neben „Usability“-Aspekten auch (sozial-)psychologische Konstrukte wie z. B. die wahrgenommene Selbstwirksamkeit, die „Mobile device literacy“ oder die „Mobile device fear“ berücksichtigt werden. Ferner sollte mit dem Aspekt des ständigen Zwangs zur Auseinandersetzung mit dem Thema Versicherung im Tagesablauf (z. B. wiederholtes Einloggen per Near-Field-Communication-Technologien im Fitness-Studio) eine weitere Besonderheit der Technologieakzeptanz in dem beschriebenen Szenario berücksichtigt werden.

Interessante Erweiterungen dieser theoretischen Ansätze können ferner in den Themenfeldern der „Technophobia“- sowie „Anti-consumption“-Forschung vorgefunden werden (z. B. [15] oder [16]). Einen besonders interessanten Aspekt stellt auch die Analyse der Akzeptanz des Einsatzes mobiler Endgeräte bei älteren Versicherten dar [15]. Als zentraler Ansatzpunkt für die zukünftige Forschung ergibt sich folgende Forschungsdirektive: *Welche Faktoren beeinflussen die Akzeptanz des Ansatzes von mobilen Endgeräten bei Versicherungsnehmern?*

Wesentliche Merkmale lassen sich hier aus dem Bereich der Assistenz-Dienstleistung herleiten. Mit der „Mobilisierung“ der Versicherung ergeben sich von Kundenseite neue Anforderungen an die Versicherung. Wie bereits in anderen Bereichen seines Lebens vertraut der Kunde zunehmend einer mobilen IT-gestützten Abwicklung von Geschäftsprozessen. Eine Akzeptanz dieser Lösung ist insbesondere dann gegeben, wenn die Produktlösung zuverlässig, intuitiv, einfach und unmittelbar für den Kunden überprüfbar ist.

### 2.2.2.2 Ausgestaltung des Einsatzes von mobilen Endgeräten bei Privatversicherten

Über die Analyse der Akzeptanz aufseiten der Versicherten hinaus, ist auch interessant zu untersuchen, wie der Einsatz mobiler Endgeräte bzw. „Mobile services“

ausgestaltet sein sollte. In einer ersten Stufe könnte hierbei die grundlegende Bedürfnisstruktur der Versicherten aufgedeckt werden. Auf Basis dieser Ergebnisse können relevante Dienstleistungsmerkmale entsprechender „eHealth service“-Konzepte abgeleitet werden (z. B. Art und Umfang der von Versicherten zu offenbarenden Informationen oder alternative Formen der Prämiengestaltung). Darauf aufbauend kann die nutzenoptimale Gestaltung des Einsatzes von „Mobile services“ im Kontext der Vermeidung Moral-Hazard-induzierter Probleme festgelegt werden. Als zentraler Ansatzpunkt für die zukünftige Forschung ergibt sich in diesen Zusammenhängen die nachfolgende Forschungsdirektive: *Wie soll der Einsatz von mobilen Endgeräten im Kontext der Vermeidung des Moral-Hazard-Verhaltens von Versicherten ausgestaltet sein (Dienstleistungseigenschaften und Bündelung von Dienstleistungseigenschaftsausprägungen)?*

Ein Beispielhaftes Szenario an dieser Stelle könnte wie folgt aussehen: Der Versicherungsnehmer einer PKV ist Mitglied in einem Fitness Club. Bei jedem seiner Besuche loggt er sich zu Beginn seines Besuchs mittels Near Field Communication (NFC) auf seinem Smartphone am Standort des Clubs ein. Hierdurch wird die Eindeutigkeit der Handlung sichergestellt. Am Ende seines Besuchs loggt er sich mittels NFC wieder aus. Die Daten über seinen Besuch sendet er per App-Lösung an den Versicherer, der nun aufgrund individueller Daten entscheiden kann, welche Prämien er dem Versicherten anbietet. Beispiel hierfür könnte die Kostenübernahme des vorangegangenen Besuchs sein.

### **2.2.2.3 Analyse der Effekte des Einsatzes von mobilen Endgeräten bei Privatversicherten**

Nachdem die Akzeptanz sowie die nutzenoptimale Ausgestaltung des Einsatzes mobiler Endgeräte analysiert worden ist, kann unter Berücksichtigung von Daten aus Modellprojekten analysiert werden, welche Effekte der oben umrissene Einsatz von mobilen Endgeräten auf gesundheitsökonomische Outcome-Größen haben kann. Hierbei können zunächst „weiche“, verhaltensspezifische Konstrukte, wie z. B. Einstellungsänderungen oder Änderung des erinnerten bzw. tatsächlichen „Compliance“-Verhaltens, fokussiert werden. Die Begründung hierfür liegt u. a. darin, dass der Kausalzusammenhang zwischen dem Einsatz von mobilen Endgeräten und diesen „weichen“ Zielgrößen weniger stark durch exogene Faktoren (z. B. Multimorbidität der Versicherten) mediert bzw. moderiert wird. Ergänzend zu diesen Analysen müssen sodann aber auch „harte“ gesundheitsökonomische Größen (konkrete Kosten- und Qualitätswirkung) in die Betrachtung eingeschlossen werden. Hierbei sollte auch untersucht werden, bei welchen (chronischen) Krankheitsbildern der Lösungsansatz effizient ist. Demzufolge ergibt sich als weiteres Themenfeld für die zukünftige Forschung die Direktive: *Welchen Effekt hat der Einsatz von mobilen Endgeräten auf das Moral-Hazard-Verhalten von Versicherten sowie auf gesundheitsökonomische Outcome-Größen?*

Innerhalb eines abgegrenzten Szenarios (z. B. Großstadtbereich) werden PKV-Versicherten die neuen mobilen Lösungen zur Verfügung gestellt. Sollte innerhalb

des Pilotprojekts ein Effekt nachweisbar belegt werden, so ergeben sich hieraus weitreichende Implikationen. Für die Versicherung können stärker individualisierte Beitragskalkulationen durch die gewonnenen Daten erzeugt werden und durch eine nachweisbare Gesundheitswirkung der Maßnahmen können Kosten im gesamten Gesundheitswesen durch eine Reduktion von Krankheitsfällen minimiert werden. Durch die Verknüpfung von Versicherung und Gesundheit kann nahezu jede beliebige Aktion eines Versicherten bepreist werden, so dass dieser selbst entscheidet, wie viel ihm seine Gesundheit wert ist.

## 2.3 Schlussbemerkungen

Der demographische Wandel sowie der medizinisch-technische Fortschritt führen dazu, dass die Kosten im Gesundheitswesen auch in der Zukunft massiv steigen werden. Wenn es die deutsche Gesundheitspolitik weiterhin nicht schafft, in größerem Umfange Reformen durchzuführen, werden die zu erwartenden Kostensteigerungen in einer Situation resultieren, in denen das momentan bestehende Gesundheitssystem in sich zusammenbrechen kann. Vor diesem Hintergrund haben alle Akteure im Gesundheitswesen die gesellschaftliche Verpflichtung, ihre Kräfte zu bündeln, um diesen Systemkollaps zu verhindern oder die Auswirkungen des Kollapses zumindest abzumildern. Das Phänomen des Moral Hazard stellt in diesen Zusammenhängen eine systemimmanente Problematik des Gesundheitswesens dar, welche in einer ineffizienten Ressourcen-Allokation resultiert und an der Krankenversicherungen sowie die Krankenversicherten gemeinsam im Sinne einer Beseitigung von Ineffizienzen im Gesundheitswesen ansetzen können.

Basierend auf theoretischen Grundlagen hat der vorliegende Beitrag aufgezeigt, dass eine asymmetrische Informationsverteilung zwischen Versicherungsgebern und -nehmern und das daraus resultierende Moral-Hazard-Verhalten zur Verschwendungen von Ressourcen im Gesundheitswesen führt. Unter Berücksichtigung der technologischen Entwicklung und anhand konkreter Szenarien-Entwürfe aus dem PKV-Bereich wird sodann argumentiert, dass der Einsatz mobiler Endgeräte Möglichkeiten eröffnet, die Informationsasymmetrien und somit auch die Moral-Hazard-Anreize zu beseitigen bzw. abzumildern. Mit dieser Idee wird aber sowohl aus der Perspektive der faktischen Anwendung bei Krankenversicherungen als auch der wissenschaftlichen Perspektive Neuland betreten. Insofern umreißt der vorliegende Beitrag drei Themenfelder, auf denen in Zukunft Forschung betrieben werden sollte. Neben der *Analyse der individuellen Akzeptanz des Einsatzes mobiler Endgeräte* sind die Frage nach der *Ausgestaltung des Einsatzes von mobilen Endgeräten bei Privatversicherten* sowie die Untersuchung der *Effekte des Einsatzes von mobilen Endgeräten bei Privatversicherten* in diesem Zusammenhang zentral. Es ist zu erwarten, dass diese Forschungen einen wesentlichen Beitrag zur Behebung von Ineffizienzen im Gesundheitswesen und somit zur gesellschaftlichen Nachhaltigkeit liefern können.

## Literaturverzeichnis

1. Cohen A, Siegelman P (2010) Testing for Adverse Selection in Insurance Markets. *J Risk Insurance* 77(1):39–84
2. Guinnane TW, Streb J (2011) Moral Hazard in a Mutual Health Insurance System: German Knappschaften, 1867–1914. *J Econ His* 71(1):70–104
3. Strassl W (1988) Externe Effekte auf Versicherungsmärkten. Eine allokativtheoretische Begründung staatlicher Regulierung; zugl.: München, Univ., Diss., Tübingen, Mohr
4. Arrow KJ (1970) Essays in the theory of risk-bearing. Amsterdam, London, North-Holland
5. Zweifel P, Manning WG (2000) Moral Hazard and Consumer Incentives in Health Care. In: Culyer AJ, Newhouse JP (eds) *Handbook of Health Economics*, vol 1, Amsterdam, Elsevier, pp 409–459
6. Schulenburg J-M, Greiner W (2007) *Gesundheitsökonomik*, 2. Auflage, Tübingen, Mohr Siebeck
7. Weber JM, Kopelman S, Messick DM (2004) A Conceptual Review of Decision Making in Social Dilemmas: Applying a Logic of Appropriateness. In: *Personality and Social Psychology Review* 8(3):281–307
8. Kollock P (1998) Social Dilemmas: The Anatomy of Cooperation. In: *Annual Review of Sociology* 24(1):183–214
9. Susilo W, Kin TW (2007) Securing Personal Health Information Access in Mobile Healthcare Environment through Short Signature Schemes. In: *Int J Mobile Commun* 5(2):215–224
10. Ergincan Y (2005) Protecting Investors and Lessening Moral Hazard in Capital Markets by Using Mobile Communication and Internet Technologies and the Role of Central Registry Agency of Turkey. In: *J Econ Soc Res* 7(1):31–50
11. Dryfoos JG (1991) Preventing High-risk Behavior, in: *American Journal of Public Health* 81(2):157–158
12. Bowmann D, Heilman CM, Seetharaman PB (2004) Determinants of Product-use Compliance Behavior. *J Marketing Res* 41(3):324–338
13. Jeon D, Menicucci D (2005) Optimal second-degree price discrimination and arbitrage: on the role of asymmetric information among buyers. *RAND J Econ* 36(2):337–360
14. Hu PJ, Chau PYK, Sheng ORL, Kar YT (1999) Examining the Technology Acceptance Model Using Physician Acceptance of Telemedicine Technology. *J Management Inf Sys* 16(2):91–112
15. Hogan M (2006) Technophobia Amongst Older Adults in Ireland. *Irish J Management* 27(1):57–77
16. Lefebvre A (2000) All We Need to Fear Is Fear Itself: Overcoming the Internet Resistance of Child Psychiatrists. *CyberPsychology & Behavior* 3(6):951–957

---

# Kapitel 3

## Smart Automotive Apps für kundenorientierte Dienstleistungen im Fahrzeug

Tobias Schlachtbauer, Gerhard Stanzl, Maximilian Pühler, Michael Schermann und Helmut Krcmar

**Zusammenfassung** Smart Automotive Apps, d. h. intelligente IT-basierte Anwendungen im tertiären Anwendungsbereich des Autofahrers, spielen in der Automobilindustrie eine zunehmend wichtigere Rolle für die Wertschöpfung. Jedoch sind die Fachexperten der unterschiedlichen Anwendungsdomänen nur selten in der Lage ihre Visionen, Anforderungen und Einschränkungen vollständig zu explizieren, wenn es um die Entwicklung innovativer Softwareanwendungen geht. Die besonderen Herausforderungen in der Automotive-Domäne, aufgrund der hohen Anforderungen an die Software im Fahrzeug, erhöhen zudem die Komplexität in der Entwicklung von Smart Automotive Apps. Hierbei kann die schnelle Entwicklung und Evaluierung von Prototypen mit Hilfe der graphischen Modellierungssprache ASML (Automotive Services Modelling Language) zusammen mit der Prototyping-Plattform HIMEPP (Highly Integrated Modular Embedded Prototyping Platform) einen entscheidenden Vorteil bringen. Dank der kurzen Umsetzungszeit und großen Flexibilität für die Entwicklung innovativer Prototypen kann mit der Kombination aus ASML und HIMEPP die Kluft zwischen dem domänenspezifischen Wissen der Fachexperten auf der einen Seite und den technischen Kenntnissen der Automotive-App-Entwicklern auf der anderen Seite geschlossen werden. Dieser Ansatz bietet somit die Möglichkeit, auch Fachexperten ohne technische Kenntnisse effektiv

---

Tobias Schlachtbauer  
Technische Universität München, Lehrstuhl für Wirtschaftsinformatik, Garching b. München,  
E-mail: Tobias.Schlachtbauer@in.tum.de

Gerhard Stanzl  
Audi AG, Geschäftsentwicklung IT, Ingolstadt, E-mail: Gerhard.Stanzl@audi.de

Maximilian Pühler  
Audi AG, Audi IT Open Innovation, Ingolstadt, E-mail: Maximilian.Puehler@audi.de

Michael Schermann  
Technische Universität München, Lehrstuhl für Wirtschaftsinformatik, Garching b. München  
Helmut Krcmar  
Technische Universität München, Lehrstuhl für Wirtschaftsinformatik, Garching b. München,  
E-mail: krcmar@in.tum.de

---

mit in die Definition und Entwicklung von technisch machbaren Smart Automotive Apps einzubinden. Insgesamt ermöglicht das hier vorgestellte modellbasierte Prototyping eine sehr effiziente und effektive Entwicklung innovativer Smart Automotive Apps.

### 3.1 Einleitung

Mobile Dienste haben in den letzten Jahren ihren Einzug in nahezu alle Lebensbereiche gefunden und prägen somit den persönlichen und beruflichen Alltag ihrer Benutzer immer stärker. Dieser Trend ist besonders stark im Bereich der Smartphones wahrnehmbar. Doch auch im Automobilbereich nehmen IT-basierte mobile Anwendungen, vor allem im tertiären Anwendungsbereich des Fahrers durch sog. Smart Automotive Apps, einen immer größeren Stellenwert ein. Unter Smart Automotive Apps verstehen wir hierbei IT-basierte Anwendungen im tertiären Aufgabenbereich des Fahrers, die auf der im Fahrzeug verbauten IT-Plattform ausgeführt werden. Diese Anwendungen werden als „smart“ bezeichnet, da sie den Fahrer durch ihre interne Logik bei der Lösung seiner Aufgaben und Probleme unterstützen und doch eine einfache und sichere Verwendung erlauben.

Der Einzug von tertiären IT-basierten Anwendungen in das Fahrzeug wird maßgeblich durch die Möglichkeiten der ubiquitären Verfügbarkeit von Rechenleistung und mobilem Breitbandinternetzugang begünstigt. Das Fahrzeug entwickelt sich hierbei zunehmend zu einem weiteren internetfähigen mobilen Endgerät und stellt die Automobilhersteller hierdurch vor neue Herausforderungen [1]. Jedoch ergeben sich hieraus auch völlig neue und bedeutende Möglichkeiten der Wertschöpfung für die Automobilindustrie. Damit die Potenziale der smarten Automotive Apps in dem von hartem Wettbewerb gekennzeichneten Automobilmarkt für die Automobilhersteller zum Tragen kommen, ist ein gründliches Verständnis der Anforderungen sowie der Nutzungsszenarien erforderlich. Außerdem muss der vom Kunden tatsächlich wahrgenommene Wert einer solchen App frühzeitig evaluiert und in der Entwicklung berücksichtigt werden.

Es ist daher nicht überraschend, dass sich die Automobilindustrie Vorgehensmodelle aus Industrien mit ähnlichen Mustern, z. B. der Mobilfunkindustrie, zu eigen macht und die Idee des Prototyping aufgreift. Prototyping ermöglicht es Entwicklern, die wichtigen Anforderungen durch die Ausführung und das Testen von ausgewählten Funktionalitäten einer Smart Automotive App herauszufinden. Das Ziel des Prototyping ist es dabei, Kunden bereits in frühen Phasen des Service Engineering zu integrieren und ein gemeinsames Verständnis für die App unter den Stakeholdern zu schaffen [2]. Untersuchungen zeigen, dass durch Prototyping das Risiko von nicht erfüllten Anforderungen des Nutzers erheblich reduziert wird. In diesem Zusammenhang steigt durch Prototyping folglich auch das Potenzial für die Entwicklung erfolgreicher IT-basierter Mehrwertanwendungen in Form von Smart Automotive Apps.

Insbesondere für das Services Engineering innovativer Anwendungen in der Automotive-Domäne mangelt es jedoch bisher den Stakeholdern an der Fähigkeit

ihre Visionen, Anforderungen und Einschränkungen zu explizieren. Hierbei mangelt es ihnen einerseits an dem notwendigen IT-Wissen, d.h. an dem Verständnis bzgl. der Potenziale und Grenzen von IT-Systemen. Dies ist jedoch für die Entwicklung innovativer Smart Automotive Apps zwingend notwendig. Andererseits sind sie Experten in ihrer beruflichen und persönlichen Domäne und besitzen daher das implizite Wissen über das, was eine App können sollte. Ihnen fehlt es jedoch an einer Möglichkeit, dieses Wissen auszudrücken und in eine klare Vorstellung über die App zu verwandeln. Eine Konfrontation der Stakeholder mit einer möglichen Ausprägung für eine App führt zwangsläufig zu einem neuen umfassenderen Verständnis der eigenen Anforderungen sowie der technischen Möglichkeiten [3]. Daher ist für die Festlegung der wesentlichen Servicebestandteile eine häufige Iteration zwischen dem Sammeln und Evaluieren von Anforderungen erforderlich. Insbesondere sollten die Stakeholder in die Lage versetzt werden, ihre Vision und Anforderungen in Konzepten ihrer Domäne auszudrücken, ohne dabei ein Verständnis bzgl. der technischen Potenziale und Grenzen besitzen zu müssen.

Dieser Herausforderung kann mit Hilfe des Modellierungsansatzes ASML (Automotive Services Modelling Language) für Smart Automotive Services begegnet werden. Hierzu wurde am Automotive Services Lab des Lehrstuhls für Wirtschaftsinformatik der Technischen Universität München in Zusammenarbeit mit der Abteilung für IT im Fahrzeug der Audi AG, Ingolstadt, ein integrierter Software Development Kit (SDK) entwickelt, welcher drei wesentliche Aspekte in sich vereint:

- Modularität: Smart Automotive Apps können als ein Bündel von standardisierten Komponenten verstanden werden. Diese Komponenten erfüllen dabei jeweils eine ganz bestimmte wiederverwendbare Aufgabe wie der Empfang eines GPS-Signals, der Aufbau einer mobilen Internetverbindung oder Spracheingabe und -ausgabe.
- Produktivierung: Dies bedeutet in diesem Zusammenhang das Lernen aus verschiedenen existierenden individuellen Services [4, 5]. Im Gegensatz zu bestehenden Entwicklungsprozessen für Services fordert Produktivierung die Analyse bestehender Lösungen und die Entwicklung neuer Services durch die Identifikation von Gemeinsamkeiten unterschiedlicher implementierter Lösungen, Prozesse und Strukturen.
- Integration: Um die Kommunikation zwischen den Experten der Anwendungsdomäne und den Entwicklern zu erleichtern, wurde eine enge Integration der technischen Implementierung der Komponenten einer Smart Automotive App mit deren visueller Repräsentation eingeführt und umgesetzt.

Durch diese drei Charakteristika des SDK – ASML/HIMEPP – wird es auf der einen Seite nicht-technischen Stakeholdern ermöglicht, sich auf die wichtigen Eigenschaften der Ausgestaltung einer Smart Automotive App zu konzentrieren, ohne sich dabei um die technische Implementierung kümmern zu müssen. Auf der anderen Seite ist es den technischen Stakeholdern immer noch möglich, die Komplexität und den Aufwand für die Umsetzung des Services abzuschätzen. Hierdurch ermöglicht das SDK einen evolutionären Ansatz zur Entwicklung innovativer Smart Automotive

Apps, ohne dabei die grundlegende Idee des Prototyping, nämlich ein Artefakt in der gewünschten Anwendungsdomäne zu schaffen, zu kompromittieren. Denn die mit dem SDK erstellten Modelle sind direkt in der IT-Infrastruktur des Fahrzeugs einsetzbar und können hierdurch auch gleich im Anwendungskontext evaluiert und weiterentwickelt werden.

## 3.2 Herausforderungen bei der Gestaltung innovativer Smart Automotive Apps

### 3.2.1 Hintergrund

In den letzten Jahren hat sich gezeigt, dass Smart Automotive Apps zunehmend an Bedeutung für die Automobilindustrie gewinnen. Es wurde hierbei deutlich, dass die Wertschöpfung bei modernen Systemen maßgeblich auf Software basiert [6]. Zudem verursacht die Softwareentwicklung mehr als 50 % der gesamten Entwicklungskosten [7]. Die Integration von Smart Mobile Apps ins Fahrzeug wird zudem durch die zunehmende Effizienz von Systemressourcen wie Rechenleistung, Speicherkapazität oder hochauflösenden Displays ermöglicht. Hierdurch werden erweiterte Dienste und Anwendungen in modernen Fahrzeugen möglich, die den Anwender bei der Erfüllung seiner Aufgaben unterstützen. Auch die bevorstehende flächendeckende Verfügbarkeit von mobilen Datenverbindungen wie UMTS (3G) und LTE (4G) fördert diesen Trend. Ein prominentes Beispiel hierfür ist das aktuelle Audi Multimedia Interface (MMI) im Rahmen der Audi connect genannten Vernetzungsstrategie.

Geprägt durch die ubiquitäre Verfügbarkeit von Technologien im privaten und beruflichen Umfeld der Kunden steigen auch die Erwartungen an die Systeme in modernen Fahrzeugen. Beispielsweise führt der Erfolg und die Verbreitung von Smartphones wie Apples iPhone und den damit verbundenen Möglichkeiten zu einer erweiterten Erwartungshaltung der Kunden hinsichtlich der Verfügbarkeit des mobilen Zugangs zum Internet und den Anwendungen im tertiären Bereich eines modernen und gut ausgestatteten Automobils. Dabei sind die Apps im Fahrzeug, wie in Abb. 3.1 ersichtlich ist, nur ein Teilaspekt der von einem vernetzten Fahrzeug abdeckbaren Möglichkeiten.

Smart Automotive Apps werden jedoch nicht nur bedingt durch die Erwartungshaltung der Kunden getrieben, sondern erfahren auch aufgrund der aktuellen Herausforderungen im Automobilmarkt eine zunehmende Notwendigkeit. Beispielsweise werden, im Zuge der Diskussionen über umweltfreundliche Alternativen zum Verbrennungsmotor aufgrund des Klimawandels, Arbeiten an innovativen Fahrzeugen im Bereich der Elektromobilität vorangetrieben. Bedingt durch die Herausforderungen von batterieelektrischen Lösungsansätzen, welche gegenüber herkömmlichen Verbrennungsmotoren aktuell noch verschiedenen Einschränkungen hinsichtlich Flexibilität und Reichweite aufweisen, wird über neue Mobilitätskonzepte und



**Abb. 3.1** Smart Automotive Apps als Teilaspekt des vernetzten Fahrzeugs

ein verändertes Nutzungsverhalten im Zusammenhang mit Elektrofahrzeugen nachgedacht. Innovative multimodale Mobilitätskonzepte, verbunden mit Fahrgemeinschaften und hybriden Lösungen, können jedoch nur in Verbindung mit den entsprechenden IT-Lösungen im und um das Fahrzeug realisiert werden.

Jedoch ist die Entwicklung von Smart Automotive Apps und den zugehörigen IT-Lösungen im Backend weder in der Entwicklungs- noch in der Produktionsphase preiswert: Selbst heute belaufen sich die Kosten für die Softwareentwicklung in der Automotive-Domäne bereits auf 20 % bis 40 % der gesamten Produktionskosten [8]. Software ist dabei zu einem unverzichtbaren Teil des Produktes selbst geworden, da die Funktionalität, welche durch Software realisiert wird, nicht durch Hardware ersetzt werden kann. Darüber hinaus stützt sich die Wertschöpfung von modernen Produkten vermehrt auf die durch Software bereitgestellten und ermöglichten Funktionen [6].

### 3.2.2 Sicherstellung der Machbarkeit von Smart Automotive Apps

Im Vergleich zur Entwicklung von Smart Mobile Apps in anderen Service-Domänen ist die offensichtlichste Herausforderung für die Entwicklung von Smart Automotive Apps, die Anwendungen so zu gestalten, dass sie auch im Nutzungsszenario während der Fahrt verwendet werden können, ohne den Fahrer von seiner primären Aufgabe – dem Fahren – abzulenken. Neben dem Fahrer existieren mit dem Beifahrer sowie den Passagieren noch weitere Benutzergruppen mit eigenen Anforderungen und Einschränkungen bzgl. der Nutzung derselben Anwendungen im Fahrzeug. Eine Smart Automotive App muss daher den Anforderungen und Einschrän-

kungen je nach Nutzungskontext (Fahren oder Parken) und Anwender (Fahrer oder Beifahrer/Passagier) entsprechen. Im Fokus steht hierbei der Fahrer. Bezogen auf diesen muss die Anwendung besonderen Designrichtlinien für eine hohe intuitive Usability folgen, so dass die Nutzung nur ein geringes Maß an Aufmerksamkeit des Anwenders erfordert. Daneben ist für einen IT-versierten Beifahrer oder Passagier auch eine umfangreiche graphische Benutzeroberfläche möglich. Hierdurch erhöht sich die Komplexität der Anwendungen, da sie u. U. zur gleichen Zeit die Anforderungen verschiedener Nutzergruppen erfüllen und zusätzlich noch deren jeweilige Einschränkungen berücksichtigen müssen.

Neben den Einschränkungen, die sich aus dem Nutzungskontext einer Smart Automotive App ergeben, existieren auch Limitationen aufgrund der vorhandenen Hardware im Fahrzeug. Die Apps laufen im Fahrzeug auf eingebetteten Systemen. Diese verfügen nur über begrenzte Rechenleistung und sind zudem an die langen Lebenszyklen des Fahrzeugs gebunden. Somit sind die Apps im Fahrzeug stärkeren Limitationen bzgl. der verfügbaren Hardware unterworfen als vergleichbare Anwendungen im Bereich der Unterhaltungselektronik oder dem Mobilfunk. Auch im Hinblick auf die Nutzung und Interaktion mit einer Smart Automotive App existieren durch die begrenzten Ein- und Ausgabemöglichkeiten weitere Einschränkungen. Der Innenraum von Fahrzeugen erlaubt in der Regel nur den Einbau relativ kleiner Displays für die Anzeige. Außerdem sind für die Interaktion mit Anwendungen im Fahrzeug im Allgemeinen Funktionstasten und Dreh-Drücksteller üblich. In modernen Fahrzeugen werden diese Eingabemöglichkeiten zunehmend noch durch Spracheingabe und -ausgabe ergänzt. Insgesamt sind jedoch die Interaktionsmöglichkeiten des Fahrers mit einer Smart Automotive App, insbesondere während der Fahrt, stark begrenzt. Aus diesem Grund scheitert die schlichte Übertragung von Apps aus anderen Bereichen in die Automotive-Domäne an so grundlegenden Problemen wie dem Fehlen von äquivalenten Eingabemöglichkeiten wie Tastatur und Maus für Computer.

Des Weiteren sind Smart Automotive Apps aufgrund von technischen, designbedingten und sicherheitsbezogenen Restriktionen eng integrierte Kombinationen von Hardware, Software und externen Diensten [9]. Diese aktuell vorhandene enge Verzahnung von Hardware und Software schränkt durch die sehr unterschiedlichen Lebenszyklen dieser beiden Komponenten die Flexibilität im Bezug auf die Realisierung neuartiger softwarebasierter Anwendungen stark ein. Der Lebenszyklus eines Fahrzeugs samt der darin verbauten IT beträgt durchschnittlich bis zu 20 Jahre. Im extremen Gegensatz dazu weisen Software- und Servicekomponenten meist eine relativ kurze Lebensdauer auf, in der sie aktuell sind, bevor sie durch eine verbesserte und erweiterte Version abgelöst werden. Jedoch erwarten die Kunden, ungeachtet der langen Lebensdauer des Fahrzeugs, die neusten Technologien in ihrem Fahrzeug vorzufinden und dort nutzen zu können. Ebenso wie sie es von ihren persönlichen mobilen Endgeräten gewohnt sind, soll auch das Fahrzeug dem aktuellen Stand der Technik entsprechen.

Der Prozess von einer Idee für eine neue App bis zu deren Verfügbarkeit im Fahrzeug ist jedoch meist mit hohen Risiken verbunden, beansprucht viele Ressourcen und ist zeitaufwändig. Derzeit laufen auch bereits Bestrebungen verschiede-

ner Automobilhersteller, den Problemen durch die enge Kopplung von Hardware- und Softwarekomponenten durch die Schaffung einer Serviceplattform zu begegnen. Mit der Erhöhung der Flexibilität wird jedoch auf der anderen Seite die Gewährleistung, dass die Anwendungen im Fahrzeug den strengen Design- und Sicherheitsrichtlinien in der Automotive-Domäne genügen, erschwert.

Zusammenfassend lässt sich festhalten, dass erfolgreiche Smart Automotive Apps die speziellen Anforderungen des jeweiligen Nutzungsszenarios erfüllen müssen. Außerdem soll die App, innerhalb der begrenzten Interaktionsmöglichkeiten des Anwenders, einen positiven Beitrag zum Fahrerlebnis leisten. Darüber hinaus muss eine Smart Automotive App eine starken Business Case aufweisen, damit sich der Ressourceneinsatz während des aufwändigen und risikobehafteten Prozesses von der Idee bis zur Verfügbarkeit der App für den Kunden im Fahrzeug auch lohnt. Um dieses Risiko in der Entwicklung von Smart Automotive Apps weitestgehend einzugrenzen ist es für die Automobilhersteller entscheidend wichtig, ein gründliches Verständnis über die Wertversprechen der Ideen für diese Services im Nutzungskontext – dem Fahren – zu entwickeln.

### ***3.2.3 Die Nutzenpotenziale von Smart Automotive Apps verstehen***

Wie oben bereits erwähnt, bieten erfolgreiche Smart Automotive Apps einen Mehrwert für den Kunden und ergänzen das Fahrerlebnis. Gleichzeitig ermöglichen sie es auch dem Serviceanbieter, von diesem Mehrwert zu profitieren. Damit die Nutzenpotenziale dieser Anwendungen auch wirksam zum Einsatz kommen, ist eine Integration von technischen und nicht-technischen sowie internen und externen Stakeholdern notwendig.

Im Allgemeinen wirken interne Stakeholder als „Gate Keeper“ bei der Entwicklung von Smart Automotive Apps. Während sich Stakeholder aus den technischen Bereichen mit der Bewertung der Umsetzbarkeit einer Smart Automotive App beschäftigen, evaluieren die Stakeholder aus den nicht-technischen Bereichen, welche strategischen und wirtschaftlichen Auswirkungen mit dieser App verbunden sind. Auch die externen Stakeholder lassen sich meist dem nicht-technischen Bereich zuordnen, z. B. kann es sich hierbei um Kunden handeln, die Ideen für neue Services vorschlagen und mögliche Umsetzungen ihren Wünschen und Anforderungen entsprechend bewerten. Laut Erkenntnissen aus der Open-Innovation-Forschung führt die Integration von externen Stakeholdern in frühen Phasen des Entwicklungsprozesses zu besonders innovativen Services [10]. Als weiterer externer Stakeholder haben auch Regulierungsbehörden einen Einfluss auf die Ausgestaltung einer Automotive App und somit auch auf deren Nutzenpotenziale, indem sie durch gesetzliche Vorgaben, z. B. Sicherheitsvorschriften, die möglichen Nutzungsszenarien einschränken.

Die durch Smart Automotive Apps zusätzlich realisierbaren Nutzenpotenziale lassen sich dabei in einen Basisnutzen (Automatisierung, Flexibilität, Vernetzung) und einen Zusatznutzen (Filterung, Simplexity, Personalisierung) unterscheiden.

Der Basisnutzen stellt hierbei zum einen das ihm inhärente Nutzenpotenzial dar und bildet zum anderen die Voraussetzung für die Realisierung weiterer Nutzeneffekte. Der Zusatznutzen weist für die Bewertung von Smart Automotive Apps eine besondere Bedeutung auf. In seiner Gesamtheit bezeichnet er nämlich eine neue Form der Informations- und Wissensnutzung: Der Nutzer muss Informationen nicht mehr aktiv suchen, sondern er kann die situationsbedingt erforderlichen Informationen direkt selbst „erleben“. Durch den Einsatz von innovativen, anwenderzentrierten Smart Automotive Apps wird somit eine völlig neue Form der Informations- und Wissensnutzung im Fahrzeug ermöglicht.

Zusammenfassend lässt sich sagen, dass, um die Nutzenversprechen und den realisierbaren Mehrwert einer Smart Automotive App zu verstehen, eine integrierte Darstellung von technischen und nicht-technischen Eigenschaften dieser Anwendung erforderlich ist. Beispielsweise konzentrieren sich Kunden auf den Mehrwert, den ihnen die Anwendung bietet, während sich die Entwickler mit der Sicherstellung der technischen Machbarkeit beschäftigen. Darüber hinaus müssen Automobilhersteller die Einhaltung von gesetzlichen Vorschriften sicherstellen.

### ***3.2.4 Integriertes Prototyping zur Validierung der Nutzenversprechen von Smart Automotive Apps***

Laut Brooks „the hardest single part of building a software system is deciding precisely what to build. [...] No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later“ [11]. Daher hängt die Entwicklung einer erfolgreichen Smart Automotive App maßgeblich von der Fähigkeit ab, ein gemeinsames Verständnis zwischen Anwendern und Entwicklern zu schaffen. Darüber hinaus müssen insbesondere sehr innovative Funktionen frühstmöglich auf die Einhaltung von geltenden Sicherheitsvorschriften getestet werden. In Literatur und Praxis wird hierfür eine Vielzahl an Ansätzen zur Lösung dieser Herausforderungen vorgeschlagen [12–14].

Hierbei ist ein umfassendes Verständnis der Anforderungen des Kunden zwingend notwendig. Jedoch gilt: „stakeholders seldom understand their requirements fully at the beginning of a project. They often lack a clear vision of what the system should do and change their minds during development. In general, stakeholders only become surer about what they want when they see a system that does not exhibit the necessary features“ [3]. Deshalb sollen Services in einem engen Dialog mit dem Kunden entwickelt werden [15]. Für die Validierung des gemeinsamen Verständnisses aller Stakeholder bieten Prototypen eine gute Möglichkeit. Deshalb wird ein methodischer Ansatz zur evolutionären Systementwicklung vorgeschlagen [2].

Die Schaffung eines gemeinsamen Verständnisses zwischen den fachlichen und technischen Stakeholdern ist in der Automotive-Domäne von großer Bedeutung.

Aufgrund der besonderen Einschränkungen, wie z. B. die Auswirkungen auf die Sicherheit oder die spezielle Zielumgebung (Fahrzeug), ist die Definition valider Anforderungen für die Entwicklung von tertiären IT-basierten Anwendungen im Fahrzeug besonders zentral.

Die Einigkeit von Forschung und Praxis sowie das Wissen um die Notwendigkeit des Prototyping für die Entwicklung guter und erfolgreicher Smart Automotive Apps ändert jedoch nichts an der Tatsache, dass die Entwicklung solcher Prototypen mit einem hohen zeitlichen und finanziellen Aufwand verbunden ist. Dabei erhöhen sich diese Aufwendungen mit jeder durchlaufenen Iteration [16]. Um mit den Stakeholdern in Dialog über den Prototypen treten zu können müssen die Entwickler diesen in das Fahrzeug integrieren. Das in der Evaluation eingesetzte Feedback muss anschließend wieder in Änderungen an den zur Realisierung des Prototyps verwendeten Komponenten (Hardware, Software, Services) übersetzt werden.

Insgesamt lässt sich hier festhalten, dass Prototypen eine Schlüsselrolle für die Entwicklung von Smart Automotive Apps einnehmen, da sie die Schaffung und Validierung eines gemeinsamen Verständnisses aller Stakeholder über die Anforderungen und Nutzenpotenziale der zu realisierenden Anwendung ermöglichen. Jedoch ist ein flexibler Ansatz erforderlich, wenn Prototypen sowohl für die Ideenfindung und Anforderungserhebung als auch für die Evaluation und Bewertung verwendet werden sollen. Dies ist allerdings notwendig um eine integrierte Iteration zwischen dem Domänenwissen der Fachexperten und dem technischen Wissen der Entwickler zu ermöglichen.

### **3.3 Die Modellierungssprache ASML – Automotive Services Modelling Language**

In diesem Kapitel stellen wir die Grundlagen und Besonderheiten der Automotive Services Modelling Language (ASML) vor. ASML wurde speziell dafür entwickelt, die oben beschriebenen Herausforderungen bei der Entwicklung von Smart Automotive Apps zu lösen. Das Ziel von ASML ist es, auch nicht-technischen Stakeholdern die Möglichkeit zu geben ihre Visionen und Anforderungen so auszudrücken, dass hierauf basierend anschließend die technische Umsetzung erfolgen kann. In diesem Zusammenhang unterstützt ASML das Prototyping, da die modellierten Anwendungen mit nur wenig bis gar keinem Implementierungsaufwand in der Zielumgebung des Fahrzeugs zur Ausführung gebracht werden können. Mit ASML können die Ideen für eine Smart Automotive App einfach auf eine Reihe von vordefinierten Entitäten projiziert und mit Hilfe der bereits vorhandenen Module umgesetzt werden. Zudem können bei Bedarf neue Module für eine noch nicht vorhandene Funktionalität auf eine einfache Art und Weise selbst modelliert und sofort in das Modell der Anwendung integriert werden.

### 3.3.1 Designprinzipien und Modellelemente von ASML

Die domänen spezifische Modellierungssprache ASML begegnet den Herausforderungen der Softwareentwicklung für smarte, tertiäre Anwendungen im Fahrzeug durch drei zentrale Designprinzipien:

- Bricks für die domänen spezifische Repräsentation von Komponenten,
- Stubs zur implementierbaren Repräsentation von Komponenten und
- Composites als Erfahrungsmanagement.

Der Notwendigkeit einer domänen spezifischen Repräsentation wird durch die Einführung von Bricks (s. Abb. 3.2) entsprochen. Diese versetzen Stakeholder in die Lage, ihr domänen spezifisches Wissen zu nutzen und somit ein besseres Verständnis der Lösung zu erlangen. Bricks sind die Grundbausteine einer Smart Automotive App. Es handelt sich hierbei um diskrete Entitäten, die verschiedene Funktionen unterschiedlichster Granularität einer Smart Automotive App repräsentieren. Diese Funktionen können von der grafischen Darstellung einer Liste von Elementen bis hin zum Versenden einer E-Mail reichen. Innerhalb eines Modells dürfen dabei mehrere Bricks des gleichen Typs verwendet werden. Eine Ausnahme bilden hier die sogenannten Singleton Bricks, die üblicherweise für technische Elemente stehen, wie z. B. eine GPS-Positionierung. Diese werden durch spezifische Hardwareelemente realisiert und können daher auch nur einmal instanziert werden. Bricks werden mit Hilfe von Streams miteinander verbunden. Ein Stream beschreibt den Kontrollfluss zwischen genau zwei Bricks und verbindet diese. Neben der Beschreibung des Kontrollflusses dienen Streams auch zur Steuerung des Datenflusses zwischen Bricks.

Die Implementierbarkeit der Repräsentation wird durch das Element des Stub (s. Abb. 3.2) realisiert. Hierbei liegt das Augenmerk auf der Umwandlung do-



Abb. 3.2 Die Modellelemente Brick (*links*) und Stub (*rechts*)

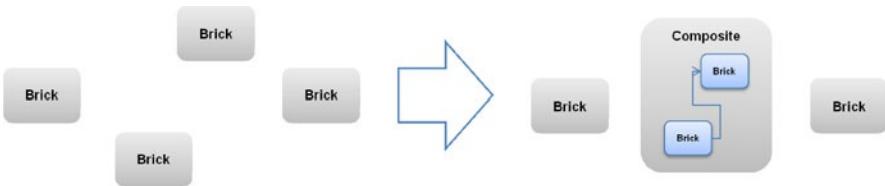


Abb. 3.3 Schematische Darstellung eines Composite

mainspezifischer Anforderungen in IT-Artefakte, die durch technische Stakeholder umgesetzt werden können. Da anzunehmen ist, dass Smart Automotive Apps von Stakeholdern mit großem domainspezifischen Fachwissen, aber nur geringem technischen Wissen definiert werden, ist es die Aufgabe von Stubs, diese nicht-technischen Stakeholder im Prozess zur Festlegung der erforderlichen Spezifikation für eine Umsetzung zu unterstützen. Umfassen Bricks also sowohl das domänen-spezifische Fachwissen als auch die technologischen Grundlagen einer Umsetzung, so stellen Stubs lediglich das domänen-spezifische Fachwissen dar. Durch die Möglichkeit, domänen-spezifische Konzepte in Form von Stubs zu spezifizieren, wird der Aufwand zur Problemlösung für die einzelnen Stakeholder wesentlich reduziert, da sie sich nur noch mit einem Element, dem Stub, und nicht mehr mit einem komplexen Konzept auseinandersetzen müssen. Dies ermöglicht eine Reduktion der kognitiven Belastung und führt auf der einen Seite dazu, dass Stakeholder ihre Vorstellungen leichter im Rahmen des Modells realisieren können. Auf der anderen Seite können die Konzepte einfacher in das domänen-spezifische Framework integriert werden.

Mit dem Composite wird der Herausforderung des notwendigen Erfahrungsmanagements begegnet. Sie stellen eine Methode für eine umfassende Artikulation von modelliertem domänen-spezifischem Fachwissen dar. Composites ermöglichen es Stakeholdern, verschiedenste Bricks beliebig miteinander in einer neuen eigenständigen Komponente zu kombinieren (s. Abb. 3.3). Die in einem Composite zusammengefassten Bricks lassen sich einfach als Baustein in einer anderen Anwendung wiederverwenden. Durch diese Produktivierung erprobter Anwendungen kann so die Fähigkeit zur Problemlösung maßgeblich verbessert werden, da bei der Gestaltung zukünftiger Anwendungen dieses erprobte Wissen in Form von Composites einfach wiederverwendet werden kann. Dadurch müssen sich die Entwickler jeweils nur mit den aktuellen Problemen beschäftigen und können bereits gelöste Aspekte einfach wiederverwenden.

Die Automotive Services Modelling Language besteht aus drei grundlegenden Gestaltungselementen: Bricks dienen dazu, domänen-spezifisches Wissen in einer von technischen Grundlagen abstrahierten Form zu repräsentieren und so verfügbar zu machen. Zu den Bricks gehören noch Streams, die funktionale und logische Übergänge von einem Brick zum nächsten darstellen. Schließlich erweitern Composites die Modellierungssprache im Sinne einer evolutionären Weiterentwicklung. Durch die Komposition komplexer Smart Automotive Apps werden neue Bausteine geschaffen, die in zukünftigen Services wiederverwendet werden können. Um mit

---

der Fragestellung gänzlich neuer Herausforderungen umzugehen, werden Stakeholder durch die Möglichkeit, eigene Stubs zu definieren, in die Lage versetzt, neue, domänen spezifische Konzepte zu nutzen.

### 3.4 Entwicklung von Smart Automotive Apps mit ASML

Durch die Nutzung von ASML ist es nun möglich, mit geringem zeitlichen und finanziellen Aufwand Prototypen von Smart Automotive Services zu erstellen. Für die Ausführung wird hierbei das domänen spezifische Framework Highly Integrated Modular Embedded Prototyping Platform (HIMEPP) von Hoffmann (2010) verwendet, mit dem fahrzeugspezifische und alternative Bedien- und Funktionskonzepte realisiert werden. Die Kombination von ASML und HIMEPP erlaubt es, die Prototypen schnell und einfach im Fahrzeug auszuführen und zu evaluieren. Aufgrund der Flexibilität dieses Ansatzes kann zudem das Feedback aus der Evaluation schnell in den Prototypen eingearbeitet werden. Außerdem ist bei diesem Prozess nicht nur die Möglichkeit der Zusammenarbeit von technischen und nicht-technischen Stakeholdern gegeben, sondern auch ausdrücklich gewünscht.

Im Folgenden wird nun eine Beispieldarstellung vorgestellt, die in der Zusammenarbeit zwischen Mitarbeitern des Automotive Services Lab und der Audi AG entstanden ist. Hierbei handelt es sich um den Personal Car Assistant, einer Anwendung zur Unterstützung mobiler Arbeit im Fahrzeug.

#### 3.4.1 Beispieldarstellung: Personal Car Assistant

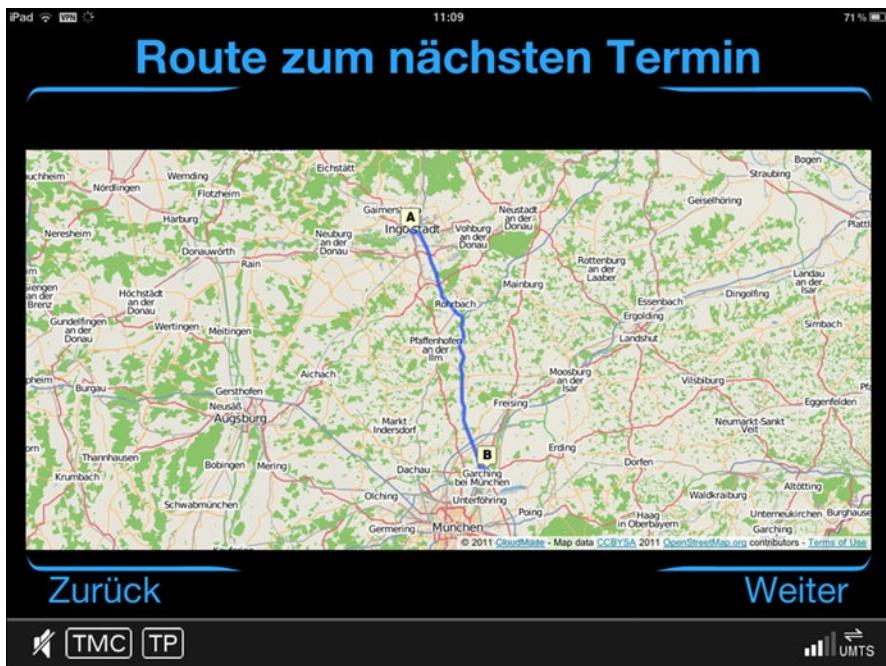
Das Automobil der Zukunft ist nicht mehr nur ein reines Fortbewegungsmittel zur Realisierung der Individualmobilität, sondern unterstützt den Nutzer auf vielfältige Art und Weise bei der Erfüllung seiner Aufgaben. Bezogen auf die berufliche Nutzung des Fahrzeugs wird es in diesem Zusammenhang in zunehmendem Maße zu einer Art „mobilem Büro“. Durch den immer weiteren Ausbau des mobilen Breitbandinternetzugangs sowie die immer leistungsfähigeren Ressourcen im Fahrzeug wird ein vielfältiges Spektrum an neuen innovativen Smart Automotive Apps zur Unterstützung mobiler Arbeit ermöglicht. Über die mobile Breitbandinfrastruktur können zudem Backend-Dienste mit in die Services im Fahrzeug integriert werden. Zusätzlich erlauben neue Bedienkonzepte wie berührungs empfindliche Touchscreens oder Spracheingabe und -ausgabe eine höchst intuitive Bedienung dieser Anwendungen im Fahrzeug.

Zusätzlich zu den notwendigen Kontroll- und Steuerungssystemen bieten Smart Automotive Apps dem Nutzer weitere Nutzenpotenziale, die weit über das Nutzenpotenzial des Fahrzeugs – die individuelle Mobilität – hinausgehen. Ein Beispiel für eine solche Anwendung, welche die Nutzenpotenziale von Smart Automotive Apps gut zur Geltung bringt, ist der Personal Car Assistant (PCA). Hierbei handelt

es sich um einen persönlichen Assistenzdienst im Fahrzeug, der den Nutzer über dessen E-Mails, Termine und Aufgaben auf dem aktuellen Stand hält. Hierzu nutzt der Service die Personalisierungsmöglichkeiten und greift über eine verschlüsselte Internetverbindung auf den Exchange-Account des Nutzers zu. Die Anwendung erlaubt es dem Nutzer zum einen, sich im stehenden Fahrzeug seine aktuellen E-Mails durchzulesen und sich einen Überblick über seine anstehenden Termine und Aufgaben zu verschaffen. Dieser Teil ist jedoch noch nicht wirklich „smart“ und kann auch nicht während der Fahrt – dem eigentlichen Nutzungsszenario des PCA – verwendet werden. Hierfür bietet der PCA zwei weitere Funktionen an: Die Zusammenfassungsfunktion „WrapUp“ sowie die Notifikationsfunktion „UpComing“. Diese beiden Funktionen bringen durch die Realisierung der weiter oben genannten Nutzenpotenziale von Smart Automotive Apps den persönlichen Sekretär ins Auto und bieten dadurch dem Nutzer einen echten Mehrwert.

Die Funktion WrapUp bringt den Fahrer durch eine kurze Zusammenfassung der aktuellen E-Mails, Termine und Aufgaben auf den neusten Stand. Diese Informationen werden ihm vorgelesen, so dass sie auch während der Fahrt genutzt werden kann. Der Anwender kann mit Sprachbefehlen, Wischgesten auf einem Touchscreen oder den Eingabemöglichkeiten des Audi MMI mit dem Service interagieren, ohne dabei von seiner primären Aufgabe – dem Fahren – abgelenkt zu werden. Lange E-Mails werden hierbei zusätzlich noch mittels einer Digest-Funktion auf etwa ein Zehntel ihrer ursprünglichen Länge zusammengefasst. Die inhaltliche Aussage des Textes bleibt jedoch auch weiterhin erhalten. Die Zusammenfassung wird hierbei nicht im Fahrzeug selbst vorgenommen, sondern durch einen Service, der über die mobile Internetverbindung auf dem Backendsystem aufgerufen wird. Durch diese Zusammenfassung erhält der Fahrer schneller einen Überblick über seine E-Mails und deren Inhalt. Außerdem kann er sich aufgrund der Kürze der Texte leichter an deren Inhalt erinnern und diesen falls notwendig für die weitere Aufgabenerledigung nutzen. Zusätzlich wird der Fahrer auch über Änderungen bei seinen Terminen unterrichtet. Wird ein Termin verschoben oder sagt eine Person die Teilnahme zu einem Termin ab, dann wird diese Änderung dem Anwender mitgeteilt, so dass dieser hierauf entsprechend reagieren kann.

Als weitere Funktion des PCA steht dem Fahrer die Upcoming-Funktion zur Verfügung. Auch diese ist dem Verhalten eines persönlichen Assistenten nachempfunden. Sie berechnet die Route zum nächsten Termin (s. Abb. 3.4) und behält nebenbei auch die aktuell anstehenden Termine des Nutzers im Auge. Stellt die Anwendung fest, dass der nächste Termin aufgrund der ermittelten Fahrzeit zum Zielort nicht rechtzeitig wahrgenommen werden kann, wird der Fahrer hierüber umgehend informiert. Für die Fahrzeit wird neben den allgemeinen Routeninformationen auch die aktuelle Verkehrslage auf der Fahrstrecke mit berücksichtigt. Bei dieser Anwendung werden sowohl die im Terminplan eingetragenen Termine auf dem Exchange-Server im Backend als auch aktuelle Verkehrsinformationen aus weiteren Backend-Systemen im Internet genutzt und zusammen mit dem aktuellen Aufenthaltsort des Fahrzeugs ausgewertet. Stellt das System einen zeitlichen Konflikt fest, wird der Fahrer hierüber informiert und bekommt die Möglichkeit, mit einfachen Sprachbefehlen darauf zu reagieren und z. B. eine E-Mail mit dem Hinweis auf seine Ver-



**Abb. 3.4** Beispielscreen der Funktion UpComing – Navigation zum nächsten Termin

spätung an die anderen Teilnehmer des Termins zu senden. Auch hierbei wird der Fahrer nicht von seiner primären Aufgabe abgelenkt, da der Service weitestgehend automatisiert und ansonsten durch Spracheingabe gesteuert die Aufgabenerfüllung übernimmt.

Der Fahrer bleibt durch die Nutzung des PCA auch unterwegs stets gut über seine E-Mails, Termine und Aufgaben informiert und kann sich durch das intelligente Assistenzsystem voll und ganz aufs Fahren konzentrieren.

### 3.5 Fazit

Der vorliegende Beitrag stellt einen Ansatz zur interaktiven Gestaltung von Smart Automotive Apps mit Hilfe der domänenspezifischen Modellierungssprache ASML vor. Unsere Absicht mit diesem Beitrag war es, auf die Herausforderungen in der Automotive-Domäne im Bezug auf die Entwicklung von Smart Automotive Apps hinzuweisen und einen möglichen Lösungsansatz hierfür vorzustellen.

Im Bezug auf die Entwicklung innovativer Smart Automotive Apps sind die Stakeholder im allgemeinen nur selten in der Lage ihre Visionen, Anforderungen und Einschränkungen von Anfang an vollständig auszudrücken. Meist ist zu Beginn der Entwicklung nur eine grobe Vision für eine neue innovative Smart Automotive App vorhanden, die sich erst im Laufe der weiteren Realisierung der Anwendung, ver-

bunden mit der Evaluation erster Prototypen, schärft. Außerdem verfügen die Experten der Fachdomäne meist nicht über das notwendige IT-Wissen, um die Potenziale und Grenzen von IT-Systemen und deren Auswirkung auf die Umsetzung einer Smart Automotive App beurteilen zu können. Im Gegenzug mangelt es den IT-Experten oft an Wissen bzgl. den Anforderungen und Bedürfnissen der Anwendungsdomäne.

Mit ASML liefern wir einen einfachen und verständlichen graphischen Modellierungsansatz, der die Zusammenarbeit von technischen und nicht-technischen Stakeholdern bei der Entwicklung und Umsetzung von Smart Automotive Apps ermöglicht. In Verbindung mit den beiden Konzepten der Modularisierung und Produktivierung ermöglicht dieser Ansatz eine sehr effiziente und effektive Entwicklung innovativer Smart Automotive Apps.

## Literaturverzeichnis

1. Bauer S (2011) Das vernetzte Fahrzeug – Herausforderungen für die IT. Informatik Spektrum 34(1):38–41. doi:10.1007/s00287-010-0504-9
2. Floyd C, Mehl W-M, Reisin F-M, Schmidt G, Wolf G (1989) Out of Scandinavia: Alternative Approaches to Software Design and System Development. Human-Com Interact 4:253–350
3. Antón AI (2003) Successful Software Projects Need Requirements Planning. IEEE Software, pp 44–46
4. Sawhney M (2006) Going Beyond the Product: Defining, Designing and Delivering Customer Solutions. In: Lusch RF, Vargo SL (eds) Toward a Service-Dominant Logic of Marketing: Dialog, Debate, and Directions. M.E. Sharpe, Armonk, NY, pp 365–380
5. Davies A, Brady T, Hobday M (2007) Organizing for solutions: Systems seller vs. systems integrator. Ind Marketing Management 36(2):183–193
6. Liggesmeyer P, Rombach D (2005) Software Engineering eingebetteter Systeme. Grundlagen – Methodik – Anwendungen. Elsevier Spektrum Akad. Verl., München
7. Baskerville RL (2008) What design science is not. Eur J Inf Sys 17(5):441–443
8. Bender K (2005) Embedded Systems - qualitätsorientierte Entwicklung: Qualitätssicherung bei Embedded Software. 1 edn Springer-Verlag, Berlin
9. Leimeister JM, Glauner C (2008) Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik. Wirtschaftsinformatik 50(3):248–251
10. Chesbrough HW (2003) Open Innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business School Press, Boston
11. Brooks F (1986) No Silver Bullet: Essence and Accidents of Software Engineering. Proceedings of the IFIP Tenth World Computing Conference
12. Briggs R, Gruenbacher P (2002) EasyWinWin: Managing complexity in requirements negotiation with GSS. In: 35th Hawaii International Conference on System Sciences. Big Island, Hawaii
13. Brügge B, Dutoit AH (2004) Objektorientierte Softwaretechnik. Mit Entwurfsmustern, UML und Java. Pearson Studium, München
14. Krauß T, Versteegen G, Mühlbauer S (2006) Model Driven Requirements Engineering. Informatik Spektrum 29(6):460–464
15. Breindahl C (2008) A Scandinavian Approach to Interaction Design. <http://www.dcdr.dk/uk/Menu/Update/Webzine/Articles/A+Scandinavian+Approach+to+Interaction+Design>. Accessed 21.11. 2008
16. Grechanik M, Conroy KM, Probst KA (2007) Finding Relevant Applications For Prototyping. In: International Conference on Software Engineering
17. Hoffmann H (2010) Prototyping Automotive Software and Services: Vorgehen und Werkzeug zur nutzerorientierten Entwicklung. Cuvillier Verlag, Göttingen

---

# Kapitel 4

## Smart Apps in einem vernetzten (auto)mobilen Umfeld: IT-Security und Privacy

B. Weyl, M. Graf, A. Bouard

**Zusammenfassung** Vernetzung von Fahrzeugen mit mobilen Kommunikationstechnologien und die Verknüpfung verteilter Daten, Funktionen und Dienste ermöglichen neue Formen von intelligenten Mobilitätsanwendungen im Fahrzeugkontext. Diese umfassen Smart (auto)mobile Apps für Fahrerinformation, Fahrerassistenz, Entertainment und mobile Communities. Die enge Verknüpfung unterschiedlicher Kontexte und Entitäten bedingt neue Bedrohungspotentiale. Dadurch leiten sich IT-Sicherheits- und Privacy-Anforderungen ab, die im Fahrzeugbordnetz und den entsprechenden Schnittstellen zu externen Geräten und Infrastrukturdiensten effizient und flexibel umgesetzt werden müssen. IT-Security- und Privacy-Lösungen sind Schlüsseltechnologien für sichere mobile Anwendungen bei gleichzeitigem Schutz persönlicher Informationen und Daten. Anhaltende öffentliche Diskussionen und Studien zum Thema Datenschutz und Sicherheit, gerade auch im Bereich sozialer Netzwerke und mobiler Anwendungen, zeigen, dass Datensicherheit und Schutz der Privatsphäre eine zentrale Rolle spielen. Es gilt folglich geeignete Vernetzungsarchitekturen mit den Randbedingungen der eingebetteten Systeme im Automobil und den neuen Herausforderungen hinsichtlich IT-Security und Privacy zu entwickeln.

---

Dr. Benjamin Weyl  
BMW Forschung und Technik GmbH, München,  
E-mail: benjamin.weyl@bmw.de

Maximilian Graf  
BMW Forschung und Technik GmbH, München,  
E-mail: maximilian.graf@bmw.de

Alexandre Bouard  
BMW Forschung und Technik GmbH, München,  
E-mail: alexandre.bouard@bmw.de

## 4.1 Einführung

Immer mehr Autohersteller bieten ihren Kunden innovative Funktionen (BMW ConnectedDrive<sup>1</sup>, Audi Connect<sup>2</sup>, GM OnStar<sup>3</sup>), basierend auf mobilen Kommunikationstechnologien an, durch die der Fahrer mit seinem Fahrzeug und seiner Lebenswelt vernetzt wird. Diese umfassen unter anderem Informationsdienste, aktuelle Verkehrsinformationen, E-Mails und Internet sowie telefonische Auskunftsdiensste. Zunehmend wird auch die Integration mobiler Endgeräte, wie Telefon und Musik-Player, unterstützt. Dabei werden unterschiedliche drahtlose und drahtgebundene Technologien wie Bluetooth, GSM oder USB eingesetzt.

In Zukunft werden weitere drahtlose Schnittstellen, wie z. B. W-LAN oder 802.11p, im Fahrzeug integriert. Damit wird die Vernetzung unterschiedlichster Daten- und Informationsquellen ansteigen. Darüber hinaus werden die Datenraten durch die Einführung neuer Mobilfunktechnologien wie beispielsweise LTE (Long Term Evolution) weiter erhöht. Für Fahrzeuge ergeben sich daraus eine Reihe von neuen mobilen Kommunikationsszenarien, die ein breites Anwendungsspektrum aus Sicherheit, Mobilität, Service und Unterhaltung abdecken und den Weg Richtung neuer intelligenter Mobilitäts-Applikationen ebnen.

Dabei gilt den Themen IT-Security, Datenschutz und Privacy besonderes Augenmerk. Anhaltende öffentliche Diskussionen und Studien zum Thema Datenschutz, gerade auch im Bereich sozialer Netzwerke [1] und mobiler Anwendungen [2], zeigen, dass Datensicherheit und Schutz der Privatsphäre eine zentrale Rolle spielen.

IT-Security und Privacy-Lösungen sind Schlüsseltechnologien zur Ermöglichung sicherer mobiler Anwendungen mit dem gleichzeitigen Schutz persönlicher Informationen und Daten. Es gilt folglich geeignete Vernetzungsarchitekturen mit den Randbedingungen der eingebetteten Systeme im Automobil und den neuen Herausforderungen hinsichtlich IT-Security und Privacy zu entwickeln [3, 4].

Zunächst wird ein Überblick über Smart Mobile Apps für zukünftige automobile Mobilitätsdienstleistungen gegeben. Hierbei werden verschiedene Anwendungsszenarien aufgezeigt und die neuen Herausforderungen dieser Anwendungen diskutiert. Insbesondere werden die IT-Security-Anforderungen und der Schutz der Privatsphäre sowie persönlicher Daten für Smart Apps im (auto)mobilen Umfeld analysiert und bewertet. Um diesen Anforderungen gerecht zu werden, wird ein Security- und Privacy-Konzept vorgeschlagen. Der Beitrag endet mit einer Zusammenfassung und einer abschließenden Diskussion.

---

<sup>1</sup> [www.bmw.de/connecteddrive](http://www.bmw.de/connecteddrive)

<sup>2</sup> [www.audi.de/de/brand/neuwagen/kommunikation/audi\\_online\\_dienste.html](http://www.audi.de/de/brand/neuwagen/kommunikation/audi_online_dienste.html)

<sup>3</sup> [www.onstar.com](http://www.onstar.com)

## 4.2 Anwendungen heute und morgen

Führende Automobilhersteller bieten mit ihren vernetzten Diensten bereits heute ein breites Angebot, das von Notruf und Pannenruf über Informations- und Auskunftsdiensste bis hin zur Ferndiagnose reicht. Künftige Car2X-Anwendungen werden darüber hinaus die aktive Sicherheit im Straßenverkehr deutlich verbessern [5]. Beispiele solcher zukünftiger vernetzter Fahrerassistenz-Anwendungen sind:

- Warnung vor Gefahren auf der vorausliegenden Strecke
- Signalisierung von Polizei- und Rettungsfahrzeugen zur Gassenbildung
- Kreuzungsmanagement mit präventiver Fahrzeugführung zur Kollisionsvermeidung
- Verbesserung des Verkehrsflusses
- Automatisches Erkennen von Verkehrszeichen durch das Fahrzeug
- Unterstützung des Fahrers durch Geschwindigkeitsempfehlungen
- Warnung vor Stopp-Signalen
- Assistenz für das Erkennen von Ampeln
- Unterstützung des Fahrers auf unübersichtlichen Strecken.

Um solche vernetzten Fahrerassistenzfunktionen zu ermöglichen, müssen zukünftig kommunikationsbasierte Informationen und durch klassische Sensorik gewonnene Umgebungsinformationen fusioniert werden.

Neben Car2X-basierten vernetzten Fahrerassistenzsystemen, bietet die zukünftige Vernetzung des Fahrzeugs weitere neue Vernetzungsszenarien. Neben der drahtlosen Kommunikation zwischen Fahrzeugen, mit Road-Side-Units (RSU) und der Infrastruktur zur Verbesserung der aktiven Sicherheit, werden hier weitere Kommunikationspartner und Mobilitätsdienstleister miteinander vernetzt. Solche Szenarien umfassen:

- Verbindung mit dem eigenen Zuhause zur Synchronisierung persönlicher Daten und Medieninhalte und der Verknüpfung zur Steuerung von Funktionen (z. B. Licht, Heizung, etc.)
- Vernetzung mit dem Arbeitgeber für ein komfortables Mobile Office im Fahrzeug
- Nutzung personalisierter Mehrwertdienste wie beispielsweise des personalisierten Radioempfangs
- Intelligente Anbindung mobiler Endgeräte im Fahrzeug zur Verwendung von mobilen Applikationen auf dem Endgerät im Fahrzeug
- Integration von Internetdiensten wie sozialen Netzwerken und mobilen Communities.

## 4.3 Wandel der Mobilität in Megacities

Neben Menge und Qualität von Anwendungen und Sensoren ändern sich auch die Rahmenbedingungen bei der Nutzung von Automobilen. Gerade für Megacities

werden neue Mobilitätsdienstleistungen erforscht um den sich verändernden Randbedingungen gerecht werden zu können. So werden neue Dienste wie beispielsweise Car Sharing<sup>4</sup> angeboten, um dem Kunden ein flexibles Mobilitätsmodell anzubieten. Ein Fahrzeug wandelt sich dadurch zu einem Nutzgegenstand, der von einer Vielzahl einander unbekannter Personen geteilt wird. Für den Zeitraum der Nutzung soll das Fahrzeug jedoch für den Kunden höchst personalisierbar und individualisierbar sein und es sollen die jeweiligen vernetzten Mobilitäts-Anwendungen zur Verfügung stehen.

Dadurch entstehen zusätzliche Herausforderungen gerade bezüglich der IT-Security und Privatsphäre der einzelnen Nutzer. Es reicht nicht mehr aus, persönliche Informationen innerhalb eines Fahrzeugs zu schützen. Vielmehr müssen neue Probleme, wie eine sichere Übertragung persönlicher Informationen ins Fahrzeug und insbesondere eine sichere Entfernung der Daten aus dem Fahrzeug, behandelt werden. Nur so lässt sich sicherstellen, dass ein Nutzer keinerlei sensible Informationen über den vorherigen Nutzer erlangen kann. Zusätzlich müssen sichere Transaktionen zwischen verschiedenen Mobilitätsdienstleistern ermöglicht werden, um dem Kunden eine möglichst effiziente und schnelle Nutzung dieser neuen Mobilitätsanwendungen zu ermöglichen.

## 4.4 Bedrohungen und IT-Sicherheitsanforderungen

Durch die ständig steigende Anzahl von Funktionen, die auf der verteilten Vernetzung von Informationsquellen und -senken basieren, steigt auch das Gefahrenpotenzial durch Angriffe auf solche Funktionen sowie auf personenbezogene Informationen des Nutzers. Um die Sicherheit der Funktionen und die Privatsphäre des Nutzers trotz der geänderten Rahmenbedingungen zu gewährleisten, ist es deshalb unverzichtbar, die IT-Sicherheit des gesamten Lebenszyklus der Daten zu garantieren.

In den folgenden Unterkapiteln behandeln wir dafür die möglichen Bedrohungen in zukünftigen Vernetzungsszenarien hinsichtlich Privacy und Security. Ausführliche Risiko- und Bedrohungsanalysen finden sich in [6].

### 4.4.1 Bedrohungen hinsichtlich Privacy

#### 4.4.1.1 Sensibilität der Daten und deren Senken

Die Kritikalität von Informationen ist für Nutzer schwer nachvollziehbar, da sie von unterschiedlichen Faktoren abhängt. Neben Erfassungshäufigkeit der Daten und Art der Datensenke muss auch die Kombinationsmöglichkeit von Informationen aus-

---

<sup>4</sup> [www.car2go.com](http://www.car2go.com), [www.drive-now.com](http://www.drive-now.com)

unterschiedlichen Datenquellen betrachtet werden und damit auch die Information, die aus dieser Kombinationsmöglichkeit abgeleitet werden kann.

Diese Problematik lässt sich beispielsweise anhand von Location Based Services darstellen. Verwendet ein Nutzer einen Dienst um ein Restaurant in seiner Nähe zu finden, überträgt er einmalig seine aktuelle Position. Seine Privatsphäre wird dadurch nur bedingt eingeschränkt. Wird dieser Dienst öfter genutzt und werden zusätzlich auch noch Nutzerinformationen, wie z. B. Nutzernname und Adresse, übertragen, so kann der Dienstleister persönliche Interessen des Nutzers und Konsumverhalten ableiten und diesem daraufhin stark personalisierte Werbung auf seinem mobilen Endgerät anbieten. Weiterhin kann der Dienstleister über den Wohnort und die Verbindung mit dem Profil anderer Nutzer ableiten, welches Konsumverhalten in einem Viertel besonders auffällig ist und dort entsprechend Werbung platzieren.

Dieses Beispiel zeigt, wie sich durch Datenaggregation unterschiedlicher Quellen neue Informationen ableiten lassen. Im Folgenden sollen weitere Angriffsszenarien erläutert werden, bei denen gerade die Relevanz von Vertraulichkeit fahrzeugrelevanter Daten betrachtet werden.

#### 4.4.1.2 Erstellung von Bewegungsprofilen

Die Verwendung der Fahrzeugpositionsdaten könnte die Erstellung detaillierter Bewegungsprofile ermöglichen, aus denen Rückschlüsse auf Wohnort oder Arbeitsplatz, aber auch auf Gewohnheiten eines Fahrzeugbesitzers gezogen werden könnten. So können Mobilfunkanbieter über die Bestimmung des Aufenthaltsortes des mobilen Endgerätes des Kunden bereits detaillierte Bewegungsprofile erstellen.

Entsprechend wäre es denkbar, dass aus safety-relevanten Informationen, wie z. B. Positions- und Geschwindigkeitsinformationen, Rückschlüsse auf das Fahrverhalten bestimmter Nutzer möglich wäre. Deshalb werden z. B. für Car2X-Kommunikation Techniken zur Verschleierung der realen digitalen Identität, basierend auf wechselnden Pseudonymen, vorgesehen. Entsprechende Techniken werden innerhalb der Security und Privacy Working Group [7] zur Standardisierung vorbereitet und aktuell in der ETSI ITS Working Group 5 [8] standardisiert.

#### 4.4.1.3 Auslesen persönlicher Informationen

Smart (auto)mobile Apps bringen immer mehr persönliche Informationen ins Fahrzeug. Durch die Verwendung von eMail, Telefon, Kalender, sozialen Netzwerken und weiteren Diensten sind detaillierte Rückschlüsse auf Tagesabläufe, Kaufverhalten, soziales Umfeld etc. möglich. Neben dem immateriellen Schaden, den ein Angreifer durch Erlangen dieser Informationen anrichten kann, sind auch materielle Verluste, beispielsweise durch betrügerische Verwendung von Bank- oder Kreditkarten-Informationen, möglich.

## 4.4.2 Bedrohung hinsichtlich IT-Security

Neben den Bedrohungen basierend auf persönlichen Daten und Informationen entstehen durch die Vernetzung von Funktionen neue Sicherheitsherausforderungen. Dazu zählen einerseits Angriffe auf die fahrzeugexterne Kommunikation, d. h. die Kommunikation mit z. B. anderen Fahrzeugen oder Infrastruktur-Entitäten, andererseits die fahrzeuginterne Kommunikation mit potentiellen Bedrohungen der Fahrzeug-Steuergeräte (ECUs) oder Sensoren/Aktuatoren.

### 4.4.2.1 Bedrohungen der fahrzeugexternen Kommunikation

Ein wesentlicher Angriffspunkt bei der fahrzeugexternen Kommunikation ist die Manipulation von Daten oder das Einschleusen von manipulierten Daten. Dies kann zu Fehlverhalten von Car2X-basierten Funktionen führen.

Daneben gibt es Denial-of-Service-Angriffe, bei denen das Kommunikationsnetz, z. B. durch Überfluten (flooding) mit Nachrichten oder durch die physische Störung der Kommunikationstechnologie (jamming), beeinträchtigt wird.

### 4.4.2.2 Bedrohungen der fahrzeuginternen Kommunikation

Neben der fahrzeugexternen Kommunikation gilt es auch eine Reihe von Bedrohungen der fahrzeuginternen Kommunikation zu berücksichtigen. Da das Fahrzeug-Bordnetz aus einer Vielzahl vernetzter Sensoren/Aktuatoren und Steuergeräte besteht, sind vereinzelte Funktionen, auf denen eine Car2X-Funktion basiert, relevante Angriffspunkte. So wäre es denkbar, dass eine Decentralized Environment Notification Message (DENM) zur Warnung vor einer Gefahrenstelle fälschlicherweise an andere Fahrzeug ausgesendet wird. Deshalb ist für dedizierte Anwendungen die Kommunikation zwischen Sensor und Steuergerät abzusichern, da hier Nachrichten manipuliert oder eingeschleust werden könnten und gegebenenfalls eine Car2X-Nachricht fälschlicherweise ausgesendet werden könnte.

Zusätzlich zur Kommunikation zwischen Fahrzeugkomponenten (ECUs, Sensoren, Aktuatoren) sind, je nach Anforderung aus dem Anwendungsszenario, diese Komponenten gegen Manipulationen an der Software zu schützen. Hier gilt es zu verhindern, dass Schadsoftware ins Fahrzeug übertragen wird, die etwaige Daten manipulieren kann.

## 4.5 Herausforderungen und IT-Sicherheits- und Privacyanforderungen

Durch diese Vielzahl unterschiedlicher mobiler Anwendungen entstehen neue technische, wirtschaftliche und rechtliche Herausforderungen:

- Sichere offene Plattformen und standardisierte Schnittstellen zur effizienten Nutzung und Wartbarkeit von Software
- Standardisierte Kommunikationstechnologien und -protokolle
- IT-Sicherheit zum vertrauenswürdigen Austausch von Informationen unter Gewährleistung der Privatheit sowie der Berücksichtigung vertraulicher Daten
- Neue Geschäftsmodelle, Prozesse und rechtliche Aspekte, wobei die Rollen und Rechte der Teilnehmer in der Wertschöpfungskette zu definieren sind, damit dem Markt funktionsfähige Geschäftsmodelle angeboten werden können
- Absicherung des gesamten Daten-Lebenszyklus: Erfassung, Haltung, Übertragung, Löschung
- Transparente und nachvollziehbare Darstellung der Privacyproblematik für den gemeinen Nutzer.

Insbesondere durch die neuen Bedrohungen müssen dedizierte IT-Sicherheits- und Privacy-Anforderungen berücksichtigt werden. Diese werden in den nächsten beiden Unterkapiteln zusammengefasst.

### ***4.5.1 Privacy und schützenswerte Daten***

Aus den bisher beschriebenen Angriffsszenarien leiten sich u.a. folgende schützenswerte Daten ab:

- Persönliche Daten des Nutzers, z. B. Telefonbuch- und Kontaktdaten
- Persönliche Einstellungen und Profile, z. B. Einstellungen für Personal Radio mit individuellem Musikgeschmack
- Fahr- bzw. Bewegungsprofile durch unautorisierte Aufzeichnung von Fahrzeugpositionsdaten im Rahmen von Location Based Services
- Detaillierte Informationen über das Fahrverhalten
- Nutzungsverhalten bei der Verwendung von Internet- und Mehrwertdiensten im Fahrzeug, beispielsweise besuchte Homepages oder Lesezeichen
- Identitäten beim Austausch von fahrzeugbezogenen Informationen, z. B. bei der lokalen Gefahrenwarnung, so dass die Privatheit des Nutzers verletzt wird.

Um die Privatsphäre und den Schutz von persönlichen Informationen und Daten zu gewährleisten, können verschiedene Sicherheits- und Privacymechanismen verwendet werden. Beispiele sind Verschlüsselung von Daten, Zugriffsschutz für private Daten, Pseudonymisierung von Identitäten oder die Anonymisierung bzw. Verschleierung von Informationen.

### ***4.5.2 IT-Security-Anforderungen***

Zum Schutz der externen und internen Kommunikation sind folgende IT-Sicherheitsanforderungen zu berücksichtigen:

- Integrität und Authentizität von Daten auf der externen Kommunikationsstrecke: Detektion von manipulierten oder eingeschleusten Daten.
- Integrität und Authentizität von Daten auf der internen Kommunikationsstrecke: Detektion von manipulierten oder eingeschleusten Daten auf fahrzeuginternen Bussystemen für ausgewählte Funktionen.
- Integrität und Authentizität von Software auf Fahrzeugkomponenten: Verhinderung oder Detektion einer Manipulation oder des Einschleusens von Software z. B. auf einem Steuergerät. Gegebenenfalls muss die Plattform fähig sein, einer externen Entität eine Attestation zur Vertrauenswürdigkeit der Plattform auszustellen.
- Zugriffskontrolle zu Funktionen und Daten: Zum Schutz ausgewählter Funktionen und Daten muss der Anforderer authentisiert werden und seine Zugriffsrechte müssen kontrolliert werden.

## 4.6 IT-Security- und Privacy-Architektur für Smart (auto)mobile Apps

Gerade bei kommunikationsbasierten Anwendungen spielt die IT-Sicherheit der Funktionen und der Schutz der Privatheit eine bedeutende Rolle. Deshalb wird in diesem Kapitel ein Überblick über eine IT-Security- und Privacy-Architektur für kommunikationsbasierte Smart (auto)mobile Apps vorgestellt.

Um eine flexible und skalierbare Security- und Privacy-Architektur zu erreichen, werden folgende Basiskonzepte vorgeschlagen [9, 10]:

- Modularisierung von IT-Sicherheitskomponenten: Durch einen modularisierten Aufbau der Architektur kann das System für spezifische Sicherheitsanforderungen angepasst und konfiguriert und somit für das Zielsystem optimal ausgelegt werden.
- Austauschbarkeit: Durch geeignete Modularisierung der Sicherheitsarchitektur und das Anbieten geeigneter Schnittstellen und Plattformen verbessern sich Austauschbarkeit sowie Erweiterbarkeit.
- Abstraktion von Schnittstellen der Sicherheitsdienste: Um die Integration von Sicherheitsdiensten, z. B. auf Anwendungsebene, zu vereinfachen, ist eine geeignete Abstraktion der Schnittstellen zu definieren.
- Konfigurierbarkeit: Statische und dynamische Konfigurierbarkeit von Sicherheitsmodulen, um nach der Vorkonfiguration die Konfiguration auch anwendungsspezifisch im Feld und während der Laufzeit dynamisch aktualisieren zu können.

In Abb. 4.1 werden die Basismodule einer IT-Sicherheits- und Privacy-Architektur und deren mögliche Umsetzung für Smart (auto)mobile Apps vorgestellt. Auf Fahrzeugseite werden Security- und Privacy-Mechanismen auf verschiedenen Schichten des ISO/OSI-Schichtenmodells über Protokoll-spezifische Layer-Enforcement-Points angebunden. Dabei nutzen die Layer-Enforcement-Points einheitliche

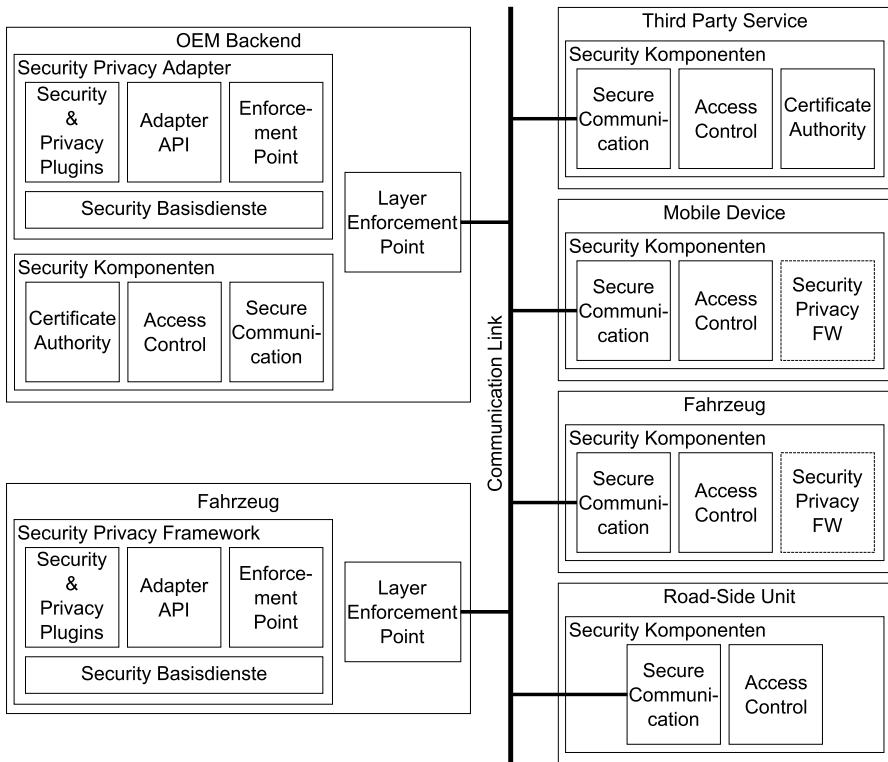
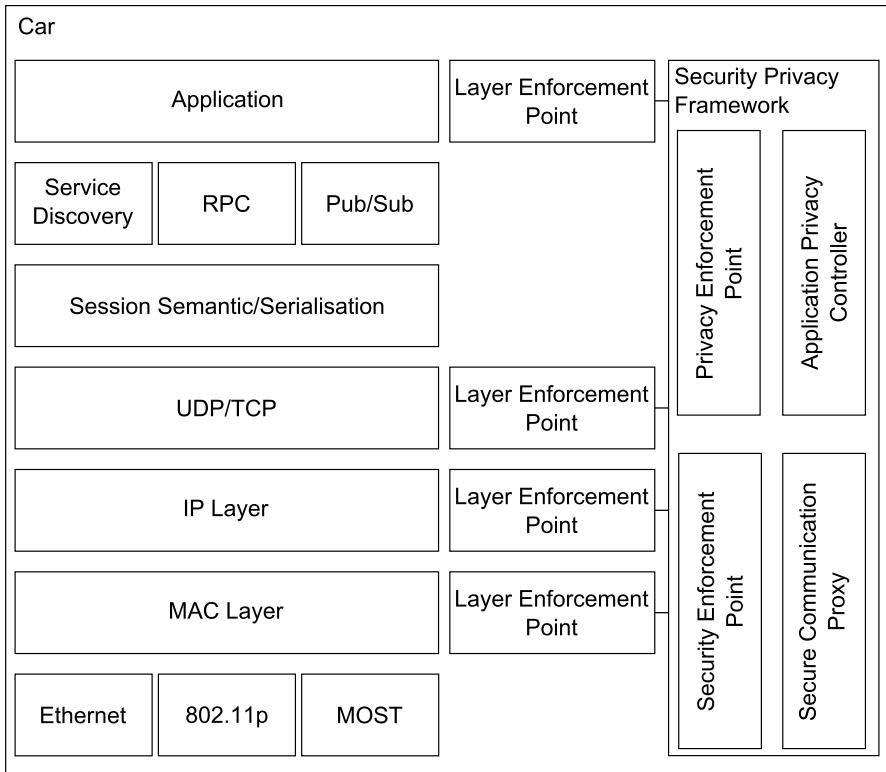


Abb. 4.1 Überblick Gesamtarchitektur

Schnittstellen der Enforcement-Points des Frameworks. Die implementierungs-spezifischen Layer-Enforcement-Points teilen dem Enforcement-Point des Frameworks die erforderlichen Security- und Privacyanforderungen mit. Diese Anforderungen werden dann vom Enforcement Point innerhalb des Frameworks umgesetzt. Innerhalb dieser Enforcement-Points werden, je nach Security- und Privacy-Anforderungen, verschiedene interne Security- und Privacy-Mechanismen (Plugins) aufgerufen. Die Plugins werden durch eine Abstraktionsschicht, die Adapter API, innerhalb des Frameworks eingebunden. Kann eine externe Entität die Anforderungen nicht erfüllen, kann je nach Policy nur eine eingeschränkte Kommunikation stattfinden.

Modularisierung und Skalierbarkeit werden durch die Adapter-API erreicht: Neue Mechanismen können als Plugins in das Framework eingebunden werden. Die Umsetzung von Anforderungen wird durch den Enforcement-Point des Frameworks abstrahiert.

Bei Anbindung mobiler Endgeräte können OEM-Applikationen entsprechende Security- und Privacy-Komponenten verwenden, ebenso wie auf OEM-Backend-Seite. Dabei können selbstverständlich auch Standard-Security-Systeme verwendet werden. Je nach Security-Mechanismus auf Third-Party-Seite, in anderen Fahrzeu-



**Abb. 4.2** Architektur des Security-Privacy-Frameworks

gen oder bei Road-Side-Infrastruktur-Komponenten sind korrespondierende Mechanismen über die Adapter-API im Fahrzeug integrierbar.

Abbildung 4.2 zeigt eine detaillierte Sicht der oben beschriebenen Architektur auf Fahrzeugseite und die mögliche Verwendung des Security-Privacy-Frameworks in einer entsprechenden IP-basierten Middleware-Architektur. Über die Layer-Enforcement-Points werden auf jeder Schicht die Security- und Privacy-Mechanismen über die Dienste des Frameworks eingebunden. Security-relevante Dienste werden über die Adapter-API des Secure Communication Proxy zur Verfügung gestellt, Privacy-relevante Dienste über die API des Application Privacy Controller. Der Enforcement-Point des Frameworks nutzt entsprechend den Anforderungen diese angebotenen Dienste. Die API bindet die benötigten Plugins ein.

Weitere Details zur fahrzeugseitigen Security- und Privacy-Architektur werden in den folgenden Abschnitten erläutert.

## 4.6.1 Secure Communication Proxy

Der Secure Communication Proxy (vgl. Abb. 4.3) hat die Aufgabe, sichere, d. h. integritätsgeschützte, authentische und vertrauliche Kommunikationskanäle zur Verfügung zu stellen sowie die Kommunikation entsprechend definierten Regeln und Heuristiken zu filtern. Dazu gehört auch die Zugriffskontrolle, d. h. Authentisierung und Autorisierung, zu Ressourcen, Daten und Funktionen. Der Secure Communication Proxy filtert die Kommunikation über den Secure-Filter-Router und baut über den Connection-Manager einen sicheren Kanal auf. Der Connection-Manager kann dabei unterschiedliche Security-Protokolle unterstützen und so bedarfsgerecht mit der externen Seite kommunizieren. Der Security Context Manager stuft den Security-Level der Verbindung zur externen Entität, z. B. einem mobilen Endgerät oder einem Backend-Dienst, ein (vlg. Abschn. 4.6.1.3). Abhängig von diesem Security-Level werden mehr oder weniger kritische Informationen an fahrzeuginterne Komponenten übertragen oder fahrzeuginterne Funktionen aufgerufen.

### 4.6.1.1 Secure Filter-Router

Als Basis der Security-Architektur dient der Secure Filter-Router, der die eingehenden Daten verarbeitet. Einerseits filtert er den Datenverkehr nach unterschiedlichen Kriterien (z. B. IP-Adresse, Port, Protocol) und bietet Sicherheitsmaßnahmen wie Verschlüsselung auf Schicht 2. Anderseits ermöglicht er sowohl sichere Ende-zu-Ende-Kommunikationsverbindungen als auch die Möglichkeit als Vermittler zwischen externen und internen Komponenten zu dienen. Die Vermittler-Funktion ermöglicht eine Entkopplung zwischen fahrzeuginternen und -externen IT-Security-

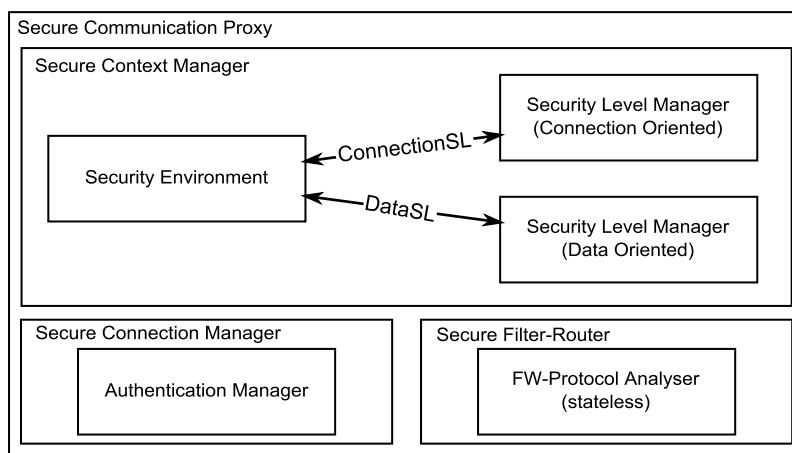


Abb. 4.3 Architektur des Secure Communication Proxy

Mechanismen um der Komplexität im Fahrzeubordnetz trotz zunehmender Vernetzung entgegenzuwirken.

#### **4.6.1.2 Connection Manager**

Der Connection Manager bietet Sicherheitslösungen um authentische und vertrauliche Kommunikationsbeziehungen aufzubauen. Dazu bietet er die Möglichkeit der Aushandlung eines Sicherheitsprotokolls basierend auf den zur Verfügung stehenden Sicherheitsmechanismen, z. B. SSL/TLS und IPsec. Aufgebaute sichere Verbindungen werden den jeweiligen Applikationen und Diensten der Middleware zugeordnet und als sichere Session verwaltet.

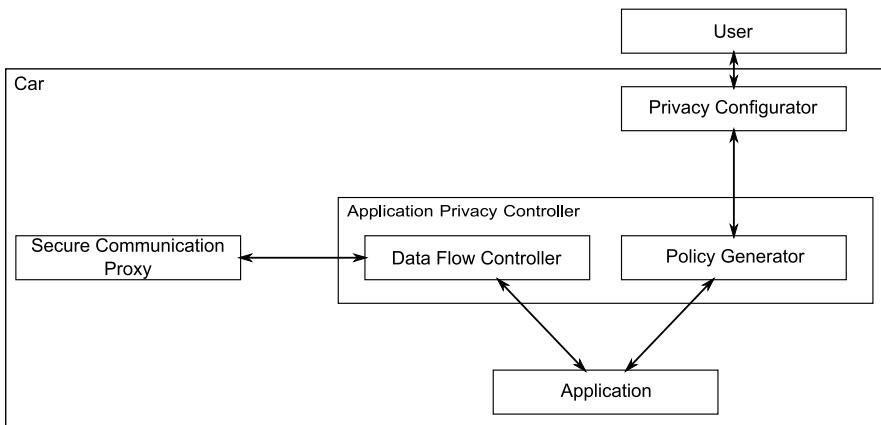
#### **4.6.1.3 Secure Context Manager**

Der Secure Context Manager analysiert Daten im jeweiligen Kontext auf Anwendungsebene für eingehende und ausgehende Nachrichten und definiert zwei Sicherheitsattribute:

- Verbindungs-orientierter Security-Level: Dieser Sicherheits-Tag charakterisiert das Vertrauen in Bezug auf Sicherheitsanforderungen (Authentifizierung, Vertraulichkeit, Integrität, etc.), das der aktuellen Verbindung zugeordnet wird. Je nach Security-Level der Verbindung werden bestimmte interne Dienste nach außen freigegeben oder externe fahrzeugintern zur Verfügung gestellt.
- Daten-orientierter Security-Level: Je nach Schutz und ermittelter Sicherheit eines Datenobjekts, z. B. Signaturprüfung und erkannte Anomalien, wird diesem Objekt ein Security-Level zugeordnet. Vor der Weiterleitung wird der Tag dem Paket angefügt. Der Empfänger kann dann selbstständig entscheiden, ob die Nachricht weiter verarbeitet wird. Je nach Policy werden erkannte Risiken bereits am Proxy ausgefiltert.

### **4.6.2 Application Privacy Controller**

Neben der Absicherung der Rohdaten und Nachrichten soll auch die Privatsphäre geschützt werden. Dies übernimmt der Application Privacy Controller. Dieser ist in zwei Komponenten aufgeteilt: den Data Flow Controller und den Policy Generator. Die Aufgabe des Application Privacy Controllers ist es zu entscheiden, welche Informationen einer Anwendung zur Verfügung gestellt werden dürfen. Da ein Nutzer selbst entscheiden können muss, wem er seine Informationen zu welchem Zeitpunkt zur Verfügung stellen möchte, definiert dieser zu jeder Anwendung Privacy-Policies, in denen Applikationen der Zugriff auf Informationen erlaubt oder verboten werden. Dabei wird der Nutzer durch den Privacy Configurator (vgl. Abb. 4.4)



**Abb. 4.4** Architektur des Application Privacy Controllers

mit Informationen über die Applikation unterstützt. Drei wesentliche Faktoren spielen dabei eine Rolle. Die Vertrauenswürdigkeit des externen Gerätes, die Kritikalität der Daten und die Vertrauenswürdigkeit des Dienstanbieters.

#### 4.6.2.1 Vertrauenswürdigkeit des externen Gerätes

Die Vertrauenswürdigkeit des externen Gerätes wird vom Secure Communication Proxy beurteilt und festgelegt. Nur durch eine hohe Sicherheit der Verbindung kann sichergestellt werden, dass die übertragenen Daten nicht von einem Angreifer mitgehört oder manipuliert werden können. Aufgrund dessen spielt die Verbindungssicherheit eine Rolle beim Privatsphäreschutz eines Nutzers.

#### 4.6.2.2 Kritikalität der Daten

Während die Bewertung der Verbindungssicherheit durch den Security-Proxy übernommen wird, stellen Kritikalität der Daten und Vertrauenswürdigkeit der Dienstanbieter große Herausforderungen für den Privacy-Proxy dar. Dies wird durch unterschiedliche Studien [2] belegt. Internetnutzer sind zwar der Meinung, über ein ausgeprägtes Privatsphärebewusstsein zu verfügen, verhalten sich aber nicht entsprechend. Grossklags et al. [11] stellen fest, dass Internetnutzer nicht einmal bereit sind, Cent-Beträge in den Schutz ihrer Privatsphäre zu investieren. Demzufolge muss eine Privacyarchitektur dem Nutzer diese Gefahren transparent und vor allem nachvollziehbar darstellen. Wie bereits beschrieben, ist eine solche Nachvollziehbarkeit jedoch eine echte Herausforderung für System und Nutzer. Schon die Menge der Einzeldaten ist für den Nutzer schwer zu überblicken. Beachtet man darüber hinaus die Informationen, die durch Aggregation und Kombination von Daten er-

mittelt werden können, ist es einem Nutzer nicht möglich, Daten und potentielle Gefahren ausreichend zu bewerten. Folglich muss eine Möglichkeit entwickelt werden, eine für den Nutzer transparente und nachvollziehbare Darstellung der Daten zu erreichen.

#### **4.6.2.3 Vertrauenswürdigkeit des Dienstanbieters**

Um die Vertrauenswürdigkeit eines Dienstanbieters zu bewerten, schlagen die Autoren ein Community Based Trust Establishment vor. Nutzer können im Rahmen einer Internetplattform ihre Erfahrungen teilen. Die so erfassten Informationen sollen den Nutzer wiederum bei der Erstellung von Privacy-Policies unterstützen.

#### **4.6.2.4 Ablauf**

Der Ablauf der Privacy-Gewährleistung wird in zwei Phasen unterteilt:

##### **Registrierung**

Wird eine neue Applikation installiert, meldet sich diese beim Policy Generator an. Dieser übermittelt die für den Nutzer relevanten Daten über Applikation, Dienstanbieter und Verbindungssicherheit an den Privacy Configurator, welcher in Abstimmung mit dem Nutzer die Privacyeinstellungen für die Anwendung bestimmt. Der Policy Generator erstellt nun entsprechend der Nutzereinstellungen eine Policy zur jeweiligen Anwendung und hinterlegt diese beim Data Flow Controller. Die Policy enthält Regeln, welche Informationen die Anwendung in welcher Häufigkeit abfragen darf.

##### **Enforcement**

Möchte eine Anwendung ein bestimmtes Datum vom Fahrzeug, beispielsweise die aktuelle Geschwindigkeit, abfragen, entscheidet der Data Flow Controller anhand der hinterlegten Policies, ob die Information zur Verfügung gestellt werden kann. Auf diese Weise soll dem Nutzer ein maximales Maß an Privatsphäre ermöglicht werden.

### **4.7 Zusammenfassung und Diskussion**

Mit der zunehmenden Vernetzung des Fahrzeugs und neuer intelligenter mobiler Anwendungen entstehen neue IT-Sicherheits- und Privacyanforderungen. Diese um-

fassen die Integrität, Authentizität und Vertraulichkeit dedizierter Daten auf fahrzeuginternen und -externen Kommunikationswegen. Zusätzlich gilt es Software auf Plattformen, Kryptoalgorithmen und Schlüsselmaterial zu schützen. Die vorgestellte IT-Security- und Privacyarchitektur ermöglicht eine bedarfsgerechte Umsetzung dieser IT-Sicherheits- und Privacyanforderungen. Je nach Anwendungsfall können entsprechende Security- und Privacy-Mechanismen zwischen Kommunikationsentitäten, Anwendungen und Diensten umgesetzt werden. Hierfür stellt die Architektur geeignete Schnittstellen zur Verfügung, um auch neue Sicherheitsmechanismen einbinden zu können. Um der Vielfalt der Security-Mechanismen auf der fahrzeugexternen Seite entsprechen zu können, findet auf Fahrzeugseite eine Entkopplung der Mechanismen zur fahrzeuginternen Seite statt. Der Security-Proxy unterstützt nach außen hin die verschiedenen Mechanismen, während diese auf fahrzeuginternen Seite vereinheitlicht sind und eine entsprechenden Umsetzung im Proxy stattfindet. Darüber hinaus verschränkt die beschriebene Architektur IT-Security- und Privacy-Mechanismen. Basierend auf den etablierten Vertrauensbeziehungen zwischen Informationsquelle und -senke und dem definierten Privacy-Schutzbedürfnis kontrolliert die Architektur den Informationsfluss. Die Sicherheit und Privatheit des Informationsflusses wird lokal auf Nutzerseite überprüft und den Bedürfnissen dynamisch angepasst. So unterstützt die IT-Sicherheits- und Privacy-Architektur für das Fahrzeug:

- die flexible Einbindung neuer Security-Mechanismen
- anwendungsspezifische sichere Kommunikationsbeziehungen mit der Entkopplung fahrzeuginterner und -externer IT-Security-Mechanismen
- den nutzertransparenten und -kontrollierbaren Umgang mit persönlichen Daten.

## Literaturverzeichnis

1. A. Poller (2009). Privatsphärenschutz in Soziale-Netzwerke-Plattformen. Studie. Fraunhofer Institut für Sichere Informationstechnologie. 2009.
2. Nokia Siemens Networks (2011). Taming the data privacy dragon - How CSPs can become the prime protectors of personal privacy.
3. EU FP7 Projekt EVITA (2011). E-safety Vehicle Intrusion Protected Applications. [www.evita-project.org](http://www.evita-project.org) (26. Oktober 2011)
4. Sicherheit in eingebetteten IP-basierten Systemen (SEIS) (2011). BMBF-gefördertes Projekt.
5. Timo Kosch (2004). Local Danger Warning based on Vehicle Ad-hoc Networks: Prototype and Simulation. In Proceedings of 1st International Workshop on Intelligent Transportation (WIT).
6. A. Ruddle, D. Ward, B. Weyl, S. Idrees, Y. Roudier, M. Friedewald, T. Leimbach, A. Fuchs, S. Gürgens, O. Henniger, R. Rieke, M. Ritscher, H. Broberg, L. Apvrille, R. Pacalet, und G. Pedroza. Security requirements for automotive on-board networks based on dark-side scenarios. Technical Report Deliverable D2.3, EVITA Project, 2009.
7. Car2Car Communication Consortium, [www.car-to-car.de](http://www.car-to-car.de) (26. Oktober 2011)
8. ETSI Working Group 5 Security und Privacy, [www.etsi.org](http://www.etsi.org) (26. Oktober 2011)
9. Benjamin Weyl et al. (2010). EVITA D3.2: Secure On-board Architecture Specification. Deliverable des EVITA-Projekts (Referenznummer 224275, ICT-2007.6.2).

10. Benjamin Weyl, Benjamin Glas et al. (2011). SEIS AP4.3: Sicherheit auf Ebene der Applikation und ihrer Middleware. Ergebnisdokument des SEIS-Projektes.
11. Grossklags, Jens und Acquisti, Alessandro (2007). When 25 Cents is too much: An Experiment on Willingness-To-Sell and Willingness-To-Protect Personal Information. In Proceedings of Workshop on the Economics of Information Security.

**Teil III**

**Anwendersicht: Business-Apps für  
Mitarbeiter**

---

# Kapitel 5

## Mobile Apps für industrielle Anwendungen am Beispiel von Siemens

Andreas Zeidler, Roland Eckl, Wolfgang Trumler und Marquart Franz

**Zusammenfassung** „Smarte mobile Applikationen“, meist nur kurz Apps genannt, etablieren sich mehr und mehr zu einem eigenständigen Zweig der Software-Entwicklung und tragen bereits heute substantiell zum Value Add einiger weniger Firmen wie beispielsweise Apple oder Google bei. Bislang sind die meisten Anwendungen jedoch dahingehend optimiert, einem einzelnen Anwender einen individuellen Mehrwert zu bieten, etwa anhand eines Reiseführers, durch den Zugriff auf ein soziales Netzwerk oder in der Form von Spielen für den kurzfristigen Zeitvertreib. Für einen weltweit tätigen Industriekonzern wie die Siemens AG ist es im Gegensatz dazu jedoch von weitaus größerem Interesse, mobile Anwendungen dabei als Teil des internen und externen Produktpportfolios zu betrachten. Hier sind beispielsweise Applikationen zu nennen, welche mit den Siemens-eigenen Produkten eng zusammenarbeiten und diese sinnvoll ergänzen können. In diesem Beitrag werden die speziellen Anforderungen an diese Klasse der industriellen Applikationen anhand der Automatisierungspyramide näher beleuchtet. Es werden anschließend einige ausgewählte Applikationen aus den Siemens-Sektoren vorgestellt, wobei die jeweils zugrunde liegende Problemstellung präsentiert und diskutiert wird. Ebenso werden erste Erfahrungen mit der jeweiligen App und der dadurch entstandene wirtschaftliche Nutzen erläutert. Abschließend wird ein kurzer Ausblick auf

---

Andreas Zeidler  
Siemens AG, Corporate Research and Technologies, München,  
E-mail: a.zeidler@siemens.com

Roland Eckl  
Siemens AG, Corporate Research and Technologies, München,  
E-mail: eckl.roland@siemens.com

Wolfgang Trumler  
Siemens AG, Corporate Research and Technologies, Erlangen,  
E-mail: wolfgang.trumler@siemens.com

Marquart Franz  
Siemens AG, Corporate Research and Technologies, München,  
E-mail: marquart.franz@siemens.com

die nächsten Schritte und die sich gerade in der Definition befindlichen Prozesse für die App-Entwicklung innerhalb der Siemens AG gegeben.

## 5.1 Einleitung

Der technische Fortschritt der letzten fünf Jahre im Bereich der Benutzerinteraktion sowie die Verfügbarkeit von Smartphones auf breiter Basis bildeten die Grundlage für die beinahe explosionsartig anwachsende Anzahl sogenannter Apps – Applikationen, welche der Anwender online auf sein Smartphone laden und direkt ausführen kann. Aktuelle Studien [1, 2] zeigen, dass das Interesse an der Entwicklung von Mobile Apps rasant steigt und auch „klassische“ Software Engineering Communities, wie Eclipse, ein immer größer werdendes Interesse an der Entwicklung zeigen.

Es gibt für nahezu jeden denkbaren Fall eine App. Die größte Anzahl an Apps findet sich jedoch in den Bereichen des Social Networking, der Spiele, der Kommunikation (Mail, SMS, Chat) sowie Anwendungen, die Informationen auf Landkarten darstellen, beispielsweise zur Navigation.

Anwendungen für Mobiltelefone gibt es bereits seit vielen Jahren. Anfänglich beschränkte sich der Funktionsumfang dieser Apps auf die im CLDC-Profil [3] definierten Möglichkeiten, welche durch die Java Community spezifiziert wurden. Eine entscheidende Rolle für den Durchbruch und die weite Verbreitung von Apps war jedoch die Einführung des iPhone und die Bereitstellung der zugehörigen API für Entwickler. Aber auch die Einführung eines großen Touch-Screen, welcher inzwischen bei praktisch allen Smartphones zu finden ist, gepaart mit einer einfachen Bedienbarkeit, öffnete den Raum für eine große Anzahl neuer Anwendungen. Angefangen bei Spielen, welche die gesamte zur Verfügung stehende Fläche ausnutzen, bis hin zu Social-Networking-Applikationen, bei denen die Eingabe von Texten eine entscheidende Rolle spielt. Mit der hohen Verbreitung im Consumer Market drängen Smartphones aber auch immer mehr in das Business- bzw. Industriesegment. Auch im Arbeitsalltag will man nicht auf den privat gewohnten Bedienkomfort verzichten und als limitiert und veraltet empfundene „Business Class“-Geräte verwenden müssen.

Verglichen mit früheren Mobiltelefonen unterscheiden sich heutige Smartphones im Besonderen durch eine ansprechend gestaltete Benutzeroberfläche, die meist mit einfachen Gesten per Finger bedient werden kann. Der zusätzliche Stylus-Pen ist zur Steuerung nicht mehr notwendig. Gleichzeitig haben sich die Interaktionskonzepte geändert, wodurch neue Wege für die Gestaltung der Benutzeroberflächen möglich wurden. Während sich bis vor wenigen Jahren Mobiltelefone hauptsächlich durch ihre technischen Merkmale unterschieden, hat sich der Fokus der Anwender hin zu einfachen und benutzerfreundlichen Oberflächen gewandt.

In diesem Beitrag sollen die Auswirkungen des beschriebenen Wandels hin zu einer mobilen Kommunikationsgesellschaft im Hinblick auf die Prozesse innerhalb größerer Firmen am Beispiel der Siemens AG erörtert werden. Die Anwenderzielgruppe stellen hier in der Regel sogenannte Professionals dar, also Service-

Techniker oder Angestellte, die Mobilgeräte in der täglichen beruflichen Praxis intern oder bei einem Kunden einsetzen.

Hierzu wird im nächsten Kapitel auf die spezifischen Anforderungen industrieller Anwendungen bezüglich mobiler Plattformen eingegangen und anhand der Automatisierungspyramide erklärt, welche Besonderheiten erst jetzt den Einsatz von Smartphones im Unternehmensumfeld als sinnvoll erscheinen lassen.

In Kap. 5.3 werden einige Beispiele mobiler Applikationen kurz vorgestellt. Es wird jeweils auf das Umfeld, die dortige Problemstellung sowie den wirtschaftlichen Nutzen der einzelnen Applikationen eingegangen. Anschließend werden in einem Ausblick die bei Siemens gerade anlaufenden Aktivitäten bezüglich App-Entwicklung angerissen sowie die nächsten Schritte vorgestellt.

## 5.2 Anforderungen an industrielle Apps

Industrielle Anforderungen an Software-Systeme und -Plattformen werden oft an einer Referenzpyramide abgetragen. In Abb. 5.1 ist diese beispielhaft dargestellt. Jede Ebene der Pyramide „löst“ dabei unterschiedliche „Probleme“ und verwendet dazu passend unterschiedliche Lösungsstrategien. Typisch ist auch, dass unterschiedliche Technologien auf der jeweiligen Ebene der Pyramide zum Einsatz kommen können.

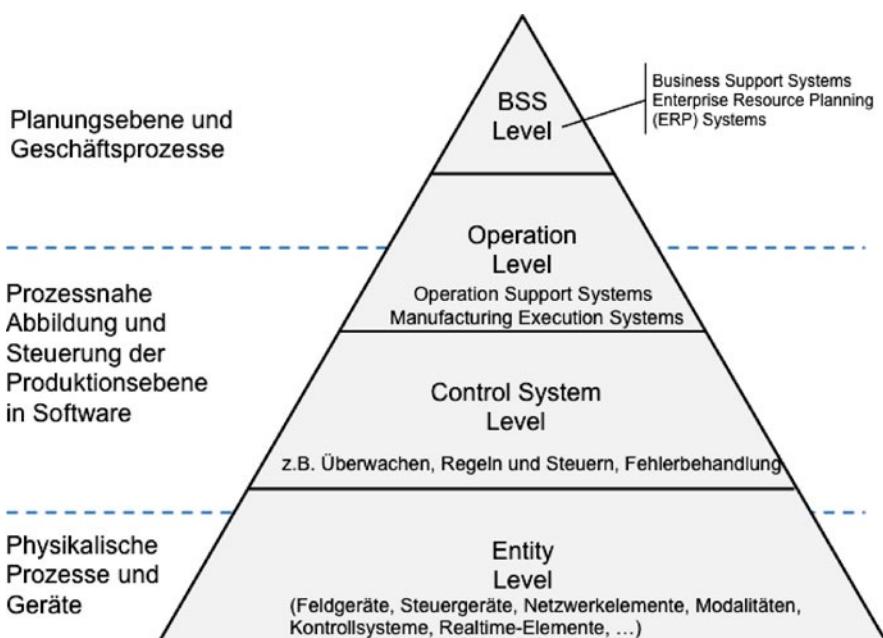


Abb. 5.1 Die allgemeine Industriepyramide als Anforderungspyramide für Siemens Mobile Apps

### 5.2.1 Charakterisierung des Problemraums

In diesem Abschnitt wollen wir anhand der Industriepyramide beispielhaft darlegen, wie sich industrielle „Apps“ von den typischen Applikationen für Endverbraucher unterscheiden. Die in Abb. 5.1 dargestellte Pyramide dient dazu, den wesentlichsten Unterschied darzustellen: Industrielle Applikationen beziehen sich auf vertikale Produkte und Märkte. Im Folgenden werden wir anhand der Ebenen erläutern, wie sich das auf die Anforderungen an die Entwicklung von Mobile Apps auswirkt.

In der Industriepyramide werden vier Ebenen unterschieden, wobei die „Breite“ der Ebene die Zahl und Komplexität der domänen spezifischen Anforderungen sowie die Relevanz für einen vertikalen Markt andeuten sollen.

Auf der untersten Ebene, der Basis der Pyramide, befindet sich die eigentliche Prozess- oder Produktionsebene (z. B. Industry- oder Energy-Sektor) beziehungsweise die Ebene der medizinischen Geräte des Healthcare-Sektors (z. B. CT-Scanner oder andere Modalitäten). Hier ist typisch, dass es sich um „Hardware“ im Sinne des Maschinenbaus oder der Elektrotechnik handelt. Anforderungen an Software auf dieser Ebene sind ebenfalls „hart“ und typisch für den Bereich der hardwarenahren oder eingebetteten Software, also z. B. harte Echtzeit, direkte Auswirkung von Einstellungen auf die Safety- und Dependability-Eigenschaften des Systems, etc. Es besteht die Notwendigkeit der weitestgehenden Kontrolle, da Eingriffe in die Software und deren Konfiguration direkten Einfluss auf die physikalischen Prozesse haben.

Auf der Ebene der Kontrollsysteme in Abb. 5.1 finden sich die typischen Systeme zur Überwachung, Steuerung und Regelung von Industrieprozessen oder medizinischen Anlagen. Auch Systeme zur Behandlung und Beseitigung von Fehlerzuständen (Diagnose) sind typischerweise auf dieser Ebene angesiedelt. In dieser Ebene erfolgt normalerweise auch die Abbildung der physikalischen Prozesse auf softwaretechnische Prozesse. Die Schnittstelle bilden oftmals Systeme der Automatisierungstechnik. Die Anforderungen hier sind strikt, dennoch nicht so hart wie auf der darunterliegenden Ebene. Dafür ist auf dieser Ebene die Varianz der möglichen Anwendungen wesentlich höher. Von einem reinen System zur Überwachung und Steuerung (z. B. SCADA), über Alarmsysteme mit automatischer Alarmprotokollierung und Prozessüberwachung, bis hin zu verteilten und dezentral organisierten Systemen finden sich hier alle Variationen. Diagnose und Wartung sind typisch für diese Ebene der Pyramide. In diesem Artikel findet dabei der Bereich der Serviceanwendungen, also Applikationen, die einen Servicetechniker bei der Wartung und Reparatur unterstützen, besondere Beachtung (vgl. iLOP und AIMES weiter unten). Aber auch Unterstützung bei der Ausführung von Arbeitsschritten in einem Krankenhaus und die Überwachung von medizinischen Geräten lässt sich auf dieser Ebene der Pyramide verorten. Typische Anforderungen an Software-Systeme auf dieser Ebene sind daher die Unterstützung verschiedenster Protokolle zur Kommunikation (z. B. OPC [4], Profinet und Profibus [5], BACNet [6] und KNX [7], etc.), Zuverlässigkeit der Kommunikation (da immer noch im kritischen Bereich der Prozesskommunikation angesiedelt, ist es somit notwendig, dass Steuerbefeh-

le auch sicher übertragen werden), Nachvollziehbarkeit der Kommunikation (z. B. Auditierbarkeitsanforderungen), Sicherheit und Schutz vor Cyber-Angriffen sowie ggf. Anforderungen an die Performanz bei kurz-zyklischen Prozessen. Insbesondere im medizinischen Bereich gesellen sich normative und gesetzgeberische Anforderungen an Zertifizierung und Normkonformität hinzu. Auf Seiten der Mobilgeräte (Tablets, Smartphones, o. Ä.) selbst bestehen Anforderungen an die Erweiterbarkeit der Hardware-Plattform (spezielle Hardware für bestimmte Prozessindustrien) und Einsatzfähigkeit unter widrigen Umständen in Industrieanlagen (z. B. Schmutz und Temperatur) oder Erfüllung hygienischer Anforderungen (z. B. Sterilisierbarkeit) bei Healthcare.

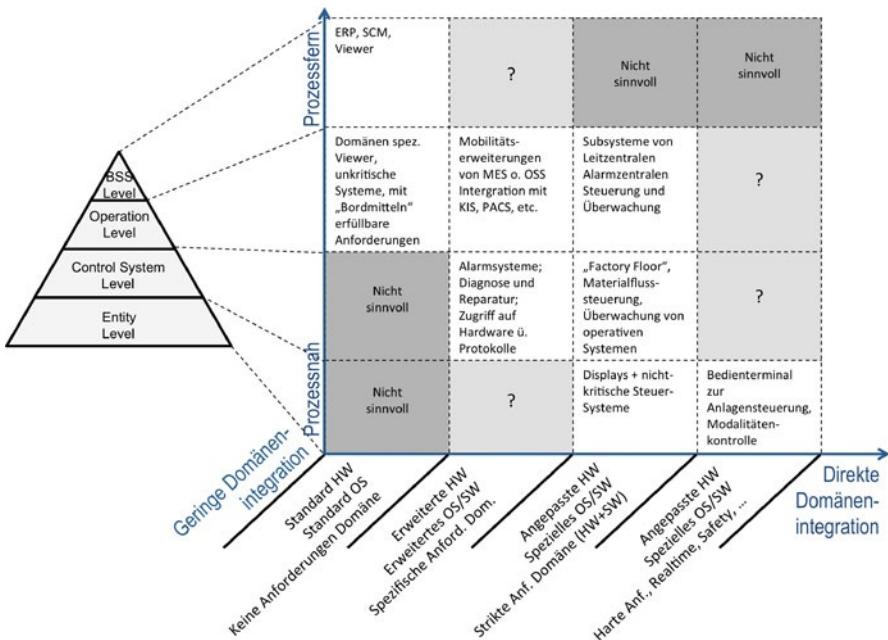
Oberhalb der Kontrollsysteme findet sich die Ebene der operationalen Systeme zur Ausführung von Produktionsprozessen und die Abbildung von Produktionsanforderungen und Stückzahlen auf Materialflüsse und Produktionsschritte (vereinfacht gesagt). Hier werden die globalen Produktionsziele auf lokale Prozessschritte abgebildet. Auf dieser Ebene ist nach wie vor ein detailliertes Wissen über die Produkte und Prozesse notwendig.

An der oberen Grenze der operationalen Systeme findet der Übergang von der operationalen Sicht auf die Geschäfts- und Planungssicht statt. Die Spitze der Pyramide bilden die sogenannten Business Support Systems (BSS), also Systeme, die unabhängig vom eigentlichen Prozess und Produkt verwendet werden können, um etwa Zulieferungen zu planen und zu kontrollieren (Supply Chain Management) oder die Produktion auf hoher Geschäftsebene nach unternehmerischen Gesichtspunkten zu optimieren.

Eine weitere Anforderung, die sich durch die geschlossene Welt der Industrieapplikationen ergibt, ist die Art und Weise der Lizenzierung und die Art der Verteilung von mobilen Applikationen. So wird eine allgemeine Applikation der BSS-Ebene sicher keine speziellen Anforderungen an Verteilung und Lizenzierung haben und über einen normalen (Siemens-)App-Store verteilt sein. Dahingegen ist es eher wahrscheinlich, dass eine spezielle Applikation der operationalen Ebene als „Bundle“ mit dem Gesamtsystem ausgeliefert wird, beispielsweise zusammen mit vorkonfigurierten Sicherheitszertifikaten oder Daten aus dem entsprechenden Engineering des Gesamtsystems. In diesem Artikel werden wir uns allerdings auf die Entwurfsanforderungen konzentrieren und haben diesen Punkt daher nur der Vollständigkeit halber erwähnt.

## 5.2.2 Charakterisierung des Lösungsraums

In Abb. 5.2 haben wir einige Beispielsysteme und -applikationen zusammengetragen und dargestellt, wie diese sich auf die jeweiligen Schichten abbilden könnten. Die jeweils geforderte Tiefe und Direktheit der Integration in die jeweilige Geschäftsdomäne bestimmt stark die Art und den Umfang der notwendigen Anpassungen, um eine mobile Hardware-Plattform, das dazugehörige Betriebssystem und den Software-Stack in die entsprechende Ebene der Pyramide zu bringen.



**Abb. 5.2** Die Integrationstiefe bestimmt die Anforderungen an eine Mobilitätsunterstützung

Zum Beispiel wird eine dispositivope Applikation auf einem iPad, also beispielsweise ein Bestell-Tool für Maschinenteile, welches über SOAP und HTTP mit dem Bestellsystem kommuniziert, keine proprietären Betriebssystemerweiterungen benötigen. Umgekehrt wird eine TÜV-zertifizierte Applikation zur Echtzeitsteuerung und -überwachung der Maschine, für die die Maschinenteile bestellt wurden, nicht auf SOAP und HTTP sowie eine komplexe Infrastruktur aufsetzen, sondern möglichst zuverlässig mit der zu steuernden Maschine direkt oder über ein SCADA-System kommunizieren.

Entsprechende Anforderungen lassen sich auch in anderen Siemens-Domänen identifizieren. Daher ist in Abb. 5.2 auf der Y-Achse der Matrix die Pyramide dargestellt sowie auf der X-Achse vier Kategorien von Mobilsystemen aufgetragen. Diese stellen qualitative Anpassungsstufen dar, die durch die Integrationstiefe in die Domäne notwendig werden können.

Gleichzeitig ergibt sich dadurch auch eine Abschätzung der Komplexität einer Integration sowie eine zusätzliche Abschätzung, welche Kombination wirtschaftlich sinnvoll sein mag und welche nicht. So erscheint es beispielsweise wirtschaftlich wenig sinnvoll, eine realzeitfähige mobile Plattform als Lösung für das oben bereits angesprochene Bestell-Tool vorzuschreiben. Hier bieten sich normale kommerzielle Produkte wie iPad oder Android Tablets an, da die Standard-Hardware und -Software ausreicht, um die Integration mit dem Bestellsystem über übliche Webservice-Schnittstellen zu realisieren.

Umgekehrt wird es nicht ohne extreme Mühen (mittels einer komplexen Infrastruktur) möglich sein, eine Standard-Hardware und -Software für eine komplexe Anlagensteuerung oder als Medizinprodukt zu zertifizieren. So haben heute z. B. iOS von Apple und auch Windows Phone 7 von Microsoft starke Einschränkungen bei der Erweiterbarkeit der Plattform selbst durch proprietäre Protokolle. Hier benötigt man offensichtlich offenere Plattformen, wie z. B. Android von Google. Wirtschaftlich ist es daher sinnvoller, für eine solche Integrationstiefe Geräte und die dazugehörige Plattform auf Basis von offenen Plattformen weiterzuentwickeln und ganz spezifisch an die Domäne anzupassen. Dazwischen gibt es „Grauzonen“ (in Abb. 5.2 mit einem Fragezeichen versehen), wo von Fall zu Fall entschieden werden muss, ob die Integration eines bestimmten Mobilsystems auf dieser Ebene der Pyramide sinnvoll umsetzbar ist oder nicht.

Zusammenfassend unterscheiden wir drei Kategorien der Erweiterungen:

- **Hardware:** Erweiterbarkeit der Hardware durch *Hinzufügen* weiterer Hardware, z. B. Smartcard-Reader, RFID, Near Field Communication, spezielle Schnittstellen und *Veränderung* der Hardware, z. B. durch „Ruggedizing“, Veränderungen von Displays, CPU, etc.
- **Software:** Erweiterung der (Standard-)Software durch *Ersetzen* von Applikationen auf einem Device oder *Hinzufügen* von Software.
- **Betriebssystem:** *Erweitern* des Betriebssystems um Treiber-Software oder Protokolle. *Ersetzen* von Betriebssystemteilen durch Anpassung an Realzeitfähigkeit oder die Veränderung der Plattform selbst.

Ein weiterer Aspekt, der berücksichtigt werden muss und der sich aus der vertikalen Integration ergibt, ist oben bereits angedeutet worden: die Möglichkeit einer Unterstützung der mobilen Applikation durch eine entsprechende gerätekionale Middleware plus einer domänenpezifischen Infrastruktur.

Im Allgemeinen hat ein aktuelles Betriebssystem heutiger Mobilgeräte bereits typische Standardfunktionalitäten eingebaut, also Standardnetzwerkprotokolle, Bibliotheken für HTTP- oder SOAP-Kommunikation über das Netzwerk, teilweise Verschlüsselung, etc. Diese „Bordmittel“ bieten eine breite Basis für die Interaktion mit einer domänenpezifischen Infrastruktur. Allerdings ist es aus ökonomischer Sicht nicht sinnvoll, jeweils jede Domänenintegration allein auf den Bordmitteln aufzusetzen, sondern es ist notwendig, eine integrationsspezifische lokale Infrastruktur auf das Gerät zu bringen, die immer wieder (standardisiert) verwendet werden kann, um auf dieser aufbauend spezifische Geschäftslogik zu implementieren. So ergibt sich ein vertikaler, domänenpezifischer Technologie-Stack, der entweder bereits auf dem Gerät als Industriestandard vorliegt oder zusammen mit der jeweiligen Applikation installiert werden muss. Abbildung 5.3 zeigt eine solche Schichtenarchitektur für typische Industrieapplikationen. Die Schichtenarchitektur zeigt im Wesentlichen, wie eine Domänenintegration aufgebaut werden kann, um eine maximale Wiederverwendbarkeit zu erreichen. Auf der obersten Ebene der spezifischen Geschäfts- und Domänenapplikationen werden entsprechende Domänenanforderungen an die Verwendbarkeit festgelegt. Hier wird auch entschieden,

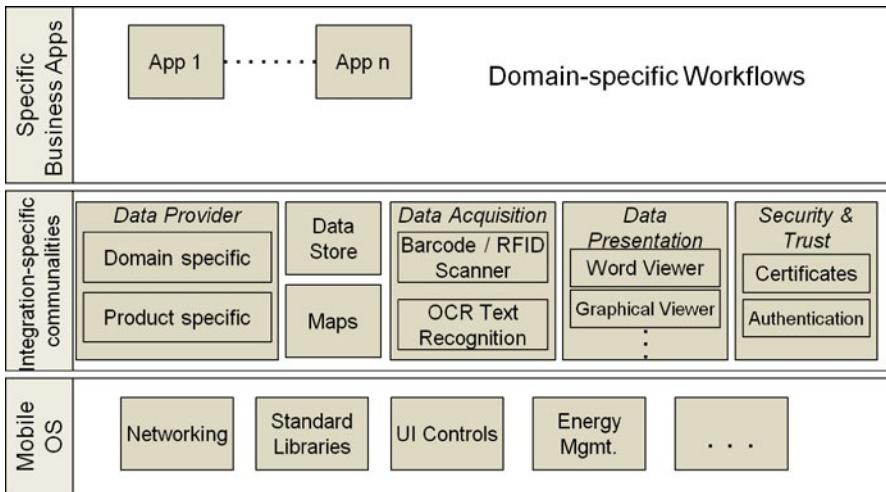


Abb. 5.3 Typischer Technologie-Stack für industrielle Mobile Apps

welcher Typ der Domänenintegration notwendig wird. Daraus leitet sich ab, welche integrationsspezifische Infrastruktur auf dem Mobilgerät notwendig ist. Typische Infrastrukturbauusteine sind spezifische Datenquellen, also Datenquellen z. B. unter Verwendung von Standardprotokollen, lokale Persistenz, Karten oder andere grafische Informationen (z. B. das Layout einer Fabrikanlage oder ein Gebäudeplan) und Bausteine zur Datenerfassung (RFID oder Barcodes, etc.). Ein weiterer wichtiger Baustein ist ein Sicherheits- und Zertifikatmodul zur Sicherung der Kommunikation. Unten in der Schichtenarchitektur findet sich das mobile Betriebssystem. Hier wird versucht, möglichst mit den Standardbausteinen der Plattform auszukommen. Je nach Integrationstiefe können hier allerdings auch spezifische Erweiterungen und Protokolle hinzukommen.

Der wesentliche Vorteil ergibt sich aus der Entwicklungseffizienz für den Entwickler von mobilen Anwendungen, da die Entwicklung der Basistechnologien unabhängig von der Entwicklung der Anwendungslogik geschehen sollte. Der Entwickler konzentriert sich ausschließlich auf die Erstellung der Applikation anhand der geschäftlichen Abläufe, und z. B. die Datenbereitstellung oder -erfassung wird standardisiert zur Verfügung gestellt.

Auf diese Weise kann eine Firma wie die Siemens AG die Applikationserstellung auf der obersten Ebene externalisieren, stark vereinfachen sowie gleichzeitig die Kontrolle über die domänen spezifischen Anteile behalten. Darüber hinaus können dadurch Qualitätsstandards und auch das Look & Feel der UI zumindest in Teilen den diesbezüglich auf Siemens ruhenden Erwartungen angepasst werden.

Nach dieser ausführlichen Betrachtung der Anforderungen an industrielle Applikationen im Rahmen der Industriepyramide wollen wir im folgenden Teil dieses Artikels einige Applikationen vorstellen und diskutieren, die bei Siemens als Prototypen oder Produkte entwickelt wurden.

## 5.3 Beispiele innerhalb von Siemens

Nachfolgend werden einige ausgewählte Apps der Siemens AG vorgestellt. Diese stellen nur einen kleinen exemplarischen Ausschnitt dar, liefern jedoch zugleich einen guten Überblick über die relevanten Anwendungsgebiete.

### 5.3.1 Teamcenter Mobility

Product-Lifecycle-Management (PLM) wird zum Verwalten von Produkten über den ganzen Lebenszyklus hinweg eingesetzt. Siemens ist führender, weltweit tätiger Anbieter von Product Lifecycle Management-Software und zugehöriger Dienstleistungen. Siemens Teamcenter [8] ist dabei de facto das meist genutzte PLM-System in der Automobil- und Luftfahrtindustrie und wird auch in anderen Branchen, wie zum Beispiel der Konsumgüter- und der High-Tech-Elektronikindustrie sowie im Schiffbau häufig eingesetzt.

#### 5.3.1.1 Problemstellung

Einsatzbereiche von PLM im Lebenszyklus eines Produkts wurden durch Integration von immer weiteren Produkt- und Entwicklungsinformationen (Produktdatenmanagement) in den letzten Jahren stetig erweitert. In den verschiedenen Phasen eines Produkts, von Planung, Entwurf und Entwicklung bis hin zur Außerbetriebnahme, müssen tausende von Entscheidungen gefällt werden und viele dieser Entscheidungen erfordern Informationen über aktuelle Produktdaten zur richtigen Zeit, am richtigen Ort und im richtigen Kontext. Informationen werden nicht nur in den Planungs- und Entwicklungsbüros, sondern auch z. B. bei Einkaufs- und Kunden gesprächen, Inbetriebnahme und Wartung benötigt.

#### 5.3.1.2 Beschreibung

Immer mehr Entscheidungsträger sind heutzutage mobil und benötigen dazu die aktuellsten Daten vor Ort. Tablets haben durch das iPad von Apple und die Android-Plattform an Popularität gewonnen. Mit Teamcenter Mobility verfügt Siemens bereits über eine Lösung für solche Tablet Computer und ermöglicht den mobilen und unmittelbaren Zugriff auf die für Entscheidungen benötigten aktuellen Informationen, welche mittels Teamcenter verwaltet werden. Benötigte Daten (etwa 2D-Zeichnungen, 3D-Daten, Analyseergebnisse und Spezifikationen) können bei Bedarf auf das Mobilgerät heruntergeladen, offline bearbeitet und später automatisch mit Teamcenter synchronisiert werden.



**Abb. 5.4** Teamcenter Mobility auf einem iPad

### 5.3.1.3 Mehrwert

Da die Produktentwicklung immer mobiler wird, muss sie auf die Nutzung von Mobiltechnologie setzen. Die Stärken von Ad-hoc-Nutzung, intuitiv nutzbarer Oberfläche und dem richtigen Maß an zugeschnittener Funktionalität bei langen Nutzungszeiten sind ideal z. B. für Ingenieure in Anlagen und Montageeinrichtungen, Servicetechniker vor Ort bei Kundenbesuchen und Mitarbeiter, die in und an großen Objekten wie Schiffen oder Flugzeugen arbeiten. Teamcenter Mobility ermöglicht das Treffen fundierter Entscheidungen, unabhängig davon, wann und wo diese Entscheidung notwendig wird.

### 5.3.2 *iLop*

Siemens ist weltweit einer der wenigen Anbieter von Gepäckförderanlagen für Flughäfen aller Größenordnungen. Insbesondere sehr große Flughäfen erfordern bei der Planung, dem Aufbau sowie der Installation der Gepäckförderanlage umfassendes Know-how und einen reichhaltigen Erfahrungsschatz, um derartige Projekte erfolgreich abschließen zu können.

### 5.3.2.1 Problemstellung

Die Installation einer Gepäckförderanlage auf dem Flughafen wird durch erfahrene Spezialisten kontinuierlich überwacht und betreut, um auftretende Probleme erfassen und möglichst schnell lösen zu können. Nur so kann ein reibungsloser Aufbau der Anlage, der typischerweise durch einen sehr engen Zeitrahmen begrenzt ist, gewährleistet werden ohne dabei den Installationsfortschritt zu gefährden.

Werden Probleme beim Aufbau festgestellt, wurden diese bisher auf einer Papiervorlage notiert und regelmäßig in ein System zur Verwaltung offener Punkte übertragen. Hierbei fällt es oft schwer, ein Problem durch eine textuelle Erläuterung adäquat zu beschreiben. In vielen Fällen würde ein Bild oder ein kurzes Video sehr viel mehr und genauere Informationen liefern, als mit vertretbarem Aufwand schriftlich erfasst werden kann.

### 5.3.2.2 Beschreibung

Die mobile Applikation iLop („Liste offener Punkte“) wurde genau zu diesem Zweck entwickelt. Sie soll helfen, offene Punkte einfach erfassen zu können, mit entsprechendem Bild- und Videomaterial anzureichern und diese Information über eine automatische Synchronisation allen Interessenten zugänglich zu machen.

Neben der Erfassung offener Punkte werden durch die App auch Dokumente zu den entsprechenden Komponenten innerhalb der Applikation zur Verfügung gestellt, so dass zu einem offenen Punkt direkt die zugehörigen Unterlagen angezeigt werden können.

Darüber hinaus kann ein offener Punkt an eine Kontaktperson per Mail verschickt werden, um Unterstützung oder Rat anzufordern. Da die Daten auf einem Server synchronisiert werden, kann jeder Anwender auf die erfassten Daten zugreifen und die ihm zugeordneten offenen Punkte bearbeiten.

### 5.3.2.3 Mehrwert

Die mobile Applikation iLop wurde von ersten Anwendern sehr positiv aufgenommen und wird trotz noch nicht vollständig abgeschlossener Entwicklung bereits in einem laufenden Projekt produktiv erprobt. Der Mehrwert erschließt sich nicht nur aus der Zeitsparnis für das Erfassen, Weiterverarbeiten und Verteilen der offenen Punkte, sondern auch in der deutlich höheren Qualität der Informationen, die durch Bild- und Videomaterial aussagekräftiger werden. Besonders bei schwierig zu beschreibenden Punkten bietet dies einen enormen Vorteil bei der Beurteilung durch einen Experten, da langwieriges Rückfragen entfällt.

Die automatische Verknüpfung der offenen Punkte mit den zugehörigen technischen Unterlagen ermöglicht einen einfachen und medienbruchfreien Zugang zu den relevanten Informationen, die in einem handlichen Gerät jederzeit verfügbar sind.



**Abb. 5.5** Ein Beispiel für die Verwaltung offener Punkte mit iLop

### 5.3.3 AIMES

AIMES steht für „Advanced Infrastructure for Medical Equipment Management and Services“ und ist ein abgeschlossenes Förderprojekt des Bundesministeriums für Bildung und Forschung (BMBF) und der ITEA2 [9], das von Siemens in einem deutsch-spanischen Konsortium geleitet wurde. Innerhalb der Siemens AG wurde das Projekt von der Zentralen Forschung und Entwicklung zusammen mit dem Healthcare-Sektor durchgeführt. Informationen zum Förderprojekt finden sich auf der Webseite von AIMES [10].

#### 5.3.3.1 Problemstellung

Das Problem, welches sich im Rahmen des Förderprojektes ergeben hatte, ähnelt stark dem, das iLop aus Abschn. 5.3.2 gelöst hat. Die Anwendung versucht im Rahmen einer geschlossenen Informationskette Medienbrüche zwischen der Erfassung von Reparatur-Aufträgen in einem Krankenhaus und der Abarbeitung durch einen Mitarbeiter des Hersteller-Services zu vermeiden. Heute ist es weitestge-



Abb. 5.6 Aimes Hospital Mobile App

hend so, dass Störungen an Medizingeräten im Krankenhaus von dortigen Service-Spezialisten manuell erfasst werden und dann per Fax, Telefon, Email oder Briefpost schließlich an den Hersteller-Service als Störungsmeldung weitergeleitet werden (wenn der Kunde nicht fortschrittliche Remote-Service-Dienstleistungen z. B. von Siemens in Anspruch nimmt). Im Endeffekt wird eine einzelne Störung in verschiedenen IT-Systemen und auf Papier erfasst und verwaltet. Das gleiche gilt ggf. für die Rechnungsstellung. Diese Situation wird von den Krankenhäusern und den Herstellern als unbefriedigend empfunden und eine weitestgehend IT-technische Erfassung ist wünschenswert. Dazu wurde ausgelotet, wie weit eine Ende-zu-Ende-Lösung gehen kann, unter Einbezug mobiler Applikationen und Geräte. Weiterhin wurde festgeschrieben, dass im Rahmen der Applikation im Förderprojekt nur Standard-Smartphones, in diesem Falle Android-Phones, zum Einsatz kommen dürfen und diese nur im Rahmen der normalen Erweiterbarkeit über Apps erweitert werden dürfen.

### 5.3.3.2 Beschreibung

Die Gesamtapplikation besteht aus zwei einzelnen mobilen Apps, die über einen Enterprise Service Bus (hier JBoss ESB [11]) und ein SAP-Hintergrundsystem der Siemens AG zusammengeschaltet werden können. Weitere Informationen zu AIMES und der Integration mittels ESB finden sich in [12]. Im Folgenden werden die beiden Applikationsteile kurz charakterisiert.

#### Hospital-Applikation

Die erste Applikation ist die zentrale Applikation zum Erfassen einer Störung (primärer Nutzen), zum Anzeigen und Filtern aller schon erfassten Störungen im System (zu einem Medizingerät) und zur einfachen „Grundinventarisierung“ von neuen

Geräten oder Teilen eines Gerätes. Darüber hinaus hat der Service-Techniker die Möglichkeit, ein sogenanntes „Multimedia Trouble Ticket“ zu erfassen, d. h. er kann zu einer Störung sowohl Fotos als auch Text, Video und gesprochene Beschreibung aufzeichnen und diese als Teil der Störung mitsenden. Dadurch hat der Hersteller-Service bereits alle hilfreichen Informationen, bevor eine Störungsbehebung geplant oder versucht wird.

Besonderes Augenmerk wurde hier auf die ausschließliche Verwendung von Standard-Protokollen zur Kommunikation (HTTP, SOAP, Webservice) und Standard-Smartphones gelegt, damit die Einstiegshürde für den Benutzer und Systemverwalter besonders niedrig ist. Da angenommen wird, dass sich der Benutzer dauerhaft an der Infrastruktur angemeldet hat, funktioniert die Datenhaltung in der Infrastruktur und das Gerät dient im Weiteren lediglich zur Erfassung und Online-Anzeige der Daten.

### Service-Applikation

Das Gegenstück ist die Service-Applikation, die der Herstellerservice verwendet. Hier ist vorgesehen, dass der Service-Techniker Einsatzlisten verwenden kann, um seine Arbeitseinsätze zu planen und darüber hinaus die vorgenannten Multimedia Trouble Tickets im Original auf sein Gerät herunterladen kann. Alle Informationen (Fotos, Video, Sprache, Text) liegen vor und können angezeigt werden. Ebenfalls kann auf Hintergrundsysteme zugegriffen werden, um weitere herstellerspezifische Datenhaltungen einzubinden (konzeptionell vorgesehen, im Prototypen nicht realisiert). Wesentliches Augenmerk wurde hier auf die lokale Datenhaltung gelegt und die „Offline“-Integration, da angenommen wird, dass der Service-Techniker nicht dauerhaft mit der Siemens-Infrastruktur verbunden ist und daher einige Daten vorhalten muss. Auch wenn der Prototyp ebenfalls auf einem Android-Smartphone entwickelt wurde, wäre hier die Integration weiterer Hardware und Protokolle (z. B. RFID, Smartcard, Protokollmodule zur Messdaten-Erfassung von Geräten) wünschenswert sowie die Auslegung als „ruggedized“ Terminal.

#### 5.3.3.3 Mehrwert

Die Integration der vorher einzelnen Arbeitsschritte in eine durchgehende Unterstützung durch Mobilapplikationen ohne die Gefahr von Medienbrüchen und der daraus resultierende Zeit- und Konsistenzgewinn ist ein wesentlicher Mehrwert des AIMES-Demonstrators für Siemens.

#### 5.3.4 *syngo.via WebReport*

*syngo.via* ist ein klinisches Bildverarbeitungssystem, das sich durch Nutzung von Standardprotokollen nahtlos in eine bestehende Klinikinfrastruktur einpassen lässt.

Es empfängt Bilder von radiologischen Aufnahmegeräten wie CT oder MR und präsentiert diese nach einer workflow-abhängigen, automatisch ablaufenden Vorverarbeitung dem befundenden Arzt. Die Vorverarbeitung wird vom Anwender konfiguriert.

#### 5.3.4.1 Problemstellung

Mittels *syngo.via* wird der Radiologe in die Lage versetzt, medizinische Bildinformationen an dedizierten Rechenstationen abzurufen und zu bearbeiten. Für eine optimale Kommunikation mit dem Patienten, z. B. im Rahmen der Befunderläuterung, ist es wünschenswert, wenn die vorgenannten Daten „überall“ verfügbar sind. Die Unabhängigkeit vom Ort ist tatsächlich bereits mit Computern im Notebook-Format erreichbar, jedoch ist dieser Formfaktor durch seine Größe und das Handling (Auf- und Zuklappen) nicht sinnvoll in der ärztlichen Praxis einsetzbar. Dies gilt insbesondere für die Verwendung als Kommunikationsmedium im Patientengespräch.

#### 5.3.4.2 Beschreibung

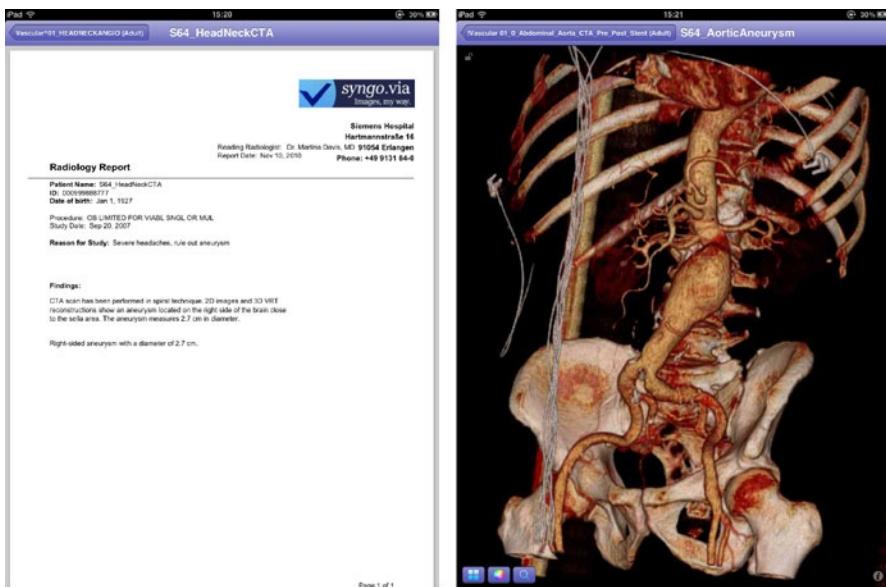
*syngo.via WebReport*<sup>1</sup> ist eine Ergänzung des Bildverarbeitungssystems, das die sichere Verteilung von diagnostischen Reports und medizinischen Bilddaten speziell auch auf mobilen Endgeräten der Firma Apple<sup>2</sup> (iPhone, iPod Touch und iPad) erlaubt. Nach Anmeldung am System präsentiert sich dem Benutzer eine Liste der im System verfügbaren Patienten, über die sowohl Text- als auch Bildinformationen zu einem Fall abgerufen werden können. Neben der Darstellung von zweidimensionalen Schnittbildern hat der Benutzer vollen interaktiven Zugriff auf 3D-Darstellungen (siehe Abb. 5.7). Durch den Einsatz von Verschlüsselungsmethoden ist die Nutzung nicht auf das interne Netzwerk des Betreibers beschränkt, eine Verbindung mit dem Server kann auch über das Internet aufgebaut werden.

#### 5.3.4.3 Mehrwert

Der Zugriff auf diagnostische Reports und medizinische Bilddaten in 2D und 3D sowie entsprechende interaktive Bildbearbeitungsoperationen auf mobilen Geräten gibt dem Arzt eine größere Bandbreite in der Kommunikation zum Patienten. Die

<sup>1</sup> *syngo.via WebReport* ist nicht für diagnostische Zwecke bestimmt. Das Produkt kann überall da eingesetzt werden, wo die folgenden Voraussetzungen vorliegen: WLAN-Anbindung an das klinische Netzwerk, Erfüllung der empfohlenen Hardware-Mindestanforderungen und die Einhaltung geltender Datenschutzbestimmungen.

<sup>2</sup> Apple®, iPhone™, iPad™, iPod Touch™ sind Schutzmarken von Apple Inc., und in USA und weiteren Ländern registriert. Ihre Verwendung kann landesspezifischen gesetzlichen Bestimmungen unterliegen - daher müssen diese jeweils für die medizinische Verwendung geprüft werden.



**Abb. 5.7** Login-Bildschirm, diagnostischer Report und interaktive 3D-Darstellung

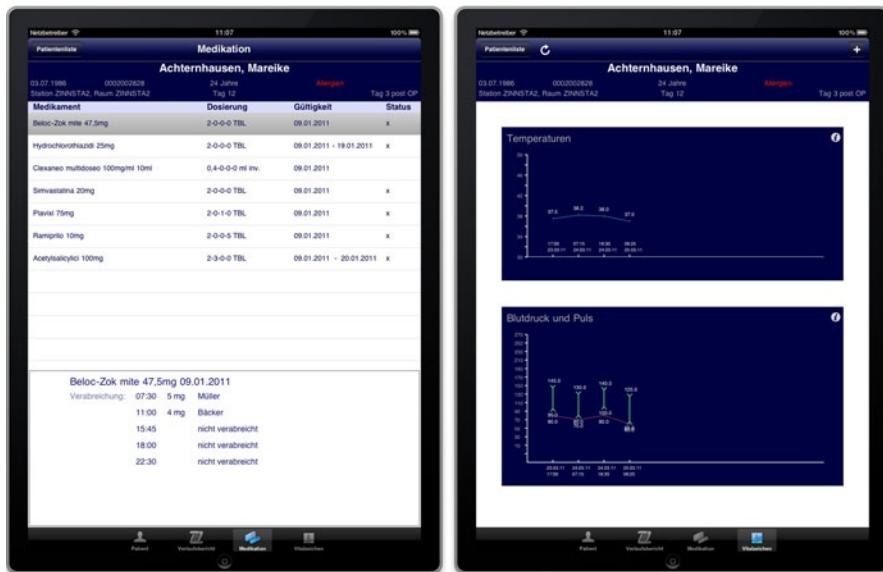
Bindung an lokale Arbeitsplatzstationen entfällt. Die Daten können auf kleinen Geräten mir dem Formfaktor „iPad“ direkt zum Patienten mitgenommen werden. Durch die Möglichkeit des sicheren Zugangs über das Internet ist sowohl eine räumliche als auch eine zeitliche Beschränkung beim Zugriff auf die Bilder aufgehoben worden. Dadurch erhält der Arzt sehr viel mehr Flexibilität im Umgang mit seinen Daten.

### 5.3.5 i.s.h.med® Visite

i.s.h.med® ist ein integriertes und auf SAP for Healthcare basierendes Krankenhausinformationssystem, mit dem Fokus auf Dokumentation, Kommunikation und Planung im klinischen Prozess. Der mögliche Einsatzbereich erstreckt sich dabei über die klassischen klinischen Prozesse wie z. B. Anordnung, Befundschreibung, Medikation, bis hin zum Bettenbelegungsmanagement.

#### 5.3.5.1 Problemstellung

Es ist jedoch weder möglich noch sinnvoll, alle Funktionen eines modernen Krankenhausinformationssystems auf mobile Geräte abilden zu wollen. Einige Szenarien, etwa die Patientenversorgung auf der Station sowie vor allem die Dokumentation von Beobachtungen und geplanten Maßnahmen, stellen allerdings bereits für



**Abb. 5.8** Ansicht von Medikation sowie Vitalwerten mittels i.s.h.med® Visite

sich selbst genommen hochgradig mobile Prozesse dar. Diese finden in der Regel direkt am Krankenbett statt. Um tägliche Prozesse dieser Art wirksam unterstützen zu können, ist es durchaus sinnvoll, hier ebenfalls mobile Lösungen für Information und Dokumentation anzubieten.

### 5.3.5.2 Beschreibung

In enger Zusammenarbeit mit den Asklepios-Kliniken entstand die i.s.h.med® Visite [13], sie ergänzt die patientennahe Planung und Dokumentation auf einem mobilen Endgerät. Dabei kann rollenübergreifend (also durch Arzt oder beispielsweise Pflegekraft) und bidirektional (sprich lesend und schreibend) auf die elektronische Krankenakte in i.s.h.med® zugegriffen werden. Somit können etwa bereits während einer Visite therapeutische Maßnahmen angeordnet und Verlaufsnachrichten erfasst werden. Auf etwaige Risikofaktoren und Allergien wird direkt auf dem mobilen Gerät hingewiesen, eine Übersicht aller aktuellen Medikamentenverordnungen kann abgefragt werden. Vitalwerte, in ihrem zeitlichen Verlauf dargestellt, erleichtern darüber hinaus die Interpretation und Plausibilitätskontrolle aktueller Messwerte. Die Sicherheit der patientenbezogenen Daten wird durch verschlüsselte Kommunikation und nicht auf dem Client persistente Datenhaltung gewährleistet.

Die i.s.h.med® Visite wurde im April 2011 auf der conhIT in Berlin in Form eines Prototyps vorgestellt. Eine Überführung in das i.s.h.med® Produktpotfolio ist nach erfolgreicher Pilotphase realistisch, ebenso eine Anbindung an die weiteren Siemens-Krankenhausinformationssysteme Soarian® Clinicals und medico.

Auf lange Sicht könnte die Anwendung somit das klassische Klemmbrett des Arztes bei der Visite ablösen.

### 5.3.5.3 Mehrwert

Die klinische Dokumentation findet bislang noch weitgehend auf Papier statt. Informationen sind dabei nicht immer sofort auffindbar, handschriftliche Notizen teilweise schlecht lesbar. Durch häufige Medienbrüche, etwa das Übertragen einer Papierkurve in die elektronische Krankenakte, können sich Fehler einschleichen, welche zu einer Verminderung der Datenqualität führen. Die i.s.h.med® Visite versucht hierbei eine möglichst lückenlose elektronische Dokumentation anzubieten, ohne dabei auf die Übersichtlichkeits- und Unmittelbarkeitsvorzüge von Papier zu verzichten. Dadurch sind Daten sofort und systemweit verfügbar, Fehler beim Übertragen werden reduziert.

## 5.4 Ausblick

Die oben vorgestellte Auswahl verdeutlicht, dass erste Apps bei der Siemens AG bereits Einzug gehalten haben und bei Kunden meist auf sehr großes Interesse gestoßen sind. Teilweise sind diese sogar schon zu öffentlich zugänglichen Erweiterungen eines Produktes geworden.

Die Relevanz des Themas zeigt sich darüber hinaus auch an der Menge von Prototypen sowie regen Diskussionen in internen Netzwerken, welche sich speziell zu dieser Thematik innerhalb der Siemens AG gegründet haben. Die von den Medien häufig angeführte „Gründerzeitstimmung“ bei der App-Entwicklung spiegelt sich also auch fernab von Start-ups und selbstständigen Entwicklern wider und scheint die großen Konzerne auch abseits des Marketing erreicht zu haben.

Aus industrieller Sicht wirft dies jedoch auch gänzlich neue Fragestellungen rund um diesen jungen Themenkomplex auf. So werden etwa Apps (innerhalb der jeweiligen mobilen Plattform und deren zentralen Stores) als gemeinsames Aushängeschild einer Firma wahrgenommen werden. Ein unreifer Prototyp in freier Wildbahn kann dabei schnell und ungewollt rufschädigende Auswirkungen nach sich ziehen und das professionelle Image der Firma beschädigen. Prototypen für Messen und interne Zwecke sollten daher gar nicht erst den Weg in einen offiziellen App-Store finden. Dadurch sind neue Prozesse notwendig, etwa um Apps über einen zentralen Account zu veröffentlichen, anstatt über viele separate Zugänge der jeweiligen Entwicklungsabteilungen. Ebenso müssen solche Anwendungen eine einheitliche UI-Sprache aufweisen, welche zwar produktspezifisch durchaus variieren kann, der Corporate Identity aber nicht entgegensteht. Konkrete Produkte oder deren Erweiterungen auf den mobilen Plattformen sollten des Weiteren eindeutig als solche identifizierbar sein, ebenso wie reine Marketing-Apps für Veranstaltungen oder andere Zwecke. Dieser Prozess wird gegenwärtig in der Siemens AG etabliert,

kann jedoch noch nicht auf alle Szenarien und Plattformen vollständig abgebildet werden.

Auf lange Sicht könnten Plattformen entstehen, mit deren Hilfe ein breites Spektrum an Anwendungsfällen abgedeckt werden kann. Häufig benötigte Komponenten könnten dabei in Form von wiederverwendbaren Building Blocks zur Verfügung gestellt werden. Mittels solcher Plattformen könnte auch Drittanbietern das Entwickeln von Apps für Industrie-Produkte ermöglicht werden. Mit der besseren Unterstützung bei der Entwicklung und dem Bereitstellen von Apps sind aber auch spezifische Applikationen möglich, die mehr als nur eine Erweiterung eines bestehenden Produktes darstellen. So ist es denkbar, bisher stationäre Konsolen und Anzeigen durch mobile Endgeräte zu ersetzen. Eine direktere Integration der mobilen Apps innerhalb der Industriepyramide wird denkbar.

Um solch zukünftigen Anwendungen wie auch Prototypen den Weg zur Produktifizierung zu erleichtern, wollen wir im nächsten Schritt zunächst unsere Erwartungen und Vermutungen hinsichtlich der Anforderungen anhand einer internen Umfrage mit nachgelagerter Studie überprüfen und wenn nötig korrigieren. Hierbei gilt es den Bedarf an Apps auch speziell für die interne IT- und Prozesslandschaft zu überprüfen, sowie zu evaluieren, inwieweit etwa bestehende Web-Applikationen einfach und übersichtlich auf mobilen Endgeräten verfügbar gemacht werden können.

## 5.5 Zusammenfassung

In diesem Artikel haben wir das aktuelle Thema „Mobile Apps“ aus der Sicht eines großen deutschen Industriekonzerns betrachtet und die Frage beantwortet, was getan werden muss, um die spezifischen Anforderungen der unterschiedlichen Siemens-Domänen adäquat durch mobile Applikationen adressieren zu können. Dabei sind in Abschn. 5.2 die Anforderungen an Apps aus Sicht der klassischen Industriepyramide aufgestellt worden und es wurde dazu passend ein Lösungsraum skizziert. Dabei ist klar geworden, dass es keinen „One-Size-Fits-All“-Ansatz für alle Domänen und alle Schichten der Pyramide geben kann.

Dennoch sehen wir eine breite Basis für eine Art industrielle Middleware-Schicht auf mobilen Endgeräten, die es den einzelnen Benutzern industrieller Anwendungen ermöglicht, die Vorteile der Smartphones und Tablets zu nutzen und gleichzeitig eine gute Integration mit den industriellen Domänen zu erreichen. Etwas, was heute so nicht bekannt ist.

Wir haben weiterhin eine Reihe einschlägiger Applikationen aus unterschiedlichen Geschäftsbereichen der Siemens AG vorgestellt und so auf breiter Basis motiviert, dass das Thema für Siemens als Portfolioergänzung angekommen ist und für weiter ausbaufähig erachtet wird. Allerdings sieht man auch, dass es hier durchaus noch Arbeit gibt, insbesondere im Bereich der Standardisierung der Applikations- und Plattformentwicklung und der Geschäftsmodelle.

## 5.6 Danksagung

Wir wollen uns abschließend bei einigen Kollegen für die Unterstützung, insbesondere bei der Beschreibung der ausgesuchten Apps aus den Siemens-Sektoren, bedanken:

- Peter Osburg und Bernhard Kuhlicke, Healthcare Sector Customer Solutions (i.s.h.med® Visite)
- Hans-Martin von Stockhausen, Healthcare Sector Imaging & Therapy Systems Division (*syngo.via*)
- Timo Wolf, Corporate Technology (iLop)

## Literaturverzeichnis

1. Eclipse Foundation (2011) Open source developer report 2011. <http://www.eclipse.org> (Oktober 2011)
2. VisionMobile (2011) Developer economics 2011, study. <http://www.visionmobile.com/devecon.php> (Oktober 2011)
3. Java Specification. <http://jcp.org/aboutJava/communityprocess/final/jsr139/index.html> (Oktober 2011)
4. OPC Foundation, <http://www.opcfoundation.org/> (Oktober 2011)
5. Profibus and Profinet International. <http://www.profibus.com/index.php?id=9> (Oktober 2011)
6. Official Website of ASHRAE SSPC 135. <http://www.bacnet.org/> (Oktober 2011)
7. KNX Association. <http://www.knx.org/> (Oktober 2011)
8. Teamcenter. <http://www.siemens.com/teamcenter/> (Oktober 2011)
9. ITEA2. <http://www.itea2.org/> (Oktober 2011)
10. AIMES-Konsortium (2011) Advanced infrastructure for medical equipment and service. <http://www.aimes-project.eu/> (Oktober 2011)
11. JBoss-ESB. <http://www.jboss.org/jbossebs> (Oktober 2011)
12. Zeidler A, Baumgarth T, Mittermeier L, Meunier R (2010) Applied soa paradigms for medical equipment maintenance. In: Proceedings of IADIS International Conference e-Health 2010. IADIS
13. Siemens AG (2011) Smart UI: Mit dem iPad auf Visite. In: *inside:health – IT Special*, pp 16–19

---

# Kapitel 6

## Neue Wege in das Internet der Dinge durch Smart Mobile Apps

Matthias Wagner, Paul Holleis und Hendrik Berndt

**Zusammenfassung** Der Begriff des Internet der Dinge („Internet of Things“) bezeichnet die Verknüpfung unserer physikalischen Welt mit Informationen und Diensten der digitalen Welt. Durch diese Verknüpfung von realen Objekten mit Internet-Diensten wird ein breites und hochinnovatives Spektrum von neuen Anwendungen möglich. Hierbei wird die automatische Identifikation von physikalischen Objekten mittels Auto-ID und RFID-Technologie häufig als wichtige Grundlage angesehen. Insbesondere steht mit Near Field Communication (NFC) eine RFID-Technologie bereit, die den Einsatz von mobilen Endgeräten besonders flexibel und leicht möglich macht. In diesem Kapitel bieten wir dem Leser einen Überblick über angewandte Arbeiten von DOCOMO Communications Laboratories Europe GmbH im Bereich Auto-ID und NFC mit dem Ziel des vereinfachten mobilen Zugangs zum Internet der Dinge. Zielsetzung hierbei ist die Unterstützung einer zukünftigen Generation von Systemen für das mobile Geschäftsprozessmanagement.

---

Matthias Wagner  
DOCOMO Communications Laboratories Europe GmbH  
München,  
E-mail: wagner@docomolab-euro.com

Paul Holleis  
DOCOMO Communications Laboratories Europe GmbH  
München,  
E-mail: holleis@docomolab-euro.com

Hendrik Berndt  
DOCOMO Communications Laboratories Europe GmbH  
München,  
E-mail: berndt@docomolab-euro.com

## 6.1 Einleitung

Unzählige Gegenstände tragen heute maschinenlesbare, individuelle Bezeichnungen in Form von Klartext, Barcodes oder elektronisch gespeicherten Informationen. Diese Maschinenlesbarkeit der Namen von Objekten und Dingen ist für die Automatisierung von Geschäftsprozessen von zentraler Bedeutung. Zu diesen Gegenständen gehören beispielsweise Waren im Einzelhandel, Postsendungen, Maschinenteile, Werkstücke, Transportcontainer oder Ausweisdokumente.

Auto-ID- und Radio Frequency Identification (RFID)-Systeme machen diese physikalische Welt der Dinge für Informationssysteme zugänglich. Dadurch können relevante Daten direkt am Objekt und aus dem Prozess heraus erfasst werden, um dann ihrerseits Prozesse steuern und optimieren zu können.

Nach einer kurzen Einführung in den Stand der Technik für das Geschäftsprozessmanagement mit Unterstützung durch Auto-ID- und RFID-Systeme bietet dieses Kapitel einen Überblick über ausgewählte erweiterte Einsatzmöglichkeiten von RFID-Technologie in mobilen Diensten und Endanwendungen. Der Fokus liegt dabei auf angewandten Forschungsarbeiten und prototypischen Anwendungen mit dem Ziel des vereinfachten mobilen Zugangs zum Internet der Dinge. Besondere Berücksichtigung findet hierbei die Entwicklung intuitiver, kostengünstiger und leicht zugänglicher Lösungen durch mobile Endgeräte und Anwendungen.

Der Inhalt dieses Kapitels ist wie folgt gegliedert: eine kurze Übersicht über die Prozessunterstützung und das Geschäftsprozessmanagement im Internet der Dinge mit Hilfe von Auto-ID und RFID-Systemen sowie den zukünftigen Einsatz von Near Field Communication (NFC) zeigt Abschn. 6.2 auf. Abschnitt 6.3 vermittelt einen praktischen Eindruck von zukünftigen Herausforderungen und Forschungsfragen, die sich insbesondere zusätzlich zu den etablierten Kernanwendungen von Auto-ID und RFID ergeben. Abschn. 6.4 diskutiert benachbarte Forschungsfelder und Projekte, bevor Abschn. 6.5 mit einem Ausblick abschließt.

## 6.2 Aufbruch in das Zeitalter der mobilen Geschäftsprozesse

Die automatische Identifikation von physikalischen Objekten zur Prozessunterstützung umfasst die Vergabe, Zuordnung, Übermittlung und Verarbeitung von Objektbezeichnungen. Danach stehen die jeweiligen Ergebnisse für Informationszwecke und weitere Auswertungen zur Verfügung.

Entscheidend ist, dass Abläufe und Zustände aus der realen Welt direkt in die Welt der Informationssysteme (IT) abgebildet werden können. Hieraus ergeben sich enorme Vorteile für die gesamte Wertschöpfungskette von der Produktion über die Logistik bis hin zum Endverbraucher.

### 6.2.1 Prozessunterstützung durch Auto-ID- und RFID-Systeme

Auto-ID- und RFID-Systeme machen die physikalische Welt von Objekten und Dingen bereits heute für IT-Systeme zugänglich. Dadurch können relevante Daten direkt am Objekt aus dem Prozess erfasst werden, um dann wiederum den Prozess steuern und optimieren zu können. Auto-ID- und RFID-Systeme ermöglichen es somit, Daten mit einem individuellen Objekt zu verknüpfen. Diese meist mobilen Objekte durchlaufen dabei eine vernetzte Infrastruktur und steuern auf diese Weise auch Geschäfts- und Fertigungsprozesse [1, 2].

Im Bereich Auto-ID am weitesten verbreitet sind heute optische Barcodes mit einem geschätzten Marktanteil von 75 % am Gesamtaufkommen von Identifikationssystemen [3]. Die Symbole werden von Scannern erfasst, die den Barcode anstrahlen und das reflektierende Licht messen. Die enthaltenen Informationen werden decodiert und von IT-Systemen verarbeitet. Weltweit werden Milliarden von Objekten mit Hilfe von optischen Codes ausgezeichnet. Prominentestes Beispiel ist die Verwendung von eindimensionalen Barcodes zur durchgängigen Bezeichnung aller Artikel in Supermärkten. In der industriellen Fertigung haben sich dagegen 2D- oder Data-Matrix-Codes etabliert. Gründe hierfür sind die höhere Speicherdichte, größere Robustheit und die flexibleren Aufbringungsmöglichkeiten auf den verschiedensten Materialien und Oberflächen.

Neben diesen Systemen spielt auch RFID [4, 5] eine wichtige Rolle. Sie besitzt gegenüber anderen Systemen entscheidende Vorteile, was Robustheit und Einsatzmöglichkeiten angeht. Ein RFID-System besteht mindestens aus einem Lesegerät und einem mobilen Transponder als Datenspeicher, der unter Verwendung eines hochfrequenten Übertragungsverfahrens kontaktlos, und insbesondere auch ohne Sichtkontakt, gelesen werden kann.

Den beiden vorgestellten Systemen ist jedoch eines gemein: Dank Auto-ID- und RFID-Systemen haben sich industrielles Prozessmanagement und Wertschöpfungskette drastisch verändert. Bestanden Produktion und Zulieferung bisher vorwiegend aus gleichartigen Komponenten, geht der nachhaltige Trend nunmehr hin zur wissensbasierten Zulieferung und Fertigung. Ein Beispiel hierfür ist die Automobilindustrie, wo ganze Baugruppen und Produktionssegmente vorgefertigt angeliefert werden. Agilität, schnelle Reaktionszeit und die Möglichkeit, flexibel auf Änderungen beim Hersteller zu reagieren, sind grundlegende Anforderungen an jeden Lieferanten. Und natürlich steigen auch die Anforderungen der Kunden, die immer weniger dazu bereit sind, Ware „von der Stange“ zu akzeptieren.

Diese Entwicklung hin zur „individuellen Fertigung für alle“ (Mass Customization) fordert den durchgängigen Ablauf aller Geschäftsprozesse, vom Design bis zur Produktion. Auch die Logistikkette spielt eine entscheidende Rolle, in der Systeme zur automatischen Identifikation unerlässlich sind.

## 6.2.2 Geschäftsprozessmanagement im Internet der Dinge

Auto-ID- und insbesondere moderne RFID-Systeme gelten als Basistechnologien für die zukünftige globale Wirtschaft. Damit weichen die bisherigen statischen Produktions- und Lieferbeziehungen einem globalen Geflecht von dynamischen Prozessketten, die über das Internet gesteuert werden.

Insbesondere die Begriffe Internet der Dinge und RFID sind schon seit längerer Zeit untrennbar miteinander verbunden. Dabei geht es um die Erarbeitung einer globalen Lösung für die lückenlose Verfolgung von Waren auf der Basis einer eindeutigen Identifikationsnummer. Jeder Artikel wird vom Hersteller mit einem RFID-Transponder versehen und kann auf diese Weise entlang der gesamten Produktions- und Lieferkette automatisch erfasst werden. Die gewonnenen Daten spiegeln zu jedem Zeitpunkt den Zustand der gesamten Prozesskette wider und erlauben so deren umfassende Optimierung.

Aufgrund der hohen Verfügbarkeit von besonders preisgünstigen und dabei leistungsstarken Einweg-Transpondern kommen RFID-Systeme bereits in zahlreichen Bereichen zum Einsatz:

**Smart Factories:** Die automatische Identifikation mittels RFID zählt gerade in der Produktion vielfach bereits zum Stand der Technik. Bei diesen Anwendungen steht die automatische Steuerung der Produktionsabläufe anhand individueller Objektdaten im Fokus. Beispielsweise werden im Automobilbau die Fertigungsroboter gemäß der genauen Kundenbestellung spezifisch angesteuert. Die Werkteile tragen hier alle Informationen über ihre individuelle Bearbeitung und Montage, so dass z. B. Lackierarbeiten auftragsgenau ausgeführt werden können.

**Asset Management:** Auch die Verwaltung materieller Vermögensgegenstände zählt zu den ausbaufähigen Anwendungsmöglichkeiten von RFID. Hier geht es vornehmlich um die Optimierung von Lagerbeständen sowie Lagermitteln wie Containern, Umlaufbehältern oder Werkzeugen. Einerseits muss eine ausreichende Zahl solcher Assets zur Verfügung stehen. Andererseits sind Assets gebundenes Kapital ohne direkten Ertrag, so dass ein möglichst kleiner Bestand anzustreben ist. Dank RFID kann der Lebenszyklus dieser Objekte lückenlos nachvollzogen und bei Objekten, die den Zugriffsbereich eines Betriebes verlassen sollen, eine klare Aussage getroffen werden, ob dies ggf. unzulässig ist. Zudem lassen sich weitere betriebliche Prozesse mit dem RFID-gestützten Asset-Management verbinden, was die Wirtschaftlichkeit der Gesamtlösung weiter erhöhen kann.

**E-Integrität:** Ein ganz anderes Anwendungsfeld für RFID eröffnet sich bei Lebensmitteln, Medikamenten oder sensiblen technischen Komponenten. Hier ist die absolute Integrität des Produktes unverzichtbar. Es muss beispielsweise gewährleistet werden, dass das richtige und vor allem das Originalmedikament beim Patienten ankommt. Oft werden hier die Begriffe der elektronischen Integrität und des elektronischen Stammbaums von Produkten verwendet. Hierzu sind zukünftig lückenlose Nachweise über die Herkunft und Stationen der Lieferkette und möglicherweise sogar der Produktionskette zu führen. Dies ist nur durch den Einsatz von Technologien zur automatischen Identifizierung möglich. Insbesondere für den Echtheitsnachweis

ist RFID aufgrund von Speicherkapazität und physikalischer sowie elektronischer Robustheit besonders geeignet.

**Gesundheitssektor:** Ob zur Dosierung von Medikamenten oder zur Kontrolle von Allergien und Diäten – mit moderner IT- und RFID-Technologie ist eine deutlich verbesserte Pflege von Patienten möglich. Im Operationssaal der Zukunft könnte RFID außerdem beispielsweise zum Tracking von OP-Utensilien und Personal eingesetzt werden, um die Sicherheit der Patienten und deren korrekte Behandlung sicherzustellen. Auch potentielle Kosteneinsparungen sowie eine schnellere, dabei aber sichere Verwaltung von Patientendaten sprechen für eine flächendeckende Einführung von RFID im Gesundheitssektor.

Neben den oben genannten Beispieldurchsetzen werden sich RFID-Technologien in vielen weiteren Bereichen nachhaltig als Basistechnologie durchsetzen. Hierzu zählen differenziertere Lösungen in der Fertigungssteuerung und Produktionslogistik, dem Tracking und Tracing von Objekten, speziell von Gepäck und Waren in der Flughafenlogistik, die Optimierung von Supply Chain Networks, die erweiterte Fahrzeuglogistik, oder auch die Postautomatisierung.

### **6.2.3 Mobile Anwendungen mittels NFC**

Die Mobilität, Fülle und Qualität von Daten, die durch Auto-ID- und RFID-Systeme erreicht wird, bildet die Basis für eine neue Entwicklungsstufe komplexer Systeme, die zunehmend unser Alltagsleben bestimmen. Durchlaufen in den meisten industriellen Anwendungen derzeit noch mobile Transponder ein fest installiertes Netzwerk von RFID-Verarbeitungsgeräten, so wird mit Near Field Communication (NFC) [6] eine Technologie bereitgestellt, die den Einsatz von mobilen Verarbeitungsgeräten flexibel und leicht möglich macht.

NFC beschreibt einen Standard für drahtlose Schnittstellen zwischen mobilen Endgeräten, wie z. B. Mobiltelefonen, die die Datenübertragung mittels Hochfrequenztechnologien und Übertragungsprotokollen analog zu RFID-Systemen bereitstellt. Die für die NFC-Datenübertragung typische geringe Reichweite von wenigen Zentimetern bedingt, dass sich der jeweilige Kommunikationspartner im Nahfeld des Senders befindet, woraus die Bezeichnung „Near Field Communication“ abgeleitet ist. Häufig werden NFC-basierte Anwendungen auch mit dem Paradigma der touch-basierten Interaktion verbunden. Dies bedeutet, dass die Datenübertragung mittels beinaher oder tatsächlicher Berührung durch ein NFC-Endgerät gesteuert wird.

NFC ist kompatibel mit bestehenden RFID-Standards und unterstützt verschiedene Betriebsmodi, die sowohl das Lesen als auch die Simulation eines Transponders ermöglichen. Damit kann der Anwendungsbereich von NFC bzw. RFID entscheidend erweitert werden. Im sogenannten Reader Emulation Modus tritt das NFC-Endgerät als vollwertiges RFID-Lesegerät auf. Dadurch lassen sich bestehende RFID-Systeme leicht und effizient durch mobile Endgeräte, wie z. B. NFC-fähige Mobiltelefone, ausbauen. Im Card Emulation Modus erlaubt NFC die Emu-

lation von passiven Standard-RFID- bzw. NFC-Transpondern. Somit können NFC-basierte Endgeräte als reguläre passive Transponder in bestehende Systeme eingebracht werden. Wird ein NFC-Gerät in diesem Modus in die Nähe eines normalen RFID-Lesegeräts gebracht, so verhält es sich aus der Sicht des Lesegeräts wie ein einfacher Transponder. Dieser NFC-Modus wird derzeit häufig für mobile Anwendungen im Zahlungsverkehr angewendet. Insbesondere hierfür ist NFC ebenfalls kompatibel mit Mifare<sup>1</sup>, einer weltweit bereits weit verbreiteten kontaktlosen Chipkartentechnologie, sowie mit Felica<sup>2</sup>, einem kontaktlosen Chipkartsystem mit großer Bedeutung für Bezahlsysteme im asiatischen Massenmarkt.

Seit dem Jahr 2005 gingen weltweit NFC-Anwendungen in Betrieb. Am häufigsten sind solche Anwendungen bis heute im öffentlichen Nahverkehr und eben im Zahlungsverkehr vertreten. Und bereits jetzt ist absehbar, dass sich die Mobiltelefone der Zukunft als persönliche NFC- bzw. RFID-Geräte etablieren werden. Mobiltelefonnutzer werden stark davon profitieren, dass künftig eine Vielzahl von alltäglichen Dienstleistungen mit Hilfe von NFC-fähigen Mobiltelefonen erledigt werden kann. Ein großer Vorteil von NFC ergibt sich auch durch die Erschließung einer völlig neuen Klasse von RFID-Anwendungen: Hier können die typischen Rollen von ortsgebundenem RFID-Lesegerät und mobilem Transponder in NFC-basierten Lösungen einfach ausgetauscht werden. Beispiele hierfür sind Smart Poster und Ticketing-Anwendungen. Hier werden die Bezahlfunktionen sicher im Mobiltelefon abgewickelt und die einfachen, passiven NFC-Transponder in der Umgebung dienen nur noch der Standortbestimmung. Generell lässt sich sagen, dass viele Anwendungen, bei denen Installation und Vernetzung von herkömmlichen RFID-Lesegeräten zu schwierig oder zu teuer erschien, z. B. in Fahrzeugen oder für eine landesweite Abdeckung, durch NFC-Technologie und NFC-fähige mobile Endgeräte möglich werden.

### **6.3 Neue Forschungsfelder und Herausforderungen des mobilen Geschäftsprozessmanagements**

Neben den etablierten Kernanwendungen von RFID- und NFC-Technologie im Bereich Bezahlsystems ergeben sich weiterführende Fragestellungen und Forschungsaufgaben für das Geschäftsprozessmanagement im Internet der Dinge. Im Folgenden skizzieren wir die entsprechenden Forschungsfelder und belegen diese jeweils mit praktischen Vorarbeiten.

Verfolgt wird der von vielen Experten als wegweisend angesehenen Forschungstrend, die nächste Generation des Geschäftsprozessmanagements durch eine Verknüpfung des Internet der Dinge, des Internet der Dienste, des Semantic Web sowie mobiler Komponenten zu realisieren. Besonderer Wert wird hierbei auf erweiterte

---

<sup>1</sup> <http://www.mifare.net>

<sup>2</sup> <http://www.sony.net/Products/felica>

Lösungen durch mobile Endgeräte gelegt, die intuitive, kosteneffiziente und leicht zugängliche Systemlösungen versprechen.

### ***6.3.1 Service-Infrastrukturen und Ökosysteme für ubiquitäre Dienste***

Zukünftige Systeme für das Geschäftsprozessmanagement sind durch eine flexible Integration des Internet der Dinge, des Internet der Dienste sowie dynamischer Geschäftsnetze zu unterstützen. Hierzu sind insbesondere hochflexible RFID-Systeme und darunter liegende Dienst-Infrastrukturen notwendig.

RFID-Systeme sind bereits heute in geschlossene und offene Systeme unterteilt. Bei geschlossenen Systemen werden Objekte nur innerhalb einer Institution erfasst und verbleiben auch dort bzw. kehren immer wieder in diese zurück. Ein offenes System hingegen erlaubt die Verwendung durch viele Benutzer für ihre eigenen Dienste und Anwendungen. Der Objektfluss ist hier in der Regel kein Kreislauf. So können z. B. Transponder auf einer Autokarosserie beim Hersteller zur Produktionssteuerung, beim Händler zur Inventur, in der Werkstatt zum Service und bei einem gewerblichen Kunden zur Fuhrparkverwaltung verwendet werden.

Gerade offene RFID-Systeme stehen noch am Anfang ihrer technischen Entwicklung und können noch wesentlich von Forschungsbeiträgen profitieren. Komplexe Fragestellungen ergeben sich insbesondere aus der Forderung nach offenen Schnittstellen und wiederverwendbaren Komponenten – und konnten bisher nur zum Teil beantwortet werden. So wird beispielweise durch das Projekt PERCI (Pervasive Service Interaktion) [7] eine Infrastruktur zur flexiblen Einbettung semantisch annotierter Web Services in NFC-basierten ubiquitären Diensten bereitgestellt [8]. PERCI-Anwendungen lassen sich dann, z. B. in Form von durch RFID-Transponder angereicherten gedruckten Postern, direkt in verschiedene Benutzerumgebungen integrieren und sind dabei besonders benutzerfreundlich und leicht zugänglich.

Durch das PERCI-Framework ist es somit möglich, NFC-basierte ubiquitäre und mobile Endbenutzeranwendungen ohne großen Aufwand auf der Basis bestehender Ökosysteme von Internet-Diensten zu entwickeln. Abbildung 6.1 zeigt als Beispiel einen Nutzer, der mit einem PERCI-Poster interagiert. Das Poster zeigt eine Übersicht eines öffentlichen Verkehrsnetzes einer Stadt und bietet eine Reihe von Diensten an, die mit einem Mobiltelefon aktiviert werden können. Der Benutzer kann direkt mit dem Telefon den gewünschten Zielort auf dem Poster anwählen. Das Telefon reagiert auf den hinter dem gedruckten Poster angebrachten NFC-Chip, liest die dort gespeicherten Informationen und verbindet sich automatisch mit dem darin definierten Dienst. So ist es unter anderem leicht möglich, aktuelle Informationen über eine Verbindung anzusehen oder direkt Parameter (wie Zeit der Abfahrt und Anzahl der Fahrgäste) anzuwählen und mit dem entsprechenden Dienst Fahrkarten für eine Strecke zu reservieren oder zu kaufen.



**Abb. 6.1** Beispiel eines PERCI-Posters basierend auf NFC Tags, die hinter dem Poster angebracht sind. Das Telefon verbindet sich automatisch mit den relevanten Diensten

Die Erstellung unterschiedlichster Anwendungen wird durch die generische Infrastruktur und die flexiblen Dienstbeschreibungen in der PERCI-Architektur ermöglicht und im Gegensatz zu ihrer unabhängigen Entwicklung stark vereinfacht. Werden durch PERCI derzeit hauptsächlich einzelne Dienste an mobile NFC-Schnittstellen gebunden, so wäre in der Zukunft die durchgängige Unterstützung von verknüpften Diensten und den hiermit verbundenen Geschäftsprozessen wünschenswert [9].

### 6.3.2 Kontext-bezogene Apps und Smart Labels

Eine erste Industrienorm für ein weltweites Netzwerk einheitlicher RFID-Bezeichner wird von der Standardisierungsinitiative EPCglobal Network mit dem Electronic Product Code (EPC) bereitgestellt. Ein hohes Wachstums- und Entwicklungspotential für das Geschäftsprozessmanagement der Zukunft besitzt das Internet der Dinge auch in erweiterten semantischen und kontextsensitiven Verfahren zum Labelling und zur Handhabung von markierten Objekten.

Durch die Verknüpfung von semantischen und kontextsensitiven Methoden mit dem Internet der Dinge sollen Warenströme künftig höchst flexibel und mit lokaler Intelligenz sowie unter Berücksichtigung ihres jeweiligen Kontexts vom Ausgangs- zum Zielpunkt geschleust werden. Damit sind RFID-Systeme nicht mehr auf eine zentrale Steuerung oder eine vollständig vernetzte Informationsinfrastruktur angewiesen. Denn ubiquitäre semantische RFID-Objekte können ihre Information lokal speichern und selbstständig verwalten. Prozessketten und Warenströme können sich in solchen Systemen autonom und flexibel an variablen Kontext und Situationen anpassen.

Zukünftig sollen intelligente Labelling-Verfahren, wie das digitale Produktgedächtnis, um detaillierte semantische Beziehungen zwischen Objekten und ihrer Umgebung ergänzt werden. Intelligente kontextverarbeitende Methoden für mobile Systeme auf der Basis von Semantic Web Technologien wurden beispielsweise im Projekt IYOUIT entwickelt [10]. Umgebungsinformationen werden hier von Senso-



**Abb. 6.2** Der NFC-basierte Dienst „MyState“: vorgefertigte oder selbst zu beschriftende Smart Labels werden per Telefon mit einer virtuellen Nachricht versehen (*links*) und an Objekte wie zum Beispiel eine Tür (*Mitte*) angeheftet. Das Auslesen solcher Labels sendet die gespeicherte Nachricht automatisch und komfortabel an Facebook oder andere soziale Netze (*rechts*)

ren und mobilen Endgeräten erfasst, um situationsnahe Daten aus Internet-Diensten ergänzt, hochgradig verdichtet, aggregiert und schließlich auf einer abstrahierten semantischen Ebene dem Benutzer oder anderen Anwendungen zur Verfügung gestellt. Kontext-Ontologien erlauben hier z. B. die effiziente Klassifikation von gesammelten Daten in differenzierte Situationen, die ihrerseits die Grundlage für weiteres wissensbasiertes Schlussfolgern bilden [11].

Ein weiterer Vorteil solcher Technologie ist der hohe Anteil an potentieller Personalisierbarkeit. Wie in Abb. 6.2 illustriert [12], ist es z. B. möglich, dass der Nutzer selber Objekte mit Smart Label in Form von sichtbaren oder unsichtbaren Markierungen versieht um physikalische Objekte mit Informationen und Diensten anzureichern. So kann beispielsweise das Verbreiten wiederkehrender Ereignisse über soziale Netzwerke vereinfacht werden: Ein Nutzer markiert einmal die Sporttasche, das Fahrrad, die Bürotür oder den Schreibtisch mit einem Smart Label (hier in Form eines NFC-basierten Aufklebers) und versieht diesen per Mobiltelefon mit einer virtuellen Nachricht. Ein schnelles Berühren des Labels mit dem Telefon reicht dann bereits aus, um die gewünschte Nachricht an die gewählten Netzwerke wie Facebook, Twitter, etc. weiterzuleiten. Zur potentiell schnellen Verbreitung solcher Tags auch in öffentlich zugänglichen Bereichen trägt die Tatsache bei, dass auch andere Leute all diese Tags verwenden können um einfach Nachrichten auf ihre eigenen sozialen Netzwerkseiten zu verbreiten. Natürlich müssen diese dafür die gespeicherten Nachrichten vorher lesen und eventuell ihren eigenen Bedürfnissen anpassen können.

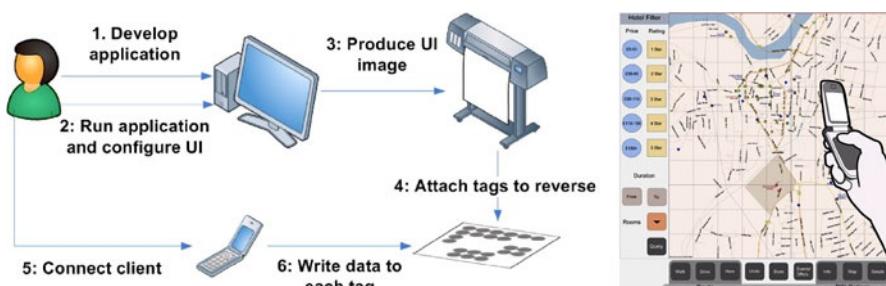
### 6.3.3 Benutzerinteraktion und einfache Benutzbarkeit

Für seinen zukünftigen Erfolg ist ein einfacher Zugriff auf das Internet der Dinge durch verschiedene Arten von Benutzern von entscheidender Bedeutung. So muss beispielsweise sichergestellt werden, dass Wartungsarbeiter industrielle RFID-basierte Prozesse leicht überwachen und steuern können. Gleichermaßen können Endbenutzer und Verbraucher von einem einfachen und intuitiven Zugang zum Internet der Dinge profitieren. Ein solcher Fall liegt z. B. dann vor, wenn sie Produkte dann

informiert verwenden können, wenn sie vorher einen direkten Einblick in den elektronischen Stammbaum eines solchen Produktes erhalten haben.

Mit Rücksicht auf die oben aufgezeigten Trends hin zu offenen, lose gekoppelten und großflächig einzusetzenden Systemen versprechen NFC-fähige mobile Endgeräte, wie z. B. Mobiletelefone, einen besonders leichten Zugriff für Jedermann und an jedem Ort. Über den einfachen Zugriff auf einzelne Objekte hinaus ist es sogar möglich, RFID- oder NFC-Transponder systematisch als erweiterte Benutzerschnittstellen für viele neuartige Dienste zu verwenden.

Das Projekt MULTITAG untersucht genau diesen Aspekt [13]. Hierbei werden NFC-basierte Endbenutzeranwendungen in verschiedene Kategorien aufgeteilt: In die Kategorie „Touch & Go“ fallen so zum Beispiel Anwendungen mit einfachen Interaktionsmustern wie Zugriffskontrollsysteme, Reportingsysteme in der Logistik sowie einfache Ticketing-Systeme. Das NFC-Gerät verhält sich hierbei wie eine kontaktlose Chipkarte, auf der ein Zugriffscode oder Ticket gespeichert ist, und wird nur einmalig und kurz am Transponder vorbeigeführt. In der Kategorie „Touch & Confirm“ finden sich Anwendungen wie Zahlungslösungen, bei denen der Anwender die Interaktion durch einen Tastendruck oder die Eingabe einer PIN in das NFC-Gerät akzeptieren muss. In die Kategorie „Touch & Interact“ fallen schließlich Anwendungen zur Unterstützung mehrstufiger Interaktionen, wie sie beispielsweise bei Verwendung eines RFID-basierten Posters mit verschiedensten Auswahloptionen für einen komfortablen Ticketing-Dienst nötig sein können. Alle diese Anwendungen können mit dem MULTITAG-Software-Framework effizient entwickelt und angeboten werden. Die Grundidee ist hierbei die einmalige Entwicklung einer Kernanwendung mit den anschließenden automatischen Transformationen in verschiedene RFID- bzw. NFC-fähige Medien. Hierbei wird insbesondere sichergestellt, dass Anwendungsverhalten und Benutzbarkeit (Look-and-Feel) über verschiedene Medien und Auflösungsstufen hinweg konsistent gehalten werden. Das MULTITAG-Framework basiert dafür auf einer Sammlung sogenannter Widgets, die eine Einheit von Daten, Darstellung und Interaktionsmöglichkeiten bilden. Durch diese Konzentration auf ein festgelegtes aber flexibles Konzept kann eine Anwendung auf unterschiedlichen Medien dargestellt werden, ohne eine Angleichung des Programmco-



**Abb. 6.3** Die Entwicklung und Gestaltung einer MULTITAG-Anwendung kann bequem auf einem PC stattfinden. Soll die finale Benutzerschnittstelle zum Beispiel ein gedrucktes Poster sein (*rechts*), wird das Interface ausgedruckt und mit Tags versehen (*links*)

des notwendig zu machen. Die Anwendungsbeispiele beinhalten Informationssysteme mit RFID-basierten Schnittstellen mittels Bildschirmen, Postern und Flyern in unterschiedlichen Auflösungen. Abbildung 6.3 illustriert den Prozess der Entwicklung solcher Anwendungen und ein Beispiel eines Benutzerinterfaces für eine Landkartenanwendung.

## 6.4 Verwandte Arbeiten

Neben den bereits zitierten verwandten Arbeiten ist speziell für den Bereich Service-Infrastrukturen innerhalb Deutschlands der BMBF-Cluster Allianz digitaler Warenfluss<sup>3</sup> (ADIWA), der von den beiden größten deutschen Herstellern von Unternehmenssoftware SAP AG und Software AG unterstützt wird, besonders hervorzuheben. Durch ADIWA soll erstmals auf der Ebene der Realwirtschaft ein offenes, hochauflösendes und äußerst flexibles Geschäftsprozessmanagement mit Hilfe von RFID und Funksensoren realisiert werden. NFC-Technologie, mobile Benutzerschnittstellen und insbesondere mobile Endbenutzeranwendungen spielen hierbei bislang allerdings noch eine untergeordnete Rolle.

Im Bereich kontextbezogener Dienste und Objekte ist das BMBF-Projekt Sem-ProM<sup>4</sup> mit einem besonderen Fokus auf das intelligente Warenmanagement mittels Smart Labels ausgerichtet. Es geht um die Idee, Waren und Produkte mit einem digitalen Gedächtnis auszustatten und so intelligente logistische Prozesse zu unterstützen. Mobile, eingebettete und RFID-basierte Elemente ermöglichen die Kommunikation zwischen Warenobjekten – und damit ein digitales Produktgedächtnis. Die Fähigkeiten solcher intelligenter Waren sollen dabei weit über die reine Identifikationsfunktion von heutigen RFID-Transpondern hinausgehen. Sie sollen beispielsweise die Daten verschiedener eingebetteter Sensoren (Temperatur, Helligkeit, Lage, etc.) auswerten und daneben alle relevanten Produkt- und Betriebsdaten erfassen. Im Sinne des Internet der Dinge steigert sich somit das Potenzial für integrierte Geschäftsprozesse durch die direkte Verfügbarkeit von Produktinformationen sowie dem einfachen und aktiven Austausch mit der Produktumgebung, anderen Warenobjekten sowie den Benutzern.

Die Idee für interaktive Dienste im Internet der Dinge wurde bereits in [14] skizziert. Hier finden sich erste Beispiele zur Verknüpfung alltäglicher Gegenstände (z. B. Bücher, Unterlagen und Visitenkarten) mit digitalen Ressourcen (z. B. elektronische Dokumente, URLs und E-Mails) durch RFID-Tags. Mit der Zielsetzung der einfachen Benutzbarkeit wurden seitdem verschiedene Technologien für interaktive ubiquitäre Dienste erprobt. Neben der bereits erläuterten Verwendung von RFID und NFC zur Kennzeichnung von – und besonders auch zur Interaktion mit – alltäglichen Objekten in der realen Welt sind außerdem hervorzuheben: frühe Arbeiten auf der Basis von Bluetooth und Infrarot [15], die Interaktion mit physikalischen

<sup>3</sup> <http://www.adiwa.net>

<sup>4</sup> <http://www.sempron.org>

Objekten mittels Laserpointern [16] sowie die Verwendung von visuellen Markern zur erweiterten Benutzerinteraktion [17]. Letztere wird darüberhinaus zunehmend durch moderne Verfahren der Bilderkennung ohne die explizite Verwendung von visuellen Markern abgelöst [18].

## 6.5 Zusammenfassung und Ausblick

Im Fokus dieses Kapitels steht der nachhaltige Trend zur Implementierung und zum Management von Geschäftsprozessen mit Hilfe des Internet der Dinge. Mobile Dienste und Apps sind hierbei von zunehmender Bedeutung, insbesondere zur selektiven Erweiterung existierender industriell etablierter sowie forschungsnaher RFID-basierter Systeme.

Wie oben erläutert steht hierbei mit NFC eine Technologie zur Verfügung, mit deren Hilfe bestehende Anwendungen sehr einfach um flexible und mobile RFID-Schnittstellen erweitert werden können. Wie bereits kurz erwähnt sind die typischen Rollen von ortsgebundenen RFID-Lesegeräten und mobilen Transpondern in NFC-basierten Anwendungen austauschbar, um beispielsweise kostengünstige großflächige Abdeckungen zu erreichen.

Offensichtlich ergeben sich durch die flexible Bereitstellung der entsprechenden Lösung allerdings auch vielfältige Fragestellungen hinsichtlich ihrer Benutzbarkeit sowie der effizienten und effektiven Systemintegration. Dies ist gerade dann der Fall, wenn herkömmliche RFID- und mobile NFC-Komponenten zusammen in einem einzigen integrierten System bereitgestellt werden sollen. Weiterführende Arbeiten könnten sich deshalb genauer mit grundlegenden Systemkonzepten zur flexiblen Integration von NFC-Komponenten in bestehende Entwicklungen beschäftigen.

In diesem Kapitel haben wir besonderen Wert auf die Differenzierung zukünftiger Arbeiten gegenüber bestehenden mobilen Lösungen in drei Bereichen gelegt:

- Innovative Infrastrukturen für RFID-basierte ubiquitäre Dienste
- Kontextbezogene Apps und Smart Labels
- Spezielle Aspekte mobiler Endanwendungen wie mobile Benutzerinteraktion und einfache Benutzbarkeit

Wir erwarten, dass sich diese Bereiche neben den Kernanwendungen in Fertigungs- und Bezahlssystemen als wichtige Felder mobiler Anwendungen von RFID- und NFC-Technologie etablieren werden. Hierbei müssen sich zukünftigen Arbeiten, über die genannten verwandten Arbeiten und Forschungsprojekte hinaus, zudem an globalen Technologietrends in dem Bereich der RFID-Chip-Produktion und Labelling-Technologie orientieren. Zusätzlich zu den bereits aufgeführten Entwicklungen zählen hierzu die zunehmende und extrem hohe Integration von RFID-Chips auf Miniaturgröße, die Entwicklung druckbarer Transponder mit wesentlich reduzierten Produktionskosten sowie die Entwicklung neuartiger RFID-Transponder mit Zusatzfunktionen wie integrierter Sensorik oder visuell lesbaren digitalen Displays.

## Literaturverzeichnis

1. Hansen W-R, Gillert F (2008) RFID for the Optimization of Business Processes. Wiley
2. Bartneck N, Klaas V, Schönherr H (2008) Prozesse optimieren mit RFID und Auto-ID: Grundlagen, Problemlösungen und Anwendungsbeispiele. Publicis
3. Seideman T. Barcodes Sweep the World. Wonders of Modern Technology. <http://www.barcoding.com> (26. Oktober 2011)
4. Paret D (2005) RFID and Contactless Smartcard Applications. Wiley
5. Finkenzeller K (2008) RFID Handbuch. Hanser
6. NFC Forum. Standardization of Near Field Communication Technology. <http://www.nfc-forum.org> (26. Oktober 2011)
7. Broll G, Rukzio E, Paolucci M, Wagner M, Schmidt A, Hussmann H (2009) PERCI: Pervasive Service Interaction with the Internet of Things. IEEE Internet Computing
8. Paolucci M, Broll G, Hamard J, Rukzio E, Wagner M, Schmidt A (2008) Bringing Semantic Services to the Internet of Things. Int J Sem Web Inform Sys (IJSWIS) 4(1), Special Issue on Mobile Services and Ontologies
9. Bertoli P, Kazhamiakin R, Paolucci M, Pistore M, Raik H, Wagner M (2009) Continuous Orchestration of Web Services via Planning. In: Proc of the 19th Int Conf on Automated Planning and Scheduling (ICAPS'09), Thessaloniki, Greece
10. Böhm S, Koolwaaij J, Luther M, Souville B, Wagner M, Wibbels M (2008) Introducing IY-OUIT. In: Proc of the 7th Int Semantic Web Conf (ISWC'08), Karlsruhe, Germany
11. Luther M, Fukazawa Y, Wagner M, Kurakake S (2008) Classification-based situational reasoning for task-oriented mobile service recommendation. Knowledge Engineering Review (KER) 23(1), Special Issue on Context & Ontologies, Cambridge Journals
12. Hardy R, Rukzio E, Holleis P, Wagner M (2011) MyState: Sharing Social and Contextual Information through Touch Interactions with Tagged Objects. In: Proc of the 13th Int Conf on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'11), Stockholm, Sweden
13. Hardy R, Rukzio E, Holleis P, Wagner M (2010) Mobile Interaction with Static and Dynamic NFC-based Displays. In: Proc of the 12th Int Conf on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'10), Lisbon, Portugal
14. Want R et al (1999) Bridging Physical and Virtual Worlds with Electronic Tags. In: Proc SIGCHI Conf on Human Factors in Computing Systems (CHI'99), ACM Press, Pittsburgh, USA, pp 370–377
15. Kindberg T et al (2002) People, Places, Things: Web Presence for the Real World. Mobile Network Applications 7(5):365–376
16. Rukzio E et al (2006) An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing, and Scanning. In: Proc 8th Int'l Conf Ubiquitous Computing (Ubi-comp 06), LNCS, Springer-Verlag, pp 87–104
17. Rohs M (2004) Real-World Interaction with Camera Phones. In: Proc of the 2nd Int. Symposium on Ubiquitous Computing Systems (UCS'04), Tokyo, Japan, pp 74–89
18. Quack T, Bay H, Van Gool LJ (2008) Object Recognition for the Internet of Things. In: Proc Internet of Things Conf (IOT 08), pp 230–246

---

# Kapitel 7

## Eine neue Generation von Geschäftsanwendungen

Michael Kern

**Zusammenfassung** Der Markt für mobile Endgeräte boomt und die Nachfrage nach mobilen Geschäftsanwendungen wächst rasant. Mittlerweile stehen den Unternehmen eine wachsende Anzahl an Business Applikationen (Apps) zur Verfügung. Deinen Erwartungen an die neue Technologie sind entsprechend hoch: Sie wollen ihre Prozesse optimieren, die Zusammenarbeit zwischen Kunden, Partnern und Mitarbeitern verbessern, flexibler und agiler werden sowie kürzere Reaktionszeiten erzielen – und das zu Recht, denn die Anwendungen werden diesem Anspruch gerecht.

Die Unternehmen der Zukunft sind mobil. Diese Vision wird durch mobile Endgeräte wie Tablets zunehmend greifbar und in vielen Unternehmen steht dieses Thema sogar auf der Vorstandsgesellschaft. Mit dem iPad schafft es zum ersten Mal ein „Computer“ diejenigen zu begeistern, für die die Informationstechnologie bisher allenfalls Mittel zum Zweck war und die ihre E-Mails eher von ihrem Sekretariat ausdrucken ließen, anstatt sie auf dem Bildschirm ihres PCs zu lesen. Das liegt daran, dass Tablets und Smartphones viele Vorteile in sich vereinen: Sie sind immer und überall einsatzbereit, vor allem leicht und intuitiv zu bedienen, bieten rund um die Uhr Internetzugang und ermöglichen Video-Telefonie, beispielsweise mit den Kollegen in China oder dem Kunden in den USA. Die Buchung eines Videokonferenzraums wird überflüssig und die Kommunikation effektiver.

Mobile Endgeräte wie Smartphones und Tablets verändern die Geschäftswelt nachhaltig. Durch mobile Geschäftsanwendungen erhalten Führungskräfte und Mitarbeiter jetzt auch den Zugang zu Unternehmensinformationen und -prozessen. Schon heute erledigen sie immer öfter wichtige Aufgaben abseits ihres Arbeitsplatzes und treffen ihre Entscheidungen von unterwegs. Die neuen Technologien werden zudem die Trennung zwischen Privat- und Geschäftsleben noch weiter auflösen.

---

Michael Kern  
sovanta AG, X-House, Heidelberg,  
E-mail: michael.kern@sovanta.com

## 7.1 Wegbereiter einer neuen Generation von Geschäftsanwendungen

Unternehmen sehen in mobilen Endgeräten wie Tablets zunehmend eine echte Alternative zum herkömmlichen PC oder Laptop und beschäftigen sich infolgedessen intensiv damit, diese in ihre IT-Infrastruktur einzubinden und erste Anwendungen zu realisieren. Vor diesem Hintergrund wächst die Zahl verfügbarer mobiler Geschäftsanwendungen stetig. Beim Großteil der verfügbaren Anwendungen handelt es sich bis dato allerdings noch um Nachschlagewerke sowie Office-, Produktivitäts- oder Kollaborationswerkzeuge. Auch das Angebot an Analyse-Tools, die Informationen und Kennzahlen aus betriebswirtschaftlichen Systemen bereitstellen, ist bereits relativ groß. Mobile Anwendungen, die umfangreiche Geschäftsabläufe unterstützen, sind hingegen noch nicht in diesem Umfang verbreitet. Genau hier verbirgt sich ein großes Potenzial, lag doch gerade bei betriebswirtschaftlichen Anwendungen in der Vergangenheit der Fokus darauf, komplexe Geschäftsprozesse in möglichst vielen Varianten abbilden zu können. Die Folge davon: Die Benutzeroberflächen der Geschäftsanwendungen waren eher durch die dahinterliegende Datenbankstruktur geprägt als durch die Anforderungen der Anwender. Im Rückblick gilt dies sowohl für hostbasierte Lösungen in Zeiten der Großrechner als auch für die Client-Server-basierten Anwendungen in den 1990er Jahren. Die browserbasierten Anwendungen im Zuge des Internet-Booms um die Jahrtausendwende haben einen ersten Schritt dahingehend gemacht, Geschäftsanwendungen zu vereinfachen und die Belange der Anwender stärker in den Vordergrund zu stellen. Die neue Gerätegeneration der Tablets, allen voran das iPad von Apple, haben diese Situation geradezu auf den Kopf gestellt. Die neuen Anwendungen zeichnen sich durch eine konsequente Ausrichtung an den Bedürfnissen der Anwender aus und stellen ganz neue Anforderungen an die grafische Aufbereitung von Informationen und Interaktionen. Anwendungen, die diese Ansprüche nicht erfüllen, werden von den Usern nicht akzeptiert und können sich nicht durchsetzen. Dies gilt insbesondere dann, wenn es sich nicht um den sogenannten Power-User handelt, der den ganzen Tag Massendaten erfasst, sondern um Gelegenheitsuser wie Fach- und Führungskräfte. Der Einzug von Tablets und Smartphones in die Unternehmen überträgt diese Prinzipien nun auch auf die Geschäftsanwendungen. Was mit webbasierten Anwendungen vor etwa zehn Jahren begonnen hat, wird nun durch die neuen Technologien und Anwendungen vollendet. Mobile Business Apps repräsentieren eine neue Generation von Geschäftsanwendungen. Diese sind anwenderorientiert, konzentrieren sich auf das Wesentliche, sind einfach zu bedienen, grafisch anspruchsvoll aufbereitet und es macht Spaß mit ihnen zu arbeiten. Und genau hier liegt das große Potenzial dieser Anwendungen: Weil es den Anwendern Spaß macht mit diesen zu arbeiten, werden bereits vorhandene, aber schlecht zugängliche Informationen in ERP-Systemen (Enterprise Resource Planning) für die Entscheidungsfindung besser genutzt. Prozesse laufen effizienter ab, weil die Beteiligten besser eingebunden sind und ihre Aufgaben schneller erledigen können. All das fördert die Zusammenarbeit sowohl zwischen den Abteilungen als auch zwischen Mitarbeitern und deren Führungskräften. So erhalten alle wieder den Blick für das Wesentliche – ihr eigentliches Kerngeschäft. Vor dem Hin-

tergrund, dass gerade Führungskräfte mit der Bewältigung ihrer täglichen Aufgaben und Verantwortungen stark ausgelastet und dazu noch häufig unterwegs sind, können sie von solchen „entschlackten“ Anwendungen besonders stark profitieren. Mobile Business Apps ermöglichen ihnen einen schnellen und einfachen Blick auf die Kernprozesse im Unternehmen und eröffnen zudem neue Anwendungsperspektiven. Sie erhalten eine präzise Grundlage für ihre Entscheidungen, egal ob sie gerade im Flugzeug sitzen, sich in einem Meeting befinden oder auf dem Weg zur nächsten Niederlassung sind.

Damit die mobilen Geschäftsanwendungen diesen Ansprüchen gerecht werden, müssen die Hersteller dieser Anwendungen bei deren Entwicklung neue Ansätze berücksichtigen. Denn die Geschäftsprozesse in den Unternehmen werden ja nicht einfach weniger komplex. Die entscheidende Frage lautet demnach, wie sich bei der Entwicklung mobiler Apps die Komplexität intelligent verbergen und dadurch die Nutzung der Anwendungen vereinfachen lässt. Die Antwort: Dies gelingt nur über einen konsequent designgetriebenen Entwicklungsansatz. Dabei bedeutet Design in diesem Zusammenhang weit mehr als nur Layout und hübsche Bilder. Es geht dabei vielmehr darum, die Inhalte optimal darzustellen, die Interaktion zwischen Anwender und App sowie Prozessabläufe einfach abzubilden. Dazu bedarf es interdisziplinärer Entwicklungsteams. Das richtige Zusammenspiel von Designern, Interaktionsspezialisten, Prozess- und Technologieexperten sowie den Anwendern selbst bildet eine unerlässliche Grundlage für die erfolgreiche Gestaltung mobiler Geschäftsanwendungen.

## 7.2 Business Apps erobern das Unternehmen

Das Potenzial von Business Apps liegt durch die oben beschriebenen Vorteile auf der Hand und in einigen Geschäftsbereichen ist die Nutzung mobiler Anwendungen für die Unternehmen inzwischen fast schon selbstverständlich. Der Vertriebsaußendienst ist beispielsweise ein Geschäftsbereich, in dem sich der mobile Zugriff auf die Unternehmenssoftware sehr schnell durchgesetzt hat. Dort sind mittlerweile mobile CRM-Anwendungen (Customer Relationship Management) für das Kundenbeziehungsmanagement bereits weit verbreitet und aus dem Arbeitsalltag vieler Außenstellenmitarbeiter nicht mehr wegzudenken. Sie ermöglichen einen zeitnahen Zugriff auf Kundeninformationen, Verträge oder vereinbarte Liefertermine, wodurch Anfragen schnell und unkompliziert bearbeitet werden können. Das sorgt für eine höhere Kundenbindung und im besten Fall kann dies sogar Geschäftsabschlüsse beschleunigen.

Anwendungen im Bereich Marketing und Vertrieb nehmen sicherlich eine Vorreiterrolle ein, wenn es darum geht Geschäftsabläufe zu mobilisieren. Aber auch in anderen Bereichen halten mobile Anwendungen Einzug in den Etagen der Fach- und Führungskräfte. Im Folgenden werden anhand einiger konkreter Praxisbeispiele die Anwendungsszenarien und Potenziale der mobilen Geschäftsanwendungen aufgezeigt. Dazu dient ein Beispiel für die optimierte Energiebeschaffung für Kunden

eines Versorgungsunternehmens sowie verschiedene Anwendungen, die das mobile Personalmanagement Wirklichkeit werden lassen.

### 7.3 Großes Potenzial im Personalwesen

Gerade im Personalwesen bietet sich der Einsatz von mobilen Business Apps an. Eine erfolgreiche Personalarbeit lässt sich nur dann umsetzen, wenn Führungskräfte und Personalabteilung optimal zusammenarbeiten. Um diese Art der Zusammenarbeit zu ermöglichen, müssen die HR-Anwendungen für die Führungskräfte sehr einfach zu bedienen sein und auf das Wesentliche fokussieren. Der Informations- und der Datenaustausch wie auch die Abläufe zwischen Führungskräften, Recruitern und Sachbearbeitern lassen sich mit ihrer Hilfe über Abteilungsgrenzen und Standorte hinweg einfach gestalten. Ganz gleich ob in der Beschaffung, Entwicklung oder dem Management von Personal, eröffnen sie den Anwendern völlig neue Möglichkeiten.

So lassen sich beispielsweise innerhalb von Sekunden per Knopfdruck alle Personalinformationen auf einen Tablet-PC übertragen. Dadurch kann das Management schnell ermitteln, ob und wenn ja, welche Engpässe die Niederlassungsleiter in bestimmten Regionen bei der Besetzung von offenen Stellen überwinden müssen. Geht ein Geschäftsführer oder ein Personalleiter in ein Management-Meeting, hat er so auf jeden Fall die notwendigen Zahlen, Daten und Fakten über den entsprechenden Personalbereich zur Hand. Mit Hilfe der Business App greift er einfach auf die Daten im ERP-System zu und kann so frühzeitig über geeignete Maßnahmen entscheiden. Schon während des Gesprächs ist es beispielsweise möglich, dass der Manager wichtige Aufgaben definiert und diese über seinen Tablet unverzüglich an die verantwortlichen Mitarbeiter in der Personalabteilung delegiert. Dadurch lassen sich Effizienz und Effektivität erhöhen. Bei diesem Szenario wird die Business App sozusagen als Management-Informationssystem genutzt.

Wird diese Prozesskette weiterverfolgt, könnte eine der Maßnahmen lauten, den in den bestimmten Regionen festgestellten Personalmangel mittels einer Jobinitiative auszugleichen. Bleiben wir bei diesem Beispiel und spinnen es noch ein bisschen weiter. Im Zuge der Initiative werden Bewerbungsgespräche an verschiedenen Standorten anberaumt und durchgeführt. Ein gutes Szenario dafür, wie die Business App eine Führungskraft auch in diesem Fall situationsbezogen unterstützen kann. Der Personalmanager oder Recruiter nutzt in diesem Fall einen Interview Assistant. Auf mobilen Devices kann er sowohl einen Fragebogen aufrufen als auch durch alle Dokumente – wie Anschreiben, Lebenslauf und Zeugnisse eines Bewerbers – schnell und lautlos blättern. Und nach dem Interview kann er direkt einen standardisierten Bewertungsbogen ausfüllen und die erfassten Daten der einzelnen Stellenkandidaten per Mausklick abgleichen und entsprechende Folgeprozesse einleiten. Der komplette Auswahlprozess lässt sich also mit einer leicht bedienbaren Business App unterstützen.

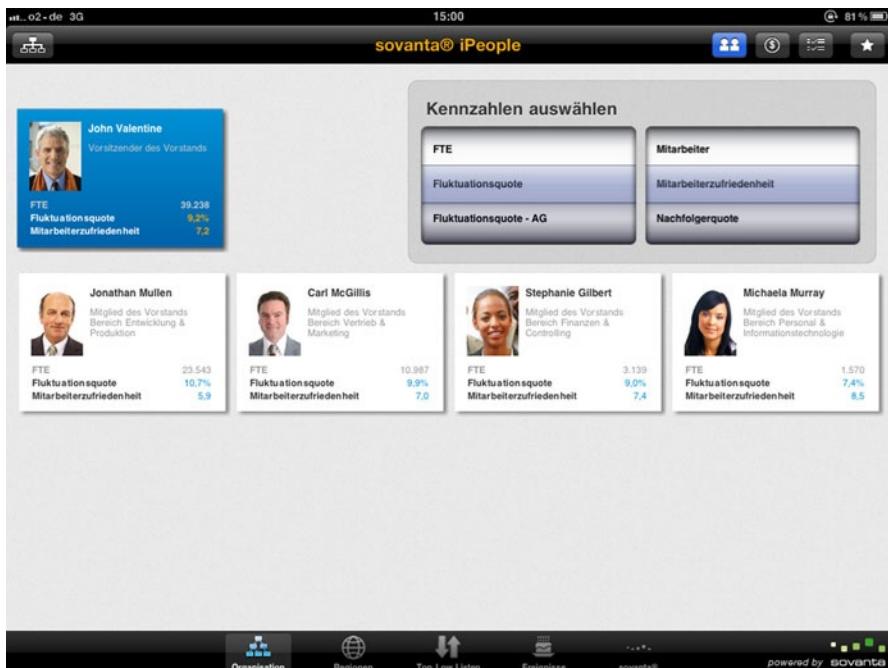


Abb. 7.1 Die Business App als Management-Informationssystem



Abb. 7.2 Beschäftigungsquoten nach Regionen

Ein weiteres Beispiel, diesmal aus der Personalentwicklung. Auch dort kann eine solche Business App die Zusammenarbeit zwischen den Führungskräften und der Personalabteilung entscheidend unterstützen – beispielsweise bei Zielvereinbarungsgesprächen und Leistungsbeurteilungen, einem der wichtigsten Personalprozesse überhaupt. Da diese Prozesse nur zwei- bis dreimal im Jahr stattfinden, kommen sie in den Unternehmen oft zu kurz. Zumal deren Rahmen oder Turnus von der Personalabteilung festgelegt wird, die Umsetzung aber in den Verantwortungsbereich des Abteilungsleiters fällt. Diese finden für die Koordination und Planung der Gespräche schlichtweg oftmals keine Zeit. Insbesondere dann, wenn die dafür zur Verfügung stehenden Werkzeuge komplex und unpraktikabel sind. Eine Business App kann auch hier leicht Abhilfe schaffen. Ihre intuitive und einfache Bedienung reduziert die Komplexität des gesamten Prozessverlaufs, steigert die Akzeptanz bei den Beteiligten und bindet die Führungskräfte richtig in den Personalentwicklungsprozess ein. So können diese beispielsweise Gespräche zur Leistungsbeurteilung bequem über ihr Tablet festhalten. Gleichzeitig wird während des Gesprächs keine Barriere zwischen Führungskraft und Mitarbeiter aufgebaut, wie sie bei der Verwendung eines Laptops zwangsläufig entsteht.

Die Führungskräfte erhalten für eine gezielte Personalentwicklung genau das, was sie benötigen: einen schnellen und unbürokratischen Überblick über Gesprächshistorie und vereinbarte Ziele sowie Leistungs- und Potenzialbeurteilungen. Sie ha-



Abb. 7.3 Mobiles Performance & Talent Management



**Abb. 7.4** Solide Entscheidungsgrundlagen durch exakte Personalinformationen

ben damit jederzeit und überall einen exakten Überblick darüber, welcher Mitarbeiter welche Ziele verfolgt oder bereits erreicht hat. Damit hat er stets eine solide Entscheidungsgrundlage zur Hand, die er für ein effizientes Performance Management seiner Mitarbeiter nutzen kann. In der Personalentwicklung lassen sich neben dem Performance Management auch die Nachfolgeplanung sowie das Talent Management durch mobile Business Apps sinnvoll unterstützen. In diesem Zusammenhang eignen sich beispielsweise Tablets und die dazugehörigen Anwendungen durch ihre vielseitigen Multimedia- und Collaboration-Funktionen besonders für das E-Learning.

Darüber hinaus gibt es noch viele weitere Szenarien, in denen der Einsatz von mobilen Business Apps das Personalmanagement unterstützen kann. In der Personalbeschaffung sind dies zum Beispiel das Employer Branding, die Suche nach Stellenangeboten, das Talent Relationship Management oder das Recruiting. Im Personalmanagement sind das die Bereiche Business Intelligence, Reporting oder digitale Personalakte. Sie ermöglichen den Führungskräften hierbei beispielsweise auch, Genehmigungsprozesse wie Urlaubsanträge, Krankmeldungen, Reisekosten oder Anträge auf Weiterbildungslehrgänge einfach und schnell elektronisch zu bearbeiten.

## 7.4 Mobile Prozesse fördern Zusammenarbeit

Die Business Apps erleichtern aber nicht nur einzelnen Personen die Arbeit, sondern verbessern durch reibungslosere Abläufe die Zusammenarbeit über Abteilungen und Standorte hinweg. Erst wenn Führungskräfte, Abteilungsleiter und Personalsachbearbeiter an einem Strang ziehen, lässt sich Personalarbeit erfolgreich gestalten.

Alle an einem Prozess Beteiligten müssen dazu die einzelnen Prozessschritte so praktikabel wie möglich durchlaufen können. Die Personalabteilung sollte deshalb einfache Prozesse definieren und den Führungskräften leicht bedienbare Anwendungen wie Business Apps zur Verfügung stellen. Denn wenn die Führungskräfte eine Anwendung nicht nutzen, beispielsweise weil sie kompliziert zu bedienen ist, bleibt der Prozess hängen. Führungskräfte müssen also in die Prozesse mit einbezogen werden, einfache Anwendungen nutzen sowie auf alle vorhandenen Personaldaten zugreifen können. Und genau dies ermöglichen Business Apps.

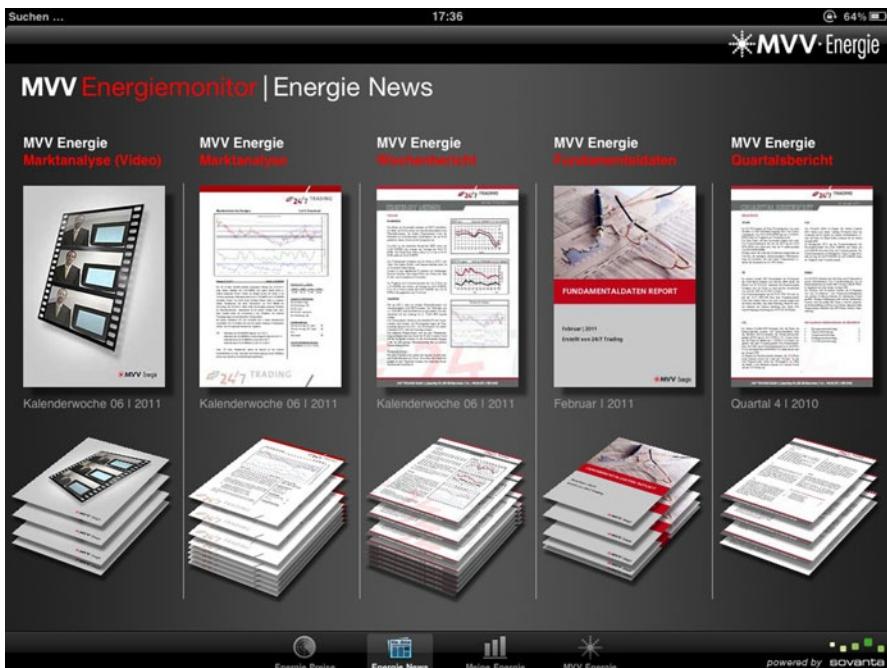
## 7.5 Nicht nur mobile Personalprozesse setzen Energie frei

Das Potenzial mobiler Geschäftsanwendungen sorgt jedoch nicht nur im Personalwesen für effizientere Prozesse und bessere Zusammenarbeit. Das zeigt ein Anwendungsbeispiel aus der Energiewirtschaft.

Hauptakteur des Referenzbeispiels ist ein Energieversorgungsunternehmen. Dieses ist mit innovativen Produkten für Geschäftskunden seit Jahren erfolgreich am Markt präsent. Einen wesentlichen Grundstein für diesen Erfolg legte das Unternehmen mit viel Weitsicht bereits vor einigen Jahren. Damals wurde ein Energiefonds ins Leben gerufen und erstmals den Kunden angeboten. Energie ist ein knappes und wertvolles Gut. Umso wichtiger war es, mit dem Energiefonds den Kunden ein einheitlich strukturiertes und transparentes Instrument zur Verfügung zu stellen, mit dem sie ihren Strom- und Gaseinkauf systematisch und nachhaltig steuern können. Mit Unterstützung einer neuen mobilen Geschäftsanwendung, dem sogenannten Energiemonitor, lassen sich Preisschwankungen optimal ausnutzen und der Einkauf erfolgt erst dann, wenn Strom- und Gaspreise auf dem Tiefstand sind.

## 7.6 Energiemonitor macht Energiebeschaffung mobil

Mit dem Energiemonitor setzt das Unternehmen nun seine Strategie, den Kunden einen optimalen Service zu bieten, konsequent fort. Auf Basis der zukunftsweisen Technologie mobiler Geschäftsanwendungen stellte das Energieversorgungsunternehmen die Weichen, um auch in Zukunft auf diesem wettbewerbsintensiven Markt weiter wachsen zu können. Mit dem im Einsatz befindlichen Monitor lässt sich die Energiebeschaffung großer Unternehmen optimieren. Der hohe Nutzwert



**Abb. 7.5** Relevante Informationen, Prognosen und Analysen übersichtlich gestaltet

des Monitors, der technisch auf dem Apple iPad basiert, resultiert daraus, dass dieser den Einkäufern auf einen Blick alle relevanten Informationen über die aktuellen Energiemarkte, Prognosen und Analysen sowie Informationen über die Beschaffungssituation des eigenen Unternehmens zur Verfügung stellt.

Die Idee des Energiemonitors beruht darauf, dass Unternehmen, die große Energiekontingente einkaufen, auf das Know-how und die Expertise von Spezialisten angewiesen sind. Als Energieunternehmen mit einer eigenen Handelsgesellschaft hat dieses die notwendige Expertise bereits im Haus. So war es nur folgerichtig, den Kunden über den Energiemonitor diese hochwertigen und umfassenden Informationen immer und überall zur Verfügung zu stellen und damit deren Einkauf optimal zu gestalten. Der Informationsvorsprung, der sich durch die mobile Geschäftsanwendung bei der Beschaffung von Energie an den volatilen Märkten ergibt, stellt einen echten Wettbewerbsvorteil und damit einen Mehrwert für die Kunden dar. Mit dem Energiemonitor können die Unternehmen ihre gesamte Energiebeschaffung optimieren und damit letztlich Kosten sparen. In Zeiten steigender Rohstoffpreise ein wichtiger Aspekt, der den dadurch erzeugten Kostendruck in den Unternehmen mindert.

Die Entscheidung, den Energiemonitor in einem ersten Schritt ab Frühjahr 2011 den großen Energiekunden des Unternehmens in Form einer mobilen Anwendung für das iPad anzubieten, fiel aufgrund der hohen Benutzerfreundlichkeit und der enormen technischen Möglichkeiten dieser Plattform.



Abb. 7.6 Übersichtliche Preisentwicklung durch Energiemonitor

## 7.7 Spielregeln einhalten

Wenn auch die Einsatzmöglichkeiten mobiler Geschäftsanwendungen beinahe grenzenlos erscheinen, sollten Unternehmen, die sich mit deren Integration in die bestehende IT-Landschaft beschäftigen, einige Grundregeln beachten. Eine Devise lautet: Erst einmal klein anfangen und Erfahrungen sammeln. Sie sollten ein Szenario auswählen und dafür ein Pilotprojekt starten. So lernen sie mobile Projekte kennen und erfahren gleichzeitig mobile Technologien. Anschließend können sie sich dann weitere Szenarien aussuchen und realisieren. Dieses schrittweise Vorgehen hat den Vorteil, dass sich mobile Geschäftsanwendungen schnell und einfach einführen lassen und sich der Nutzen sofort einstellt. Anschließend können Unternehmen dann die notwendigen Infrastrukturmaßnahmen für den flächendeckenden Roll-out vornehmen.

Eine weitere Devise lautet, das Thema Sicherheit von Anfang an mit hoher Priorität zu verfolgen. Denn gerade die mobile Verarbeitung unternehmenskritischer Daten verlangt höchste Sicherheitsstandards. Die Datensicherheit wird dabei über mehrere Mechanismen erreicht. Zum einen gilt es, die mobilen Endgeräte mit Unterstützung von MDM-Lösungen (Mobile Device Management) sicher in die IT-Infrastruktur einzubinden und auf diese Weise eine zentrale Administrierbarkeit zu gewährleisten. Dies ist absolut notwendig, um beispielsweise beim Verlust oder

Diebstahl von Geräten kritische Daten und Zugänge zentral löschen bzw. sperren zu können.

Neben dem Einsatz einer Lösung für das Mobile Device Management ist die zu wählende Architektur mobiler Geschäftsanwendungen im Zusammenspiel mit bestehenden ERP-Systemen ein weiterer, wichtiger Sicherheitsaspekt. So ist für jedes Szenario festzulegen, welche Daten tatsächlich physisch auf den mobilen Endgeräten benötigt werden bzw. welche Daten nur zur Laufzeit online aus den Back-End-Systemen bereitgestellt werden sollen. Grundsätzlich sollte dabei die Devise gelten, so viele Daten wie möglich in den ERP-Systemen zu belassen, um die vorhandenen Sicherheitsmechanismen dieser Systeme voll auszunutzen.

Authentifizierungen an den jeweiligen Systemen können über gängige Sicherheitsmechanismen wie die Anmeldung via Username und Passwort und/oder zertifikatsbasierte Lösungen realisiert werden. Darüber hinaus wird eine sichere Übertragung der Daten durch die Nutzung erprobter Protokolle wie SSL (Secure Socket Layer), TLS (Transport Layer Security) oder ähnliche sichergestellt.

## 7.8 Stationäre Systeme nicht ersetzbar

Auch wenn der Einsatz mobiler Business Apps sehr vielfältig sein kann, so hat auch dieser seine Grenzen. Mobile Anwendungen wollen und können nicht alle bestehenden stationären Anwendungen ersetzen. Für die Anwender sind Business Apps dann von Vorteil, wenn sie primär Inhalte konsumieren, präsentieren oder in Entscheidungsprozesse eingebunden sind. Alle Mitarbeiter, die fast ausschließlich am Arbeitsplatz sitzen und als sogenannte Power User umfangreiche und komplexe Inhalte verarbeiten müssen, sind nach wie vor auf einen großen Bildschirm, eine Tastatur, die Maus und leistungsfähige Software angewiesen. Sie arbeiten besser direkt an einem ERP-Back-End-System, das u. a. für die Eingabe von Massendaten konzipiert ist.

Zu erwarten ist allerdings, dass verschiedene Systeme wie PCs, Laptops, Tablets und Smartphones in ihren Bedienkonzepten und Benutzeroberflächen konvergieren. Denn die Bedienung mittels Finger und Touchscreen ist eben deutlich intuitiver als die Steuerung über Maus und Tastatur. Das zeigt sich heute bereits darin, dass die Touchpads von Laptops mittlerweile auch sogenannte „Zoomgesten“ mit zwei Fingern unterstützen.

## 7.9 Die Zukunft ist mobil

Die Arbeitswelt von morgen wird eine andere sein. Mobile Business Apps spielen eine maßgebliche Rolle in diesem Veränderungsprozess. Nach und nach werden sie die Strukturen in den Unternehmen verändern. Ihr durchschlagender Erfolg wird neben dem hohen Nutzwert vor allem durch die damit verbundenen Prozess-

innovationen erzielt. Die neue Technologie der Tablets in Verbindung mit einem designgetriebenen Entwicklungsansatz lässt eine völlig neue Generation von Geschäftsanwendungen entstehen. Dabei rückt die einfache, intuitive Bedienung für den Anwender in den Vordergrund. Gute Argumente, die den vermehrten Einsatz solcher Geschäftsanwendungen in den Unternehmen erklären. Sie verbessern Qualität, steigern Effizienz, ermöglichen mehr Flexibilität und Agilität sowie kürzere Reaktionszeiten und können damit einen wichtigen Beitrag zum Unternehmenserfolg leisten.

---

# Kapitel 8

## Herausforderungen bei der Mobilisierung von Business Applikationen und erste Lösungsansätze

Matthias Euler, Marcus Hacke, Claus Hartherz, Sascha Steiner  
und Stephan Verclas

**Zusammenfassung** Vor zehn Jahren hieß Mobilität, jedem Mitarbeiter ein Diensthandy zur Verfügung zu stellen. Diese einfache und effektive Lösung genügt jedoch angesichts der aktuellen Technologieentwicklung nicht mehr. Ein Mobiltelefon hat heute mehr Rechenleistung und Funktionalitäten als ein damaliges Notebook und die dazugehörige Breitbandabdeckung erlaubt praktisch von überall den Zugriff auf das Internet. Moderne mobile Anwendungen stehen nicht mehr nur für Telekommunikation, sondern schaffen eine direkte Verbindung zur restlichen IT-Welt. Von ihren mobilen Geräten aus, sei es vom Mobiltelefon oder vom Notebook, können Mitarbeiter heute auf Anwendungen und Daten zugreifen, die früher nur im Büro verfügbar waren. Das entscheidende Thema ist hierbei die umfassende Mobilisierung von Unternehmensprozessen und -applikationen. Vom Vertriebsmitarbeiter über den reisenden Wissensarbeiter bis zum Manager gibt es zahlreiche Anwendungsfälle über E-Mail und Kalendersynchronisation hinaus, in denen der schnelle mobile Zugriff auf Informationen oder die Einbindung in Geschäftsprozesse mit einem kurzen Griff in die Westentasche erfolgversprechend ist. Das reicht vom Genehmigungswor-

---

Matthias Euler  
Deutsche Telekom AG, Darmstadt,  
E-mail: meuler@telekom.de

Marcus Hacke  
T-Systems International GmbH, Darmstadt,  
E-mail: marcus.hacke@t-systems.com

Claus Hartherz  
T-Systems International GmbH, Bonn,  
E-mail: claus.hartherz@t-systems.com

Sascha Steiner  
T-Systems International GmbH, Darmstadt,  
E-mail: sascha.steiner@t-systems.com

Stephan Verclas  
T-Systems International GmbH, München,  
E-mail: stephan.verclas@t-systems.com

flow oder Alarmen mit entscheidungsrelevanten Zusatzinformationen über aktuellste Verkaufschancen und offene Servicetickets vor dem Kundenbesuch bis hin zur Interaktion mit dem Backoffice, um Freigaben während des Besuchs beim Kunden einzuholen. Das Smartphone wird neben dem Notebook zum universellen Begleiter und stärker als bisher verzahnt mit Kommunikationsinfrastruktur, Kollaborationstools und der Anwendungslandschaft, was damit auch die Aspekte Sicherheit, Verfügbarkeit und Managebarkeit über die gesamte Kette weiter hervorhebt. Voraussetzung dafür ist die Verfügbarkeit einer Infrastruktur, die die Anforderungen der Unternehmen insbesondere an Sicherheit erfüllt.

## 8.1 Treiber und Hemmnisse für mobile Business-Apps

Die große Verbreitung von Smartphones und Tablet-PCs und deren steigende Leistungsfähigkeit machen diese Gerätekasse zunehmend für die Nutzung als mobile Verlängerung von klassischen Unternehmensanwendungen interessant. Zudem gewährleistet die steigende Verfügbarkeit von breitbandigen, mobilen Zugangstechnologien wie UMTS, WLAN und zunehmend auch LTE einen stabilen und schnellen Zugang zum Netz und somit in die Unternehmens-IT hinein.

Bei der Einbindung von mobilen Endgeräten in die IT-Infrastruktur von Unternehmen gibt es jedoch auch vielfältige Herausforderungen. Aufgrund der Vielfalt der mobilen Betriebssystemplattformen stellen hierbei die übergreifende Managebarkeit und die sich ergebenden komplexen Integrationsszenarien die größte Herausforderung dar. Zudem muss dem erhöhten Sicherheitsrisiko mobiler Endgeräte durch Verlust oder Diebstahl Rechnung getragen werden um wertvolle Unternehmensdaten vor unbefugtem Zugriff zu schützen.

Der folgende Abschnitt soll die einzelnen Aspekte näher erläutern und die einzelnen Dimensionen des Problems herausarbeiten.

### 8.1.1 Treiber für mobile Business-Apps

#### 8.1.1.1 Boom von Smartphones und Tablets

Smartphones und Tablet-PCs sind mittlerweile massenmarktfähig geworden und in den nächsten Jahren wird ihre Verbreitung noch deutlich zunehmen. Die Anzahl der verkauften Smartphones ist von 2009 auf 2010 um 80 % gestiegen [1]. Allein in Deutschland wird für 2011 ein Absatz von 10,1 Mio Geräten prognostiziert [2].

Diese Gerätekasse hat inzwischen eine Leistungsfähigkeit erreicht, die mit der von wenige Jahre alten PCs vergleichbar ist. Damit werden sie zunehmend auch für den Einsatz als alternatives Zugangsendgerät für Unternehmensanwendungen interessant. Auch die Bedienbarkeit hat in den letzten Jahren erhebliche Fortschritte gemacht.

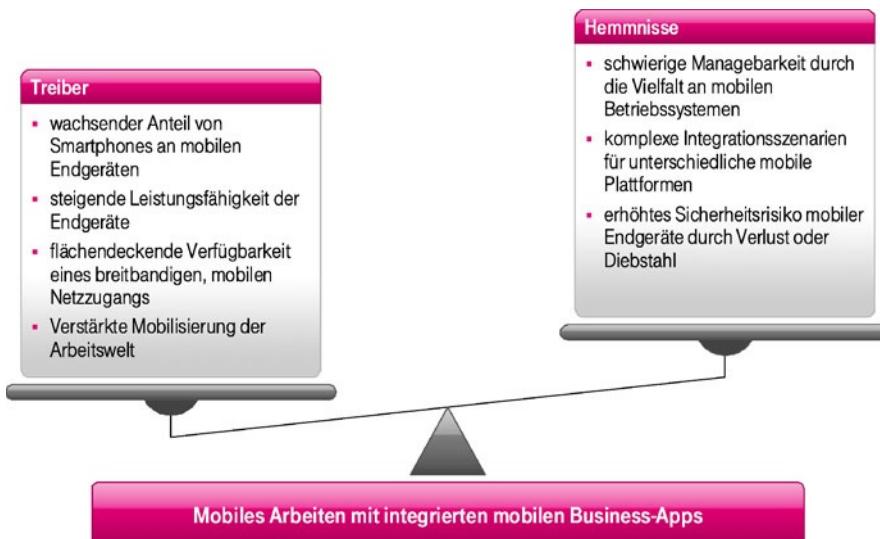


Abb. 8.1 Treiber und Hemmnisse für mobile Business-Apps

Insbesondere die Einführung des iPads hat im Markt für Tablet-PCs eine rasanten Entwicklung in Gang gesetzt. Tablets gehören mit prognostizierten 1,5 Millionen verkauften Geräten in Deutschland [3] inzwischen zu einer festen Größe im Computermarkt und etablieren sich als neue Gerätekasse zwischen Laptop und Smartphone.

### 8.1.1.2 Verfügbarkeit eines breitbandigen, mobilen Netzzugangs

Durch die Verfügbarkeit und den flächendeckenden Ausbau der breitbandigen Mobilfunknetze wird die Nutzung mobiler Apps auch für anspruchsvollere Business-Aufgaben immer interessanter.

Die mobile Netzinfrastruktur hat sich in den zurückliegenden Jahren enorm weiterentwickelt. Mit UMTS, dem Mobilfunkstandard der dritten Generation (3G), haben sich die Datenübertragungsraten auf bis zu 14,4 Megabit pro Sekunde erhöht. Die UMTS-Netze erreichen Mitte 2010 ca. 81 Prozent der Bevölkerung, in der Fläche gibt es eine Abdeckung von insgesamt etwa 70 Prozent.

Einige Netzbetreiber haben 2010 mit dem Aufbau der LTE-Mobilfunknetze begonnen, einer Weiterentwicklung der heutigen 3G-Netze. Mit dem Übertragungsstandard LTE sind Datenraten von theoretisch 300 Megabit pro Sekunde erreichbar. Neben der Datenübertragungsgeschwindigkeit bietet LTE gegenüber den Vorgänger-Generationen auch den Vorteil geringerer Latenzen, was insgesamt zu einem deutlich verbesserten Benutzererlebnis beim Surfen und anderen Applikationen führt. Beim Ausbau des LTE-Netzes wird zunächst der Schwerpunkt auf die Erschließung von bisher schlecht versorgten Gebieten gelegt. Vorrang hat hierbei

der ländliche Bereich: Die Bundesnetzagentur schreibt den Netzbetreibern einen 95 %igen Ausbau dieser Gebiete bis 2014 vor.

### 8.1.1.3 Mobilisierung der Arbeitswelt

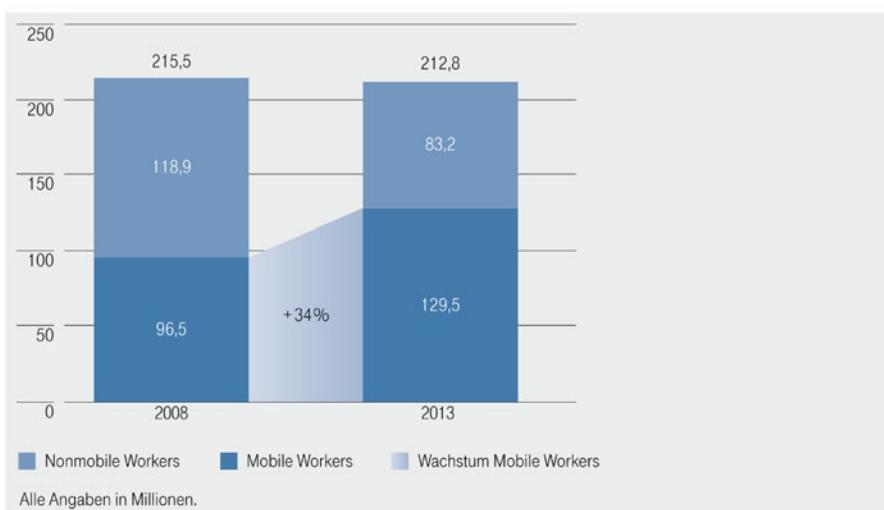
Die Globalisierung der Wirtschaft verändert die Arbeitsprozesse und damit auch die Arbeitsweise vieler Mitarbeiter. Unternehmen bilden Wertschöpfungsnetzwerke mit weltweiten Partnern und Zulieferern. Auch die Kunden verteilen sich auf alle Kontinente. Damit einher geht eine immer stärkere Mobilität der Mitarbeiter. Nach Schätzungen der Analysten von IDC arbeiten bereits heute mehr als eine Milliarde Menschen weltweit mobil [4] (vgl. Abb. 8.2). Experten zufolge werden spätestens in zwei Jahren vier von zehn Mitarbeitern ständig von unterwegs Informationen abrufen [5].

Neben den Anforderungen aus der Arbeitswelt ist auch ein Wandel der persönlichen Nutzungsgewohnheiten mobiler Endgeräte zu beobachten. Immer mehr Smartphone- und Tablet-PC-Besitzer nutzen die App-Vielfalt als kleine Helfer für verschiedenste Situationen unterwegs. Sie informieren sich über Newsticker, tauschen sich über Fachthemen in Wikis und Micro Blogs aus oder nutzen Social-Media-Kanäle wie Facebook, Xing oder LinkedIn. Der Trend zur neuen Generation der mobilen Kommunikation lässt sich auch daran ablesen, dass inzwischen bereits

---

#### Mobilitätswachstum in Westeuropa.

---



**Abb. 8.2** Mobilitätswachstum in Westeuropa. Quelle: IDC, Worldwide Mobile Worker Population 2009–2013 Forecast, Übersetzung ins Deutsche: T-Systems International GmbH

rund ein Drittel der rund 600 Millionen Facebook-Mitglieder ihren Account auch mobil nutzen (Stand 01/2011) [6].

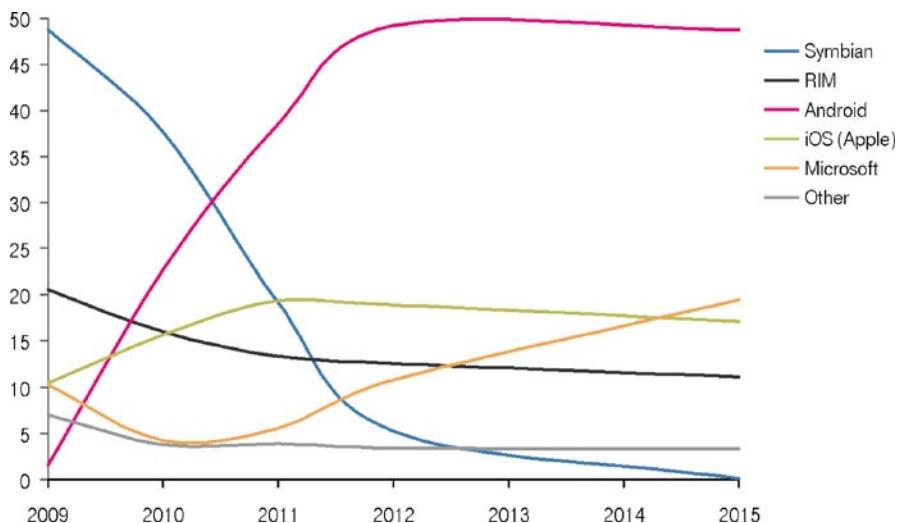
Ein weiterer Aspekt, der einen signifikanten Einfluss auf den Umgang mit mobilen Endgeräten in den Unternehmen hat, ist der starke Wettbewerb auf dem Arbeitsmarkt um die besten Köpfe. Die intensiv umworbenen „Digital Natives“ bilden die Generation, die mit digitalen Technologien aufgewachsen ist, ständig online ist und für die Smartphone oder Tablet-PC zu wertvollen Arbeitsmitteln geworden sind. Sie erheben vermehrt den Anspruch, auch an ihrem Arbeitsplatz entsprechende Geräte sowie Web 2.0-Anwendungen einsetzen zu dürfen. Sie verstehen nicht, warum sie unter Einhaltung der Sicherheitsvorschriften ihre privaten Geräte nicht für berufliche Zwecke einsetzen können. Nicht selten wird daher die Umgehung offizieller Firmenrichtlinien in Kauf genommen um private Endgeräte in das Unternehmensnetz einzubinden. Wollen Unternehmen sich weiterhin als attraktiver Arbeitgeber präsentieren, müssen sie auf diese Entwicklungen reagieren und versuchen, die Nutzeranforderungen mit den Sicherheitsbedürfnissen des Unternehmens in Einklang zu bringen.

## **8.1.2 Hemmnisse für mobile Business-Apps**

### **8.1.2.1 Vielfalt der mobilen Betriebssysteme**

Die Entwicklung auf dem Mobilfunkmarkt unterscheidet sich in Bezug auf die Betriebssysteme in einem wesentlichen Punkt von der Entwicklung im Desktop-Umfeld. In der Desktop- und Notebook-Welt dominiert Microsoft mit seinen Windows-Betriebssystemen und hat einen Quasistandard geschaffen. Dagegen wird bei den Smartphones und Tablet-PCs auf absehbare Zeit eine Vielfalt mehrerer Betriebssysteme eine signifikante Rolle spielen. Diese Entwicklung erschwert nicht nur das einheitliche Management von mobilen Endgeräten im Unternehmenskontext, sondern erhöht auch den Aufwand für die Integration in die Unternehmens-IT. Mobile Geschäftsanwendungen und deren entsprechende Schnittstellen müssen so für mehrere Betriebssystemplattformen parallel entwickelt und gepflegt werden.

Apple hat in nur wenigen Jahren mehr als 120 Millionen Geräte mit dem Apple-eigenen Betriebssystem iOS verkauft. Täglich gehen rund 250.000 Geräte über die Ladentheke. Während Apple eher das Hochpreissegment besetzt, ist inzwischen mit Android ein Betriebssystem für mobile Geräte auf dem Markt, das den Massenmarkt adressiert. Android erlebt am Markt gerade eine beispiellose Erfolgsgeschichte. War Android Anfang 2009 mit einem Marktanteil von 1,6 Prozent noch ein absolutes Nischenprodukt, läuft das Betriebssystem nach Angaben des britischen Marktforschungsunternehmens Canalys sowie des Marktforschungsunternehmens Gartner inzwischen auf mehr als einem Drittel der neu verkauften Smartphones. Laut Gartner [7] soll 2012 nahezu jedes zweite verkaufte Smartphone (49,2 Prozent) mit Android laufen.



**Abb. 8.3 „Worldwide Mobile Communication Device Open OS Sales to End Users by OS“**, Gartner, 04/2010, \* ab 2011 geschätzt

Research in Motion (RIM) ist mit seinem Betriebssystem für den Blackberry am längsten auf dem Markt. Schon 1999 begann die Erfolgsstory des Blackberrys, dass als erstes Business-Smartphone eine durchgehende Lösung für E-Mail, Kalender und Kontaktsynchronisation per Push-Dienst entwickelt und in den Unternehmen etabliert hat. RIM hat durch die starke Konkurrenz von Android und Apple deutlich an Boden verloren und muss sich im Markt mit großen Anstrengungen behaupten. Für RIM erwartet Gartner bis zum Jahr 2015 einen Marktanteilsverlust von 16,0 auf 11,1 Prozent (vgl. Abb. 8.3).

Weitere aktuelle Betriebssysteme sind Symbian-OS, das ausschließlich auf Smartphones von Nokia zu finden ist, und Windows Phone 7 (WP7). Die Zukunft dieser Betriebssysteme ist derzeit sehr ungewiss. Nokia Geräte mit Symbian-OS haben 2009/10 massiv an Marktanteilen verloren, WP7 muss erst noch zeigen welches Potenzial in ihm steckt.

### 8.1.2.2 Erhöhte Sicherheitsanforderungen mobiler Endgeräte

Aufgrund des Formfaktors und der mobilen Nutzungsszenarien ist die Gefahr des Verlusts oder Diebstahls bei Smartphones sehr viel höher als bei Notebooks. Gelangen die Geräte in falsche Hände, besteht die Gefahr, dass auf ihnen gespeicherte oder über sie zugreifbare Daten unbefugt ausgelesen werden. Die Absicherung von mobilen Endgeräten stellt daher eine ernst zu nehmende Herausforderung für die Unternehmen dar und ist gleichzeitig eine wesentliche Herausforderung für die breite Nutzung von mobilen Business-Applikationen.

Nach Schätzungen führender Marktforschungs- und Beratungsunternehmen befinden sich von allen im Unternehmensnetz abgelegten Daten mindestens 70 Prozent der Informationen parallel auch auf mobilen Geräten aller Art wie zum Beispiel in Form von E-Mail-Anhängen. Nur ein Viertel der Firmen verfügt über umfassende Verwaltungs- und Sicherheitsrichtlinien [8]. Dabei investieren Unternehmen weit über zehn Prozent ihrer IT-Budgets in Sicherheit. Doch mit den Smartphones ihrer Mitarbeiter gehen sie bislang meist eher sorglos um.

Die Nutzer sind sich oft nicht der Gefahr bewusst, dass auch Smartphones und Tablet-PCs prinzipiell anfällig für Viren oder Hackerangriffe sind. So haben laut einer Umfrage im Auftrag des BSI 47 Prozent noch nie ein Sicherheits-Update auf ihr Mobiltelefon aufgespielt. Unternehmen sind daher gut beraten, die Sicherheit von Endgeräten, die Zugriff auf Firmendaten haben, nicht dem Nutzer selbst zu überlassen [9].

### ***8.1.3 Status quo mobiler Business Anwendungen***

Eine der ersten und auch heute noch am weitesten verbreiteten mobilen Geschäftsanwendungen ist die Nutzung grundlegenden Office-Anwendungen wie E-Mail, Kalender und Kontakten auf mobilen Endgeräten. Zunächst im Wesentlichen der Managementebene vorbehalten, finden diese Anwendungen immer mehr Verbreitung in den Unternehmen.

Apps zur privaten Nutzung oder Business-Apps zur Steigerung der persönlichen Produktivität sind inzwischen weit verbreitet, die Entwicklung bei integrierten Business-Apps läuft jedoch wesentlich langsamer ab. Zwar gibt es in den diversen App-Stores eine große Anzahl von Apps, die auch geschäftliche Arbeitsabläufe unterstützen. Diese sind jedoch meist noch isoliert von den zentralen Business-Anwendungen und es fehlt eine Verzahnung mit den Prozessen der Unternehmen.

Sind in Unternehmen integrierte, mobile Unternehmensanwendungen vorhanden, stehen diese meist nur einer sehr beschränkten Nutzerzahl für ganz spezielle Anwendungsfälle zur Verfügung. Kennzeichnend für diese Anwendungen ist oft die dedizierte Kombination aus Gerätetyp, Betriebssystemplattform und mobiler Anwendung. Diese findet man z. B. bei Marketingbefragungen, in Abfertigungsterminals, beim Paketdienst oder mobilen Servicetechnikern.

Neben fehlenden technischen Voraussetzungen beim Einsatz von mobilen Applikationen im Unternehmenskontext sind auch die organisatorischen Herausforderungen im vielen Fällen noch nicht oder nur unzureichend gelöst. Nur wenige Unternehmen verfügen über ein umfassendes und durchgängiges Konzept für ihre Mobilitätsinfrastruktur. Es fehlt hierbei oftmals an verbindlichen Verhaltensvorgaben für den Umgang mit mobilen Endgeräten. Durch die Nutzungsszenarien, die mobile Endgeräte mit sich bringen, ergeben sich hier neue Herausforderungen. Bei PCs und Notebooks fällt die Trennung von privaten und unternehmenseigenen Geräten meist leicht: die einen im Büro und auf Geschäftsreise, die anderen zuhause. Mit den Smartphones und Tablet-PCs ändert sich dies – sie werden in viel stärke-

rem Maße als persönliche Geräte wahrgenommen. Wer viel unterwegs ist, muss sich bei strikter Trennung von beruflicher und privater Nutzung doppelt ausstatten: Zwei Handys oder Smartphones mit zwei Ladegeräten, zwei Rufnummern, zwei Headsets und ggf. noch zwei Tablet-PCs. Daher neigen Mitarbeiter schon aus pragmatischen Gründen dazu, verstärkt nur noch ein Gerät für den privaten und den geschäftlichen Lebensbereich zu nutzen. Fehlen klare Regelungen, kann das dazu führen, dass lokale Optimierungen und Lösungen entstehen, die nicht im Interesse des gesamten Unternehmens sind. So führt die Nutzung nicht zugelassener, privater Geräte Zugriffe auf das Unternehmensnetzwerk sowie das Umleiten von Informationsströmen in Form von E-Mails oder Telefonaten auf diese Geräte nicht nur zu Sicherheitsproblemen, sondern verhindert auch einen effizienten und kostengünstigen Support.

Das Potenzial, das sich durch den breiten Einsatz von mobilen Unternehmensapplikationen für relevante und oft genutzte Unternehmensprozesse ergibt, bleibt noch weitgehend ungenutzt.

## **8.2 Mobilisierung von Geschäftsprozessen durch Business-Apps**

Die universelle Verfügbarkeit von entscheidungsrelevanten Informationen ermöglicht es Unternehmen, agiler auf Veränderungen zu reagieren, und kann so einen Wettbewerbsvorteil verschaffen. Zwar können mobile Geschäftsanwendungen stationäre Systeme nicht ersetzen. Die Erweiterung um eine mobile Komponente kann jedoch den Geschäftsnutzen der klassischen Applikation signifikant erhöhen. Business-Apps mit Backendzugriff haben daher das Potenzial, sich als gleichwertiges Werkzeug zu etablieren.

Die komplementären Nutzenszenarien der mobilen Business-Apps ergeben auch aus den unterschiedlichen Formfaktoren von Notebooks auf der einen und Smartphones auf der anderen Seite. Mit Smartphones lassen sich Apps spontan auch im Stehen oder Gehen nutzen, während man sich mit Notebook auch unterwegs einen geeigneten „Arbeitsplatz“ suchen und genügend Zeit einplanen muss (z. B. für den Bootvorgang und den Verbindungsauflauf ins Unternehmensnetzwerk). Trotzdem behalten auch Notebooks ihre Daseinsberechtigung, da man mit ihnen Arbeit erledigen kann, die man auch in Zukunft nur eingeschränkt mit einem Tablet-PC/Smartphone durchführen wird. Smartphones und Tablet-PCs bilden damit eine neue Gerätekasse, die bestehende Notebooks ergänzt.

Beispiele für den nutzbringenden Einsatz mobiler Business-Anwendungen finden sich etwa im Personalwesen, im Vertrieb oder in der Beschaffung. Auf der Agenda könnten auch Social Networking, Informationsverarbeitung und die Unterstützung von mobilen Wissens- oder Service-Mitarbeitern sowie Außendienstkräften etwa durch Location Based Services stehen. Maschinenbauer geben ihren Kunden interaktive Bedienungsanleitungen mit. Versicherungsberater errechnen beim Kunden ein Angebot, generieren einen Vertrag und lassen ihn direkt auf dem Touchscreen-Display unterschreiben.

Bei der konkreten Ausgestaltung der Einsatzszenarien sind nicht allein Softwarehersteller wie SAP oder Oracle gefragt, die bestrebt sind, Ihre Business Software in die mobile Welt zu verlängern. Auch die Anwenderunternehmen müssen sich genau überlegen, welche Art von mobilen Anwendungen sich gewinnbringend einsetzen lassen und was für den effektiven Einsatz nötig ist. Unter Umständen müssen Prozesse und Rollenprofile an die Möglichkeiten des mobilen Arbeitens angepasst werden.

Im folgenden Abschnitt sollen einige konkrete Beispiele aufgezeigt werden, die verdeutlichen, wie durch den Einsatz mobiler Business-Apps betriebliche Abläufe optimiert und beschleunigt werden können.

### **8.2.1 Mobile CRM**

Ein attraktives Anwendungsfeld für mobile Business-Apps ist der CRM-gestützte Vertriebsprozess. Vertriebsmitarbeiter im Außendienst können effektiver arbeiten, wenn sie jederzeit Zugriff auf relevante Daten haben. Unternehmen, die ihren Mitarbeitern unabhängig von deren Aufenthaltsort Zugang zu geschäftskritischen Informationen ermöglichen, können einen echten Wettbewerbsvorteil erlangen.

Mobile CRM-Lösungen können dazu beitragen, die Datenqualität im System wesentlich zu verbessern. Vertriebsmitarbeiter können ihre Daten direkt nach einem Kundenbesuch per Smartphone aktualisieren, anstatt die Datenpflege in den Abendstunden oder am Wochenende zu erledigen. Dies hat den Vorteil, dass entscheidungsrelevante Informationen zeitnah sowohl den Kollegen als auch dem Management zur Verfügung stehen.

### **8.2.2 Mobile HR**

Gerade im produzierenden Gewerbe ist es für Mitarbeiter der Produktion oft mit größerem Aufwand verbunden, mit der Personalabteilung in Kontakt zu treten. Aufgrund der Arbeitsorganisation ist es häufig nicht ohne weiteres möglich, den Arbeitsplatz während der Schicht für ein Personalgespräch zu verlassen. Viele Mitarbeiter scheuen den Aufwand, so dass eine Interaktion zwischen Belegschaft und Personalabteilung nur in geringem Maße stattfindet.

Mit mobilen HR-Lösungen ist es nun möglich, dem Personalreferenten – ähnlich wie dem Vertriebsmitarbeiter im Falle Mobile CRM – relevante Daten über seine zu betreuenden Mitarbeiter auf einem mobilen Endgerät wie z. B. einem Tablet-PC verfügbar zu machen. Damit kann der Personalreferent auch unabhängig von seinem festen Arbeitsplatz Beratungsgespräche mit den Mitarbeitern führen – er kann also die Mitarbeiter an deren Arbeitsort aufsuchen und dort Beratungsgespräche führen. Genehmigungen von Abwesenheiten, Planung von Fortbildungsmaßnahmen, Anpassung von Arbeitszeitmodellen und ähnliche Aufgaben können vor Ort erledigt

werden. Die Daten werden unmittelbar mit dem HR-Backendsystem abgeglichen und aktualisiert.

## 8.3 Bausteine für ein Mobilitätskonzept

Ohne ein durchgängiges Mobilitätskonzept bleibt der Betrieb mobiler Geräte und die Entwicklung von Business-Apps mit oftmals (zu) hohem Aufwand verbunden und trotzdem anfällig für Fehler, Sicherheitsrisiken und technologische Sackgassen. Hier sollen nun einige Lösungsansätze vorgestellt werden, welche die Basis für eine integrierte Mobilitätsstrategie bilden.

Grundsätzlich muss eine solche Strategie Antworten auf die folgenden Fragen geben:

- Wie kann die Vielzahl der Gerätetypen und Betriebssysteme flexibel integriert werden?
- Wie kann der Zugriff auf Geräte sichergestellt werden, die unter Umständen weltweit verteilt sind und für klassische stationäre Service-Teams nicht greifbar sind?
- Wie können mobile Endgeräte gegen die zusätzlichen Sicherheitsrisiken abgesichert und Compliance-Richtlinien erfüllen werden?
- Wie können mobile Anwendungen flexibel und kostengünstig integriert, gewartet und betrieben werden?

Typischerweise umfasst ein solches Konzept folgende Bausteine:

### 8.3.1 Device Management

Angesichts der Entwicklung des Smartphone Markts (vgl. Abb. 8.3) sowie der Anforderung der Nutzer, die mobilen Endgeräte geschäftlich und privat nutzen zu können, müssen die IT-Abteilungen der Unternehmen hierzu technische Lösungen finden. Ein Device-Management-Konzept für mobile Endgeräte muss daher die unterschiedlichen Endgerätetypen und Betriebssystemplattformen integrieren können und wesentlich über das klassische Device Management im Desktop- Bereich hinausgehen.

Grundsätzlich betrachtet mobiles Device Management die Verwaltung der mobilen Endgeräte über den gesamten Lebenszyklus hinweg. Dazu gehören deren Beschaffung, Reparatur, Problembehebung, Sicherheit (Lösung von Daten bei verlorenen Geräten) sowie Software- und Lizenzverwaltung. Device-Management-Lösungen, die mehrere Plattformen unterstützen, kombinieren Flexibilität auf Seiten der Geräte mit verbindlichen Sicherheitsrichtlinien. Mit diesen Lösungen lassen sich Endgeräte und Software verwalten, Business-Anwendungen verteilen und aktualisieren und Geräte-Einstellungen per Fernzugriff ändern. Von einem zentralen

Kontrollpunkt aus sind plattformübergreifende Backups möglich. Bei Verlust können die Geräte gesperrt und alle Daten auf ihnen gelöscht werden.

Bei der Auswahl einer Lösung ist darauf zu achten, dass eine möglichst umfassende Integration geboten wird. Alle Geräte, die im Unternehmen verwendet werden, sollten unterstützt und von einer Schnittstelle aus verwaltet werden. Das gilt sowohl für Smartphones als auch für Tablet-PCs.

### ***8.3.2 Entwicklung und Management mobiler Anwendungen***

Die Entwicklung und das Management für mobile Anwendungen stellen zusätzliche Anforderungen. Eine geeignete Mobilisierung von Teilespekten aus betrieblichen Kernsystemen erfordert Überlegungen, sowohl was die fachliche Sinnhaftigkeit angeht als auch was die geeignete mobile Darstellung, Aufbereitung und Synchronisation der Daten und Funktionalitäten auf verschiedenen Geräteklassen betrifft.

Die folgenden, wesentlichen Kernfragen sind hierzu zu beantworten:

- Welche Funktionalitäten im Unternehmen sollen mobilisiert werden?
- Welche Prozesse eignen sich für eine Mobilisierung und in welchem Umfang müssen diese Prozesse verändert werden?
- Welche Applikationen lassen sich wie mobilisieren? Lassen sich wiederverwendbare Funktionalitäten kapseln und hohe Anpassungsaufwände verhindern (z. B. mit Entwicklungsguidelines und mobilen Middlewareansätzen)?
- Wie lassen sich die Applikationen während ihres Lebenszyklus aktuell halten (Rollout, Updates, Patches, Change Requests, Lauffähigkeit auf neuen Gerätetypen, Testing,...)?
- Betrieb und Aktualisierung der Mobilisierungsplattform?

Hierbei müssen Fachabteilungen und IT-Abteilungen gemeinsam entscheiden, welche mobilen Anwendungen einen eindeutigen Mehrwert bieten, der idealerweise auch monetär nachweisbar wird. Voraussetzung dafür ist, dass die mobile ICT-Landschaft eine transparente Kostenstruktur aufweist, die man gegenüberstellen kann, insbesondere wenn zunächst Mehrkosten im Vergleich zum Status-Quo für Anschaffung und Betrieb verursacht werden.

### ***8.3.3 Enduser-Support***

Auch der Enduser-Support gestaltet sich schwieriger und kostet mehr Geld als bei reinen Desktoplandschaften. Es muss klar definiert sein, wie Service-Teams die Geräte zeitnah prüfen und gegebenenfalls reparieren können, und es muss ein ausreichend großes Service-Team für den telefonischen Support zur Verfügung stehen. Per Fernzugriff lässt sich ein erheblicher Anteil von Störungen beheben, ohne dass das Gerät eingeschickt oder abgeholt werden muss.

### 8.3.4 Sicherheitskonzept

Besonders der mobile Zugriff auf sensible Geschäftsdaten wie die oben erwähnten Kunden- und Mitarbeiterdaten erfordert neue Sicherheitskonzepte und Richtlinien. Zudem sollen die Compliance-Anforderungen der Unternehmen auch auf mobilen Endgeräten sicher durchsetzbar sein. Diese Anforderungen müssen mit geeigneter Gerätekonfiguration, Managementwerkzeugen und Remote-Zugriff über den gesamten Lebenszyklus erfüllt werden können. Dazu gehören auch Richtlinien für das Verhalten und den Umgang mit mobilen Geräten, Apps und Services. Ein technisch hoch gesichertes Gerät lässt sich auch an einem öffentlich einsehbaren Ort unsachgemäß einsetzen.

Ein großer Teil der Sicherheitsanstrengungen sollte der Frage gewidmet sein, wie die Anwendungen auf den mobilen Endgeräten mit den Unternehmensservern interagieren, wo tatsächlich Daten vorgehalten werden, und wie die verschiedenen Schutzmechanismen ineinander greifen.

Für Unternehmen und Organisationen mit besonders hohen Sicherheitsanforderungen oder für das Top-Management, die mit besonders sensiblen Unternehmensdaten umgehen, sollte man bereits heute verfügbare speziellere Lösungen in Betracht ziehen. Sie sichern die Datenverbindungen zu den Unternehmensnetzwerken, den Zugriff auf die Unternehmensdaten und verschlüsseln zusätzlich die Sprachverbindung. So ist der sichere Versand von E-Mails, Kalender-Einträgen und Kontakten möglich. Der sichere Zugang läuft über eine hochsichere VPN-Verbindung. Solche Smartphones arbeiten mit einer „digitalen Identität“, also Zertifikaten, die von einer zentralen vertrauenswürdigen Stelle zur Verfügung gestellt werden. Alle sicherheitsrelevanten Aktionen – Zugang, Schlüsselerzeugung, Ver- und Entschlüsselung – funktionieren ausschließlich mit diesem Zertifikat und der dazugehörigen PIN. So sind die Smartphones auch bei Verlust noch geschützt.

## 8.4 Der Weg zu einer Mobile Enterprise Strategie

Es stellt sich für jedes Unternehmen, das die Vorteile mobiler Lösungen erschließen möchte, die Frage nach einem ganzheitlichen Zielbild, einer „Mobile Enterprise Strategie“ und auch nach dem Weg dorthin. Ein Beispiel für eine technische Zielarchitektur ist in Abb. 8.4. dargestellt. PIM-Services (E-Mail, Kontakte, Kalender) sind häufig der Ausgangspunkt und die wichtigste Anforderung. Für mobile Business-Applikationen erfolgt die Kommunikation zwischen Endgerät und Server über eine mobile Middleware, die den Zugriff auf verschiedenste Backendsysteme kapselt und die Daten für eine Mobilisierung per Push oder Synchronisation auf verschiedene Endgeräteplattformen entsprechend aufbereitet. Geräte können via Mobile-Device-Management zentral mit einheitlichen Applikationen, Daten und Konfigurationseinstellungen versorgt werden. Die Übertragungsstrecken sind entsprechend abgesichert.

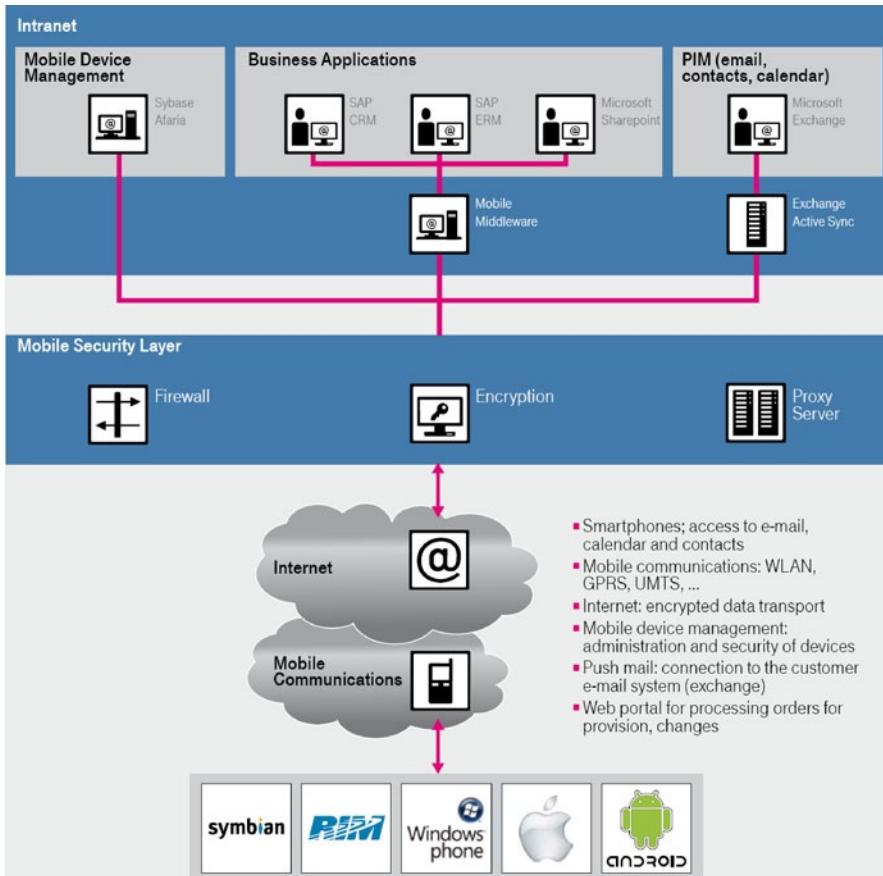
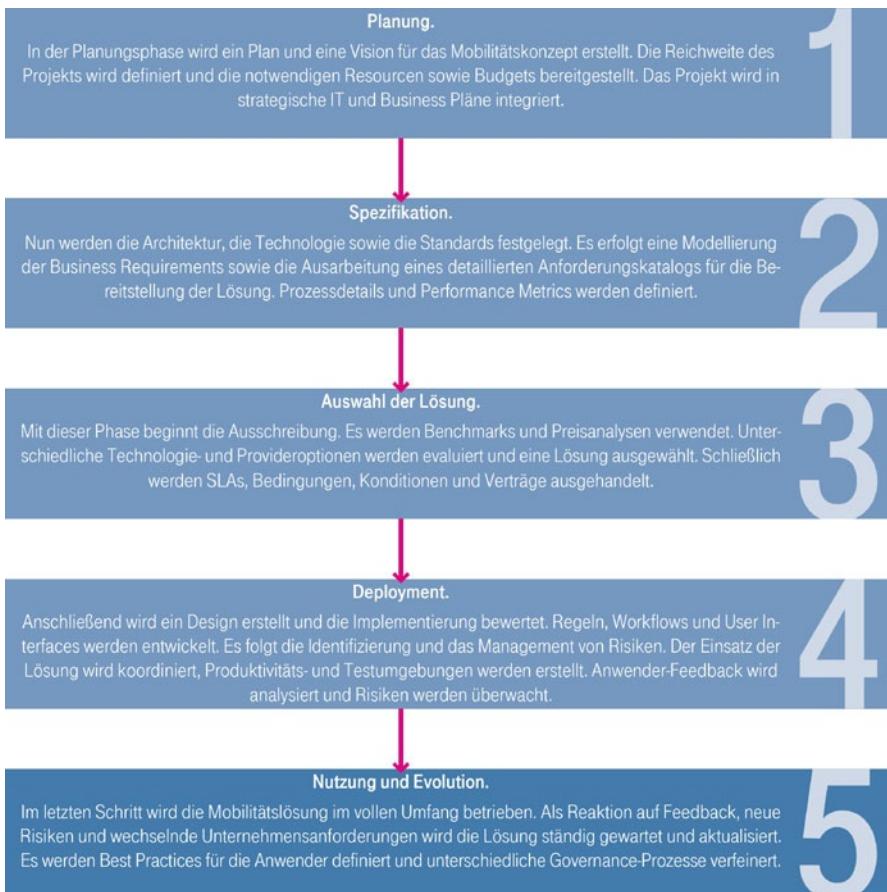


Abb. 8.4 Exemplarischer Aufbau einer Mobile-Enterprise-Infrastruktur

Nicht immer verfügen die internen IT-Abteilungen über ausreichende Kapazitäten und Know-how, ein solches Zielbild auszugestalten oder zu betreiben. Eine Option ist in diesem Fall die Zusammenarbeit mit spezialisierten Telekommunikations- und IT-Dienstleistern bis hin zum Outsourcing der Mobilitätsinfrastruktur und der mobilen Anwendungsentwicklung. Insbesondere für die Komplexität und Dynamik im mobilen Bereich werden sich hier spezielle Anbieter für Teilfunktionen, Managementsoftware, aber auch integrierte Mobilitätsdienstleistungen herausbilden, die das Risiko für ein Einzelunternehmen, in eine technologische Sackgasse zu laufen, minimieren.

Ein mögliches Vorgehen zur Entwicklung einer Mobile-Enterprise-Strategie sieht man in Abb. 8.5.



**Abb. 8.5** Vorgehen zur Umsetzung einer Mobile-Enterprise-Strategie

## 8.5 Zusammenfassung/Ausblick

Es bleibt festzustellen, dass

- der Anteil der Smartphones steigen wird,
- Endgeräte immer leistungsfähiger werden,
- die Heterogenität der Geräte und Betriebssysteme bestehen bleibt.
- der Bedarf, Aufgaben mobil zu bearbeiten, immer größer wird und
- daher die Herausforderungen an die IT-Abteilungen nicht kleiner werden.

Berücksichtigen Unternehmen die individuellen Vorlieben der Mitarbeiter, steigt laut Gartner die Zufriedenheit der Nutzer und damit ihre Produktivität. So hat iPass in einer Studie messbare Produktivitätsgewinne festgestellt [10], Wenn Angestellte mit mobilen Geräten persönliche und berufliche Anliegen erledigen dür-

fen. Laut iPass-Studie arbeiten sie im Schnitt 240 Stunden mehr pro Jahr. Den IT-Verantwortlichen raten Analysten wie Gartner, IDC oder Forrester deshalb, sich auf das kommende IT-Zeitalter vorzubereiten und sich von den herrschenden Vorurteilen zu verabschieden.

Für Forrester wird Mobilität dann selbstverständlich sein, wenn die Endgeräte die Beschränkungen von heute überwunden haben: Schmale Bandbreiten, geringe Akkulaufzeit und Speicherkapazitäten, hohes Gewicht, mangelhafte Dateneingabe werden, so Forrester, in ein paar Jahren kein Thema mehr sein.

Für diese nicht allzu ferne Zukunft heißt das: Geschäftsanwendungen werden auf jedem beliebigen Endgerät laufen. Manche davon liegen gut in der Hand, andere bleiben besser auf dem Schreibtisch. Mobiles Arbeiten wird in vielen, aber nicht in allen Geschäftsbereichen für mehr Produktivität der Mitarbeiter sorgen und die Geschwindigkeit in der Ausführung durch besser ausgefüllte Zeit und geschlossene Informationslücken erhöhen.

Fazit: Wenn es gelingt, mit einer durchgängigen Mobilitätsstrategie die Komplexität der Entwicklung und des Betriebs in den Griff zu bekommen, bietet sich Unternehmen die Chance, ihre Produktivität zu erhöhen, Prozesslaufzeiten zu verringern und letztlich die Wettbewerbsfähigkeit zu steigern. Es wird damit auch ein volkswirtschaftlicher Nutzen erzeugt, der zum Wirtschaftswachstum beiträgt.

## Literaturverzeichnis

1. Canalys. Worldwide smartphone market, market shares Q4/2010, Q4/2009
2. Bitkom (2010) Presseinformation. Smartphone-Absatz 2011 über der 10-Millionen-Marke. [http://www.bitkom.org/de/presse/66442\\_65897.aspx](http://www.bitkom.org/de/presse/66442_65897.aspx) 28. Oktober 2011
3. Bitkom (2011) Presseinformation. Tablet-PCs boomen. [http://www.bitkom.org/de/themen/54894\\_67058.aspx](http://www.bitkom.org/de/themen/54894_67058.aspx) 28. Oktober 2011
4. IDC (2010) Worldwide Mobile Worker Population 2009–2013 Forecast
5. CIO Magazin. Apple iPads in SAP einbinden. [http://www.cio.de/knowledgecenter/mobile\\_it/2276204/](http://www.cio.de/knowledgecenter/mobile_it/2276204/) 28. Oktober 2011
6. Mobile360 (2011) Mobile Facebook-Nutzer sind aktiver. <http://mobile360.de/mobile-facebook-nutzer-sind-aktiver-34603.html> 28. Oktober 2011
7. Gartner (2011) Forecast: Mobile Communications Devices by Open Operating System, Worldwide, 2008–2015
8. T-Systems (2010) White Paper Mobile Enterprise
9. BSI Pressemeldung (2011) *BSI für Bürger* mit neuem Internetauftritt
10. iPass. The iPass Global Mobile Workforce Report. 05/2011 <http://mobile-workforce-project.ipass.com/reports/mer> 28. Oktober 2011

## **Teil IV**

# **Anwendersicht: Herausforderungen der Anwendungsbranche**

---

# Kapitel 9

## Anwendungssicht mobiler Geschäftsanwendungen

Léopold Kuassi und Michael Bischel

**Zusammenfassung** Mobile Applikationen auf modernen Mobiltelefonen und Smartphones sind in der Lage Daten zu visualisieren, zu übermitteln und zu empfangen. So können aufeinander abgestimmte Komponenten genutzt werden um Geschäftsprozesse zu unterstützen und Geschäftsanwendungen mobil abzubilden. Für ihre Nutzung wird somit kein vollwertiger Computer benötigt. Welche Anforderungen an mobile Geschäftsanwendungen bestehen seitens der Anwendungsbranche? Untersucht werden Anforderungen von Mitarbeitern, Endkunden und Maschinen, die als Anwender von mobilen Geschäftsanwendungen in Erscheinung treten. Dabei werden auch die Bedürfnisse der Unternehmens-IT beim Einbinden der Geschäftsanwendung berücksichtigt. Es ergeben sich Anforderungen an die auf den einzelnen Endgeräten laufenden Applikationen, Anforderungen an die daraus resultierende Geschäftsanwendung als Ganzes und Anforderungen an die Plattform, auf der diese ausgeführt werden.

### 9.1 Anforderungen seitens der Anwendungsbranche

Mobile Geschäftsanwendungen sind allgemein Programme, die auf modernen Mobiltelefonen oder Smartphones ausgeführt werden um Geschäftsprozesse zu unterstützen. Das tragbare Endgerät wird genutzt um Daten zu visualisieren, zu übermitteln oder zu empfangen. Der Bedarf an mobilen Geschäftsanwendungen ist sehr stark in der Dienstleistungsbranche, in denen die Mitarbeiter außerhalb des Bürogebäudes des Unternehmens ihre Tätigkeiten erledigen müssen und dabei schnelle

---

Léopold Kuassi  
LeoSoft GmbH, Haar b. München,  
E-mail: info@leosoftsys.com

Michael Bischel  
LeoSoft GmbH, Haar b. München,  
E-mail: info@leosoftsys.com

und aktualisierte Information benötigen. Das Befüllen von vernetzten Konsumautomaten wie z. B. Getränkeautomaten, der Einsatz von ambulanter Pflegepersonal für die vernetzten Pflegebedürftigen oder der Express-Kurierdienst sind einige Beispiele aus der Anwendungsbranche.

In den folgenden Abschnitten wird anhand eines praktischen Beispiels einer mobilen Geschäftsanwendung ein Verständnis für die Anforderungen mobiler Anwendungen vermittelt. Anschließend werden sukzessiv der Systemkontext, der Anwendungskontext und der Plattformkontext erläutert. Anschließend folgt eine Zusammenfassung mit einer Liste der exemplarisch abgeleiteten Anforderungen (Tabelle 9.8); sie beinhaltet alle exemplarisch abgeleiteten Anforderungen der einzelnen Abschnitte (Tabellen 9.1–9.7).

Das System eines vernetzten Getränkeautomaten, dessen Vorrat zur Neige geht, könnte die Ergebnisse seiner Inventur an die entsprechende Gegenstelle übermitteln, um zu signalisieren, welche Waren in welchem Umfang aufzufüllen sind. Im Lager geht eine Bestellung ein, die ein Mitarbeiter mit den weiteren Bestellungen zusammenstellt. Er könnte von einer Applikation dabei unterstützt werden, unter anderem um die Routen für die Auslieferung vorzubereiten. Der für die Auslieferung der Ware zuständige Mitarbeiter ist unterwegs, um die Automaten zu befüllen. Er folgt der Route, die ihm die Applikation auf seinem Smartphone vorschlägt. Von Station zu Station zeigt ihm die Applikation an, was er in welcher Menge auffüllen darf. Eine der umsatzstärksten Stationen hat vor wenigen Minuten ihre Nachbestellung losgeschickt. Die Applikation des Lieferanten veranlasst ihn daraufhin seine aktuelle Station nur teilweise zu befüllen, damit genügend Ware zur Verfügung steht, um den Notstand der zwischenzeitlich eingetroffenen und wichtigeren Bestellung auszugleichen.

Mobile Applikationen können wie spielerische Features auf schicken Smartphones wirken, sie haben aber auch das Potential, als gut abgestimmte Komponenten zur Unterstützung von Geschäftsprozessen genutzt zu werden. Wegen der höheren Mobilitätsbedürfnisse in vielen Geschäftsbranchen wird die Nachfrage nach mobilen Geschäftsanwendungen immer größer. Mobile Geschäftsanwendungen sind keine komplett neue Erfindung. Sie existieren bereits in der Welt von Notebooks, allerdings sind die Notebooks als mobile Geräte in einigen Geschäftsbranchen unpraktisch, weil sie z. B. für den Transport zu groß sind, um nur ein Beispiel zu nennen. Die Smartphones sind heutzutage leistungsfähiger geworden und haben die Funktion eines kleinen Rechners bekommen. Die Herausforderung ist, die Systemgrenze der mobilen Geräte zu erkennen und die mobilen Geschäftsanwendungen so zu entwickeln, dass sie im Systemkontext passen und auf den unterschiedlichen mobilen Endgeräten lauffähig und trotzdem anwenderfreundlich sind. Dabei sind bei der Entwicklung mobiler Geschäftsanwendungen die Anforderungen der Anwendungsbranche mit den zur Verfügung stehenden technologischen Mitteln umzusetzen. Während der Entwicklungsphase einer mobilen Geschäftsanwendung soll die Anwendungssicht besonders betrachtet werden, um die unterschiedlichen Aspekte aus der Sicht der Beteiligten in die Entwicklung einfließen zu lassen. Die beteiligten Systemnutzer sind:

- Die Unternehmen, die an dem Geschäftsprozess beteiligt sind, die das System in ihr Unternehmen einbinden und ihren Kunden und Mitarbeitern zur Verfügung stellen.
- Die Mitarbeiter des Unternehmens, die das System bedienen um den Geschäftsprozess auf Unternehmenseite zu unterstützen.
- Die Endkunden des Unternehmens, die das System auf Kundenseite eventuell nutzen sollen.

Die Betrachtung der unterschiedlichen Aspekte in der Entwicklung einer mobilen Geschäftsanwendung stellt eine große Herausforderung für das System dar. Im nächsten Kapitel werden sowohl die Auswirkungen der Anforderungen an das System als auch die Komplexität des Systems erläutert.

## 9.2 Systemkontext

Definition Systemkontext:

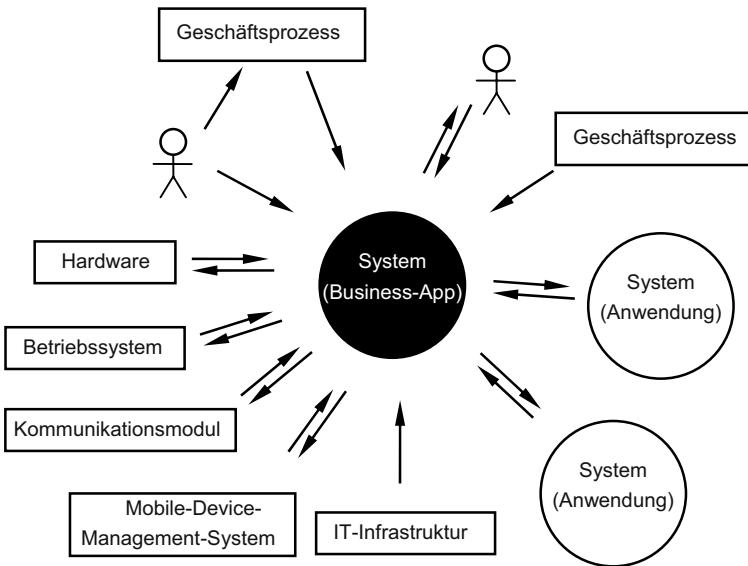
*Der Systemkontext ist der Teil der Umgebung eines Systems, der für die Definition und das Verständnis der Anforderungen des betrachteten Systems relevant ist [1].*

Eine mobile Geschäftsanwendung als System hat per se vielfache Abhängigkeiten zu weiteren Komponenten. Eine mobile Geschäftsanwendung wird in den meisten Fällen durch das Zusammenspiel mehrerer Applikationen und weiterer Systeme entstehen, wie es Abb. 9.1 darstellt. Jede dieser Applikationen ist als ausführbares Programm zunächst abhängig von einer bestimmten Laufzeitumgebung eines Betriebssystems.

*Eine Laufzeitumgebung lädt von Anwendungsprogrammierern entwickelte Programme und lässt diese auf einer Plattform ablaufen, für welche die Laufzeitumgebung gemacht worden ist. Damit stellt sie selbst eine kleine „Plattform“ dar, auf der Programme aufsetzen können. [2]*

Das Betriebssystem ist wiederum von der Hardware abhängig, auf der es läuft. In einem Systemkontext müssen alle Bestandteile oder Komponenten aufeinander abgestimmt werden, damit das gesamte Konstrukt richtig funktionieren kann. So umfasst bei mobilen Geschäftsanwendungen der Systemkontext mindestens folgende Bestandteile:

- Hardware
- Betriebssystem
- Softwareplattform
- Laufzeitumgebung
- Human Machine Interface (HMI)
- Kommunikationsmodul
- Gegenstellen (weitere Applikationen oder Serveranwendungen)
- Geschäftsprozess
- Anwender
- IT-Infrastruktur im Unternehmen



**Abb. 9.1** Komponenten und Interaktion im Systemkontext (vgl. [3])

Anhand der o. g. Bestandteile des Systemkontexts ist die Komplexität in der Kommunikation zwischen den einzelnen Komponenten auch deutlich zu erkennen. Jede der Komponenten hat in ihrer Eigenschaft eine eigene Systemgrenze, die im Zusammenspiel zwischen allen Bestandteilen des Systemkontexts eine wichtige Rolle spielt. In unserem Beispiel einer mobilen Geschäftsanwendung – „das Befüllen von vernetzten Getränkeautomaten“ – besteht das System aus den o. g. Komponenten, die in den nächsten Abschnitten näher erläutert werden.

### 9.2.1 Systemgrenze

Jedes Anwendungssystem besteht einerseits aus einem vorgegebenen und unveränderbaren und andererseits aus einem gestaltbaren Teilsystem. Mit der Systemgrenze wird festgelegt, welche Komponenten im Rahmen der Entwicklungsaktivitäten gestaltbar sind und welche als gegeben und unveränderlich gelten (vgl. [4]). Die Komponenten im Systemkontext, die außerhalb der Systemgrenze liegen, auf die man also bei der Entwicklung keinen Einfluss hat, müssen bei der Anforderungsentwicklung berücksichtigt werden, damit die gestaltbaren Komponenten mit ihnen richtig interagieren können. Sie liefern sozusagen Rahmenbedingungen, auf die beim Ermitteln der Anforderungen an gestaltbare Komponenten eingegangen werden muss.

Eine mobile Geschäftsanwendung als Ganzes entsteht durch das Zusammenspiel der einzelnen mobilen Applikationen untereinander sowie mit weiteren Systemen und Anwendungen. Betrachten wir unseren Entwicklungsspielraum global und haben die Entwicklung einer Plattform für mobile Geschäftsanwendungen im Sinn, oder sehen wir die Plattform als gegeben und betrachten eine Anwendungssuite als System, das aus mehreren Einzelsystemen besteht (seien es mobile oder sonstige Komponenten)?

Vor dem Hintergrund einer plattform- und betriebssystemübergreifenden Architektur haben beide Betrachtungen ihre Berechtigung, sie ergänzen sich und haben unterschiedliche Stakeholder. Die jeweilige Sichtweise entscheidet, welches System betrachtet wird, und legt den Kontext für die Anforderungen fest.

Wir betrachten beide Sichtweisen und reden von **Plattformsystem** im globalen Sinn und von **Anwendungssystem** im engeren Sinn. Die Systemgrenze ist jeweils eine andere.

**Plattformsystem:** Gesamtheit der entwickelten Architekturkomponenten, die es ermöglichen, weitestgehend hardware- und betriebssystemübergreifende mobile Geschäftsanwendungen zu entwickeln und auszuführen.

**Anwendungssystem:** Gesamtheit der Applikationen, die zur Abbildung konkreter Geschäftsprozesse aufeinander abgestimmt zu einer mobilen Geschäftsanwendung entwickelt werden.

**Business-App:** Einzelne Softwarekomponente, die auf einem mobilen Endgerät ausgeführt wird und zur Abbildung eines Geschäftsprozesses beiträgt.

## 9.2.2 Kontextgrenze

Zur Untersuchung, welche Komponenten innerhalb der Kontextgrenze des Systems liegen und Einfluss auf die Entwicklung mobiler Geschäftsanwendungen haben (vgl. [5]), kann die Betrachtung der nachfolgenden Fragen von zentraler Bedeutung sein:

- Welche existierenden Softwaresysteme sind an dem Geschäftsprozess beteiligt?
- Auf welchen Endgeräten sollen die Softwarekomponenten ausgeführt werden?
- Mit welchen Betriebssystemen sind diese Geräte ausgestattet? Daraus resultieren noch weitere Fragen wie die Lizenzvorschriften etc.
- Welche Bedingungen müssen erfüllt werden, damit die Softwarekomponenten in die IT-Infrastruktur des Unternehmens integriert werden können?

Aus den oben gestellten Fragen ist eine Interaktion zwischen den einzelnen Komponenten erkennbar. Somit können wir im Systemkontext von verschiedenen Teilsystemen sprechen, die miteinander über das Kommunikationsmodul kommunizieren. Diese Teilsysteme definieren auch die Kontextgrenze. Einige Beispiele für Teilsysteme im Systemkontext sind Geschäftsprozess, Benutzer und Anwendungen auf einer Seite und Hardware und Betriebssystem auf der anderen Seite.

## Geschäftsprozess, Benutzer und Anwendungen im Anwendungssystem

Ziel einer mobilen Geschäftsanwendung ist die mobile Unterstützung von Geschäftsprozessen. Hierzu werden mehrere Teilsysteme entwickelt, die aufeinander abgestimmt ein gesamtes Anwendungssystem ergeben, das die Möglichkeit bietet, mobil in den Geschäftsprozess sowohl auf Kunden- als auch auf Unternehmensseite einzutreten; sei es als Mensch oder als Maschine.

### Kommunikationsmodul

Das Kommunikationsmodul mobiler Endgeräte spielt bei Business-Apps insofern eine zentrale Rolle, dass es die Schnittstelle zu weiteren Anwendungen des Anwendungssystems darstellt. Die Verbindung über den Mobilfunkanbieter, über WLAN oder Bluetooth, zu einer Gegenstelle kann gerade bei mobilen Endgeräten häufig gestört sein. Welche Arten von Anforderungen sich aus diesem Umstand ergeben, wird bei der Verfügbarkeit der Gegenstelle und des Netzes im Abschn. 9.3.2 und 9.3.3 behandelt.

### Hardware und Betriebssystem

Auf welchen Endgeräten eine Business-App ausgeführt werden soll, hängt einerseits von der Plattformunabhängigkeit und andererseits von dem Entwicklungsfortschritt und der Stabilität der Plattform ab. Im Java-Umfeld gibt es die MIDP-Spezifikation (Mobile Information Device Profile), sie beschreibt ein Profil der Java Micro Edition (Java ME) für Applikationen (MIDlets) auf mobilen Endgeräten. MIDlets sind weitestgehend hardware- und betriebssystemunabhängig, da sie innerhalb einer virtuellen Maschine laufen und diese für die unterschiedlichen Plattformen zur Verfügung steht. Diese Technologie konnte sich trotz ihrer guten Portabilität nicht durchsetzen. Im Vergleich hierzu hat man bei der Entwicklung von nativen Apps für ein bestimmtes mobiles Betriebssystem wesentlich mehr Möglichkeiten, büßt allerdings die Plattformunabhängigkeit ein. Native Apps können nur auf Geräten ausgeführt werden, die dem Betriebssystem, für das sie entwickelt wurden, entsprechen. So müsste eine native Business-App praktisch für jede der gängigen mobilen Plattformen entwickelt werden, damit sie auf diesen zur Verfügung steht: Apples iOS, Googles Android, Microsofts Windows Phone, BlackBerries OS von Research In Motion und Nokias Symbian. Doch jede dieser Plattformen hat ihre Eigenarten, so dass ein gemeinsamer Nenner für eine Business-App nicht leicht zu finden ist. Eine Alternative zu den nativen Apps stellen Webanwendungen dar, die über den Webbrower eines Endgerätes aufgerufen werden können.

Das entwickelte System wird nur das leisten können, was im Rahmen der Hardware des Endgeräts, mit dem das System ausgeführt wird, möglich ist. Ist ein Smartphone z. B. nicht mit GPS ausgestattet, werden von diesem Endgerät keine GPS-

Informationen an das entwickelte System zur Verfügung gestellt, womit die entsprechende Funktionalitäten im System nicht genutzt werden können. Bei den Anforderungen an das System wird man dies berücksichtigen müssen, indem geprüft wird, ob auf Lokalisierungsinformationen des Netzbetreibers ausgewichen werden kann, die GPS-Funktionalität als optional anzusehen ist oder ob solche Endgeräte ungeeignet sind, um das entwickelte System auszuführen.

Mobile Endgeräte sind in der Regel mit Betriebssystem erhältlich. Aus diesem Produktpaket resultiert ein Plattformangebot, dessen Konzepte die Umsetzbarkeit von Anforderungen des entwickelten Systems beeinflussen, insbesondere in Hinblick auf die Anbindung der Endgeräte in die IT-Infrastruktur des Unternehmens.

#### IT-Infrastruktur und „Mobile Device Management System“ (englisch für Administrationssoftware für mobile Endgeräte)

Auch die Möglichkeiten zur Einbindung einer App in die bestehende IT-Landschaft eines Unternehmens variieren je nach Plattform der eingesetzten Endgeräte, auf denen sie ausgeführt werden soll. Ein Unternehmen wird seine internen Business-Apps ausschließlich seinen Mitarbeitern zur Verfügung stellen wollen.

Solche betriebsinternen Anwendungen über eine für jeden zugängliche Softwarevermarktplattform zu verteilen, ist verständlicherweise ein unakzeptabler Umstand. Darauf wird näher im Abschnitt über die Besonderheit von mobilen Apps im Kontext des Plattformsystems eingegangen.

Ein Unternehmen, das seinen Mitarbeitern Business-Apps zur Verfügung stellt, wird aus Sicherheitsgründen die Installation von Apps sowie die Konfiguration der Endgeräte kontrollieren wollen. Somit kann im Unternehmen gewährleistet werden, dass die erwarteten Versionen der Business-Apps zum Einsatz kommen. Doch auch die Mitarbeiter werden eigenständig Apps für berufliche oder private Zwecke auf ihren Endgeräten installieren wollen. Verwenden die Mitarbeiter ihre eigenen privaten Endgeräte für berufliche Zwecke, werden sie die Forderung ihres Unternehmens, die absolute Kontrolle über ihre Smartphones und die darauf installierten Apps, kaum akzeptieren wollen. In diesem Fall kann das Unternehmen den Mitarbeitern Endgeräte für berufliche Zwecke zur Verfügung stellen. Eine besondere Anforderung an das System ist, den Zugriff von Unternehmen auf die Privatsphäre durch Konfiguration und Zugriffsrechte zu unterbinden. Hierauf wird detaillierter im Abschn. 9.4 im Kontext des Plattformsystems eingegangen. Bei Business-Apps, die auf Kundenseite genutzt werden, gestaltet sich dieser Umstand schwieriger; allerdings werden diese nicht direkt auf betriebsinterne Ressourcen des Unternehmens zugreifen. Jedenfalls dürfte die Konfiguration der Endgeräte von Endkunden außerhalb der Kontrollmöglichkeit des Unternehmens liegen, das die Business-App anbietet.

In unserem Beispiel einer mobilen Geschäftsanwendung – „das Befüllen von vernetzten Getränkeautomaten“ – muss das Unternehmen sich für einen Systemkontext

**Tabelle 9.1** Abgeleitete Anforderungen im Systemkontext

id	Name	Anforderung	Systemkontext	Referenz
Req-4	App-Verteilung	Der Mechanismus zur Verteilung der Business-Apps auf mobile Endgeräte muss auf jeder Zielplattform die Anforderungen der Unternehmens-IT unterstützen.	Plattformsystem	9.2.2
Req-4.1	App-Verteilung Gruppe	Die Unternehmens-IT muss auf jeder Zielplattform die Verteilung auf die vorgesehene geschlossene Benutzer- und Endgerätegruppe sicher beschränken können.	Plattformsystem	9.2.2
Req-5	Endgerätekonfiguration	Die Unternehmens-IT muss sicherheitskritische Konfigurationen auf den mobilen Endgeräten der Anwender auf jeder Zielplattform kontrollieren können.	Plattformsystem	9.2.2
Req-5.1	Version und Sicherheit	Insbesondere soll die Unternehmens-IT die Installation und Aktualisierung von Business-Apps sowie die Einhaltung benötigter Sicherheitseinstellungen auf den mobilen Endgeräten kontrollieren können.	Plattformsystem	9.2.2, 9.4
Req-5.1.1	Konfiguration Kontext	Wenn die Benutzer von Business-Apps ihre Endgeräte auch privat nutzen, müssen die Business-Apps auf den Endgeräten innerhalb eines Kontexts laufen, den die Unternehmens-IT konfigurieren kann.	Plattformsystem	9.2.2, 9.4
Req-5.1.1.1	Konfiguration virtueller Kontext	Die Unternehmens-IT kann möglicherweise durch Einsatz von Virtualisierungssoftware und „Mobile Device Management Software“ die Konfiguration im Business-App-Kontext auf den Endgeräten kontrollieren.	Plattformsystem	9.2.2, 9.4
Req-6	Plattform-unabhängigkeit Grundfunktion	Wenn eine Business-App plattformübergreifend eingesetzt wird, muss ihre Grundfunktionalität auf den mobilen Endgeräten aller unterstützten Zielplattformen laufen.	Plattformsystem	9.2.2

entscheiden, der seinen Geschäftsprozess zum größten Teil unterstützt. Das bedeutet auch, sich auf ein bestimmtes Hardware-Betriebssystem-Paket festzulegen. Das Unternehmen ist somit abhängig von diesem System und muss im schlimmsten Fall seinen Geschäftsprozess entsprechend anpassen, um die nicht erfüllten Anforderungen auszugleichen.

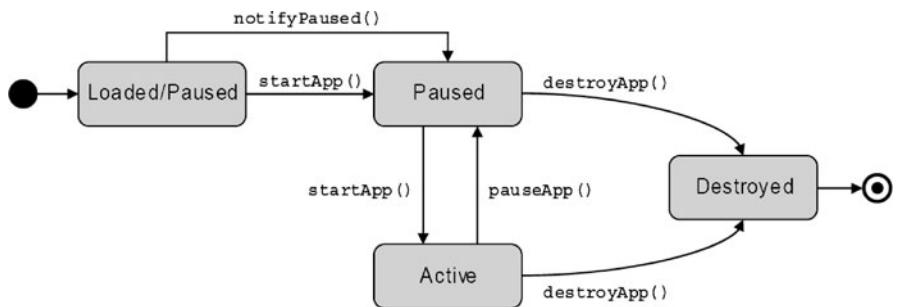


Abb. 9.2 Lebenszyklus eines MIDlets [6]

## 9.3 Besonderheit von mobilen Apps im Kontext des Anwendungssystems

Im Lebenszyklus mobiler Applikationen gibt es grundsätzlich drei Zustände: „active“ (in Deutsch „aktiv“), „paused“ (in Deutsch „pausierend“) und „destroyed“ (in Deutsch „zerstört“; hier ist „beendet“ gemeint). Abbildung 9.2 zeigt ein Zustandsdiagramm, das den Lebenszyklus eines MIDlets darstellt. Die einzelnen Plattformen mögen diese Zustände unterschiedlich benennen und eigene Variationen gebildet haben, der Grundgedanke ist aber bei allen mobilen Applikationen derselbe. Eine Applikation soll nicht nur gestartet und beendet werden können, man soll auch zwischen verschiedenen Applikationen wechseln können. Dies wird über einen Pause-Zustand der Applikation erreicht. Beim Wechsel von einer Applikation zur anderen wird die aktive Applikation in einen Pausezustand versetzt, damit eine andere gestartet oder aus der Pause aktiviert werden kann.

Aus dem Lebenszyklus einer mobilen Applikation ergeben sich bei den Anforderungen zu berücksichtigende Aspekte hinsichtlich der Ressourcenbelegung, der Datenpersistenz, der Aktualität visualisierter Daten und der Verfügbarkeit des HMI auf dem Endgerät. (HMI: „Human-Machine Interface“, steht hier für den Bildschirm mit der grafischen Benutzeroberfläche als Schnittstelle zur Interaktion zwischen Benutzer und Anwendung auf dem mobilen Endgerät).

Für mobile Geschäftsanwendungen sollte es möglich sein, den Pause-Zustand überspringen zu können. Somit können verschiedene Geschäftsanwendungen parallel laufen. Dies ist eine weitere wichtige Anforderung an das System.

### 9.3.1 Verfügbarkeit des eigenen HMI „Human-Machine Interface“

Eine Applikation kann z. B. durch eingehende Anrufe in ihrer Ausführung unterbrochen werden, so dass das HMI zur Visualisierung und zur Steuerung der Applikation temporär nicht verfügbar ist. Die Applikation wird in den Pause-Zustand versetzt

**Tabelle 9.2** Abgeleitete Anforderungen im Anwendungssystem

id	Name	Anforderung	Systemkontext	Referenz
Req-1	Lebenszyklus	Wenn eine Komponente des Anwendungssystems auf einem Smartphone ausgeführt wird, muss dieses den Lebenszyklus mobiler Applikationen implementieren.	Business-App	9.3
Req-1.1	Lebenszyklus „paused“	Die mobile Applikation muss den Wechsel in den Zustand „paused“ implementieren.	Business-App	9.3
Req-1.2	Lebenszyklus „active“	Die mobile Applikation muss den Wechsel in den Zustand „active“ implementieren.	Business-App	9.3
Req-1.3	Lebenszyklus „destroyed“	Die mobile Applikation muss den Wechsel in den Zustand „destroyed“ implementieren.	Business-App	9.3

und das „Application Management System“ des Endgeräts stellt das HMI der nun aktiven Applikation zur Verfügung. Nach der Unterbrechung soll die pausierende Applikation wieder wie gewohnt ausgeführt werden können. Aus Sicht der Applikation hat man auf die Dauer der Unterbrechung keinen Einfluss und weiß nicht, ob die Anwendung aus dem Pausenzustand wieder aktiviert oder beendet wird. Dadurch sollte beachtet werden, dass die visualisierten Daten nach der Unterbrechung möglicherweise nicht mehr aktuell sind. Beim Beenden einer Applikation werden die Ressourcen wieder freigegeben, die sie zur Laufzeit auf dem Endgerät belegt. Bezogen auf unser Praxisbeispiel „das Befüllen von vernetzten Getränkeautomaten“ aus der Einleitung stellt dieser Fall nicht unbedingt eine Ausnahmesituation dar: Der Außendienstmitarbeiter befüllt gerade Waren an einem Getränkeautomaten. Der Geschäftsprozess sieht vor, dass er die Mengen der aufgefüllten Ware über die Business-App auf seinem Smartphone eingibt. So gibt er die Mengen in die Applikation ein, bis er durch einen eingehenden Anruf unterbrochen wird. Der Anruf ist wichtig, er nimmt ihn entgegen. Das System auf seinem Smartphone versetzt die laufende Business-App in den Zustand „paused“. Der nun telefonierende Außendienstmitarbeiter hat seine Dateneingabe noch nicht beendet. Nach dem Beenden seines Telefonats ruft er wieder seine pausierende Business-App auf, wodurch das System auf seinem Endgerät den Zustandswechsel „paused“ zu „active“ veranlasst. Der Außendienstmitarbeiter wird sich ärgern, wenn er alle Daten erneut eingeben muss. Da solche Szenarien unter Umständen zum Verlust bereits eingegebener Daten durch den Anwender führen können, sollten sie in den Anforderungen an Business-Apps berücksichtigt werden. Beim Beenden der Applikation kann abhängig vom Endgerät die Zeit variieren, die einer Applikation zum Speichern von Daten und für Aufräumarbeiten (wie das Schließen geöffneter Verbindungen) zur Verfügung steht.

Es ist davon auszugehen, dass sich die Nichtbeachtung dieser Aspekte negativ auf die Zufriedenheit der Anwender der entwickelten Business-Apps auswirken wird.

**Tabelle 9.3** Abgeleitete Anforderungen im Anwendungssystem (HMI-Verfügbarkeit)

id	Name	Anforderung	Systemkontext	Referenz
Req-1.1.2	Lebenszyklus GUI	Wenn die Applikation den Zustand von „active“ zu „paused“ wechselt, entzieht das Endgerät dem Benutzer die GUI der mobilen Applikation. Der vom System abgebildete Geschäftsprozess muss berücksichtigen, dass ein Benutzer, der z. B. einen eingehenden Anruf annimmt, temporär die Applikation nicht bedienen kann.	Business-App	9.3.1
Req-1.2.1	Lebenszyklus „active“ Daten	Wenn die Applikation den Zustand von „paused“ zu „active“ wechselt, muss das System dem Benutzer aktuelle Daten visualisieren.	Business-App	9.3.1
Req-1.2.2	Lebenszyklus „active“ Datenänderung	Wenn ein Benutzer Daten in der Business-App verändert (z. B. Zeicheneingabe in leere Felder der Applikation) und der Zustand von „active“ zu „paused“ wechselt, muss diese Veränderung bei dem nächsten Zustandswechsel „paused“ zu „active“ in der Applikation noch vorhanden sein.	Business-App	9.3.1
Req-1.3.1	Lebenszyklus Datenpersistenz	Wenn die Applikation den Zustand von „active“ zu „destroyed“ oder von „paused“ zu „destroyed“ wechselt, muss das System die Daten, die der Benutzer eingegeben hat, persistent speichern.	Business-App	9.3.1
Req-1.3.2	Lebenszyklus Ressourcen	Wenn die Applikation in den Zustand „destroyed“ wechselt, muss das System die verwendeten Ressourcen (wie geöffnete Verbindungen) innerhalb der Zeit sauber freigeben, die die Zielplattform auf dem Endgerät zur Verfügung stellt.	Business-App	9.3.1

### 9.3.2 Verfügbarkeit der Gegenstelle

Eine Business-App auf einem mobilen Endgerät wird in den meisten Fällen mit Servern kommunizieren, die wiederum mit anderen Systemen kommunizieren und ebenfalls an dem unterstützten Geschäftsprozess beteiligt sind. Diese Systeme können auf Servern oder wiederum als Business-App auf anderen mobilen Endgeräten laufen. Anwendungen auf einem Server sind in der Regel gut verfügbar. Von einer ständigen Verfügbarkeit einer App als mobile Gegenstelle kann jedoch nicht ausgegangen werden. Nehmen wir an, in einem bestimmten Szenario eines Geschäftsprozesses stößt ein Benutzer in seiner Business-App eine Aktion an, die eine Reaktion einer auf einem Gerät eines anderen Benutzers laufenden Applikation erwartet. Die Gegenstelle könnte gerade ausgeschaltet sein, der Akku des Gerätes leer sein, oder

der Benutzer befindet sich einfach gerade in einem Telefongespräch oder nutzt eine andere Applikation auf seinem Smartphone. Der angestoßene und unbeantwortete Vorgang kann nach einer Zeitüberschreitung abgebrochen werden, wodurch die Aktion fehlschlägt. Die Aktion kann in einer Warteschlange mit einem längeren Timeout gehalten werden um der Gegenstelle weitere Chancen zur Reaktion zu geben. Möglicherweise gibt es auch alternative Gegenstellen, die gerade zu einer sofortigen Reaktion bereit wären.

In unserem Praxisbeispiel „das Befüllen von vernetzten Getränkeautomaten“ stellt der Mitarbeiter im Lager fest, dass durch eine eingegangene Bestellung des Systems eines Getränkeautomaten die Auslieferung der Ware optimiert werden kann. Das System im Lager zeigt dem Mitarbeiter an, welcher Außendienstmitarbeiter für eine Änderung seines Auslieferungsauftrags infrage kommt. Ein Außendienstmitarbeiter hat kürzlich die abgeschlossene Befüllung einer benachbarten Station über sein Smartphone bestätigt. Das Anwendungssystem im Lager zeigt an, dass dieser Auslieferer auch noch genügend Ware mitführt, um die neu eingetroffene Bestellung auf seiner Route berücksichtigen zu können. Der Lagermitarbeiter veranlasst in seinem System die Routenänderung dieses Auslieferers, doch das System bekommt keine positive Rückmeldung von der mobilen Gegenstelle. Kann dieser die Routenänderung gerade nicht bestätigen, weil er gerade telefoniert und die grafische Oberfläche seiner Business-App auf seinem Endgerät momentan eben nicht verfügbar ist? Das System des Lagermitarbeiters berücksichtigt diesen Fall und arbeitet weiter. Es registriert eine Veränderung der GPS-Koordinaten eines weiteren Außendienstmitarbeiters in der Nähe, auch bei ihm ist noch genügend Ware vorhanden. Das System schlägt dem Lagermitarbeiter nun diesen Auslieferer für die Routenänderung vor; durch die registrierte Änderung der GPS-Koordinaten geht das System davon aus, dass diese Gegenstelle gerade verfügbar ist. Der Lagermitarbeiter veranlasst die Routenänderung, die nun auch von der verfügbaren mobilen Gegenstelle bestätigt wird.

Die Art und Weise, wie das Anwendungssystem sich unter diesen Umständen verhalten soll, hängt von verschiedenen Faktoren ab und muss bei den Anforderungen beachtet werden.

### **9.3.3 Verfügbarkeit des Netzes**

Für die mobilen Geschäftsanwendungen ist die Verfügbarkeit des Netzes sehr wichtig. Obwohl die Entwicklung der mobilen Geschäftsanwendungen von Netzanbietern unabhängig ist, brauchen die Unternehmen für den Einsatz ihrer Geschäftsanwendungen häufig einen Netzanbieter, der flächendeckend die Verfügbarkeit des Netzes garantieren kann. Der große Ärger ist die Nichterreichbarkeit der Gegenstelle, wenn sich das entsprechende Endgerät außerhalb der Reichweite des Netzwerks oder Mobilfunknetzes befindet.

Dies kann auch das eigene Endgerät betreffen, von dem aus man gerade selbst eine Business-App bedient, womit Daten von Gegenstellen ebenfalls weder ange-

**Tabelle 9.4** Abgeleitete Anforderungen im Anwendungssystem (Gegenstellenverfügbarkeit)

id	Name	Anforderung	Systemkontext	Referenz
Req-2	Gegenstelle	Das System muss berücksichtigen, dass die Verbindung zu einer mobilen Gegenstelle jederzeit gestört sein kann.	Anwendungssystem	9.3.2
Req-2.1	Gegenstelle Timeout	Wenn eine Aktion einer Komponente des Anwendungssystems eine Reaktion von einer mobilen Gegenstelle erwartet, muss die Aktion nach einem Timeout abgebrochen werden, damit dieses System für weitere Aktionen wieder verfügbar ist.	Anwendungssystem	9.3.2
Req-2.1.1	Gegenstelle Timeout 2	Das System kann vor dem Abbrechen der unbeantworteten Aktion in einen weiteren Timeout laufen, um der Gegenstelle eine weitere Chance zur Reaktion zu geben.	Anwendungssystem	9.3.2
Req-2.1.2	Alternative Gegenstelle	Das System kann nach alternativen verfügbaren Gegenstellen suchen.	Anwendungssystem	9.3.2

fordert noch empfangen werden können. Aus Anwendersicht ist es ärgerlich, die veränderten oder neu eingegebenen Daten beim Verlassen der Applikation zu verlieren. Sie könnten zunächst lokal auf dem Endgerät des Anwenders gespeichert werden, um zu einem späteren Zeitpunkt bei verfügbarem Netz gesendet zu werden. Doch was ist, wenn zwischenzeitlich die Daten bereits zentral von einem anderen Anwender bearbeitet wurden?

Bei mobilen Endgeräten ist die Gefahr, dass die vorgesehene Verfügbarkeit aller Gegenstellen gestört ist, wesentlich größer als bei verteilten Anwendungen in einem statischen Netzwerk. So scheint es angemessen, bei der Entwicklung von Business-Apps der Datensynchronisation entsprechende Beachtung zu schenken.

## 9.4 Besonderheit von mobilen Apps im Kontext des Plattformsystems

Die Art und Weise, wie eine Applikation auf Endgeräten verteilt wird, ist von den Mechanismen abhängig, die die verschiedenen Plattformhersteller zur Installation von Applikationen vorgesehen haben. Stark zentralisierte Ansätze, bei denen die Plattform die Installationsquelle von Applikationen strikt vorschreibt, erschweren es den Unternehmen, ihre betriebsinternen Applikationen auf den Endgeräten zu verteilen. Diesem Umstand entgegnet Apple mit dem „Apple Enterprise Developer Account“. Hiermit kann ein Unternehmen eine geschlossene Abteilung innerhalb

**Tabelle 9.5** Abgeleitete Anforderungen im Anwendungssystem (Netzverfügbarkeit)

id	Name	Anforderung	Systemkontext	Referenz
Req-3	Datenverlust	Wenn ein Benutzer mit einer Business-App Daten zentral im Anwendungssystem speichert und die Verbindung zum Anwendungssystem gestört ist, muss die Business-App eine Möglichkeit bieten, den Verlust dieser Daten aus Verbindungsman gel zu verhindern.	Business-App	9.3.3
Req-3.1	Datenverlust lokal	Das System kann die Daten zunächst lokal auf dem mobilen Endgerät des Benutzers speichern.	Business-App	9.3.3
Req-3.1.1	Datenverlust zentral	Das System kann die temporären Daten zentral speichern, wenn die Verbindung zum Anwendungssystem wieder besteht.	Business-App	9.3.3
Req-3.1.1.1	Datenverlust Synchronisation	Werden temporär gespeicherte Daten von der Business-App eines mobilen Endgeräts zeitversetzt zentral im Anwendungssystem gespeichert, muss das Anwendungssystem die Daten so synchronisieren, dass zwischenzeitlich zentral aktualisierte Daten durch den Speichervorgang nicht überschrieben werden.	Anwendungssystem	9.3.3

des App Stores kaufen, um Business-Apps für einen geschlossenen Benutzerkreis zugänglich zu machen. Microsoft stellt momentan nur den allgemein zugänglichen „Market Place“ ohne geschlossene Benutzergruppen bereit. Bei Android gestaltet sich die Verteilung von Applikationen flexibler, da sie nicht zentralisiert ist und von jeder beliebigen Quelle wie einem unternehmensinternen Server ermöglicht wird. Bei BlackBerry wird der Bezugskanal von Applikationen „BlackBerry App World“ genannt. Für den Einsatz von BlackBerrys im Unternehmensumfeld bietet der „BlackBerry Enterprise Server“ (BES) der Unternehmens-IT die Möglichkeit, die Softwareinstallationen auf den einzelnen Endgeräten zu steuern. Die Applikationen können automatisch „over the air“ (OTA) installiert und deinstalliert werden – benutzerspezifisch, und ohne dass der Benutzer darauf Einfluss nehmen kann. Die Installation unternehmensfremder Apps kann dem Benutzer des Endgeräts so auch gänzlich entzogen werden.

Aus Sicht der Unternehmens-IT wird man kontrollieren wollen, welche Sicherheitseinstellungen auf den Endgeräten der Mitarbeiter eingehalten werden und welche Apps in welcher Version dort installiert sind. Dies ist über sogenannte „Mobile Device Management“-Software möglich (z. B. „Aeria“ von Sybase oder

„Tarmac“ von Equinux). Über Profile werden die Konfigurationen der Endgeräte sowie die Installationen von Apps zentral verwaltet und OTA bereitgestellt. BlackBerry gilt im Unternehmensumfeld in Verbindung mit dem BES als Musterknabe. Das BlackBerry verbindet sich über eine verschlüsselte Verbindung mit dem BES, wo die Unternehmens-IT über so genannte Policies und Gruppen die erwünschte Konfiguration der mobilen Endgeräte detailliert festlegen kann. Manche Hersteller von Android-Smartphones (Motorola, HTC, Pantech, Sharp, Sony/Ericsson) arbeiten auch zusammen an einer gemeinsamen Mobile-Device-Management-Software (vgl. [7]).

Eine andere Lösung stellt der Virtualisierungsansatz dar, den VMware verfolgt, damit Unternehmen gesamte virtuelle Umgebungen auf den einzelnen Endgeräten der Mitarbeiter kontrollieren können (vgl. [8]). Die Business-Apps könnten so in einem sicheren, vom Unternehmen konfigurierbaren Kontext laufen, den der Benutzer des Endgeräts verlassen kann, um für nichtberufliche Zwecke selbst die Konfiguration seines Endgeräts zu übernehmen.

Aus der Art und Weise, wie ein Unternehmen seinen Mitarbeitern bestimmte interne Unternehmensdaten auf mobilen Endgeräten zur Verfügung stellen möchte und inwieweit die Konfiguration der Endgeräte sowie die App-Installationen von der IT kontrolliert werden wollen, entstehen teilweise sicherheitskritische Anforderungen, die nicht von jeder Plattform im gleichen Maße unterstützt werden. So kann die Berücksichtigung dieser Aspekte im Systemkontext von Business-Apps eine Entscheidung für oder gegen eine bestimmte Plattform bedeuten.

#### 9.4.1 Plattformunabhängigkeit

Das Thema der Plattformunabhängigkeit wurde in diesem Kapitel bereits im Abschnitt „Kontextgrenze“ angesprochen. Laut einer Studie „Developer Economics 2011“ [9] tendieren Entwickler von mobilen Applikationen dazu, sich von der Symbian- sowie Java-ME-Plattform abzuwenden, offenbar werden Apples iOS sowie Googles Android von ihnen immer stärker bevorzugt.

An dritter Stelle folgt das „mobile Web“, gemeint sind mit HTML und JavaScript realisierte, plattformübergreifende Webanwendungen [10].

Zu dem Trend der Entwickler hin zum mobilen Web nennt die aktuelle Studie von VisionMobile vier Gründe (vgl. [11]):

- Mobiles Web ist die erste Wahl für die plattformübergreifende Entwicklung, um über iPhone- und Android-Endgeräten hinaus vielen Endgeräten gerecht zu werden.
- Webentwickler, die sich mit HTML und JavaScript auskennen, reizt es auch, für mobile Endgeräte zu entwickeln, zumal sie gewohnt sind, verschiedene Eigenarten und Formate diverser Zielgeräte zu berücksichtigen, und somit auf den Multi-Plattform-Aspekt tragbarer Endgeräte vorbereitet sind.

**Tabelle 9.6** Abgeleitete Anforderungen im Plattformsystem

id	Name	Anforderung	Systemkontext	Referenz
Req-5	Endgerät-Konfiguration	Die Unternehmens-IT muss sicherheitskritische Konfigurationen auf den mobilen Endgeräten der Anwender auf jeder Zielplattform kontrollieren können.	Plattformsystem	9.2.2
Req-5.1	Version und Sicherheit	Insbesondere soll die Unternehmens-IT die Installation und Aktualisierung von Business-Apps sowie die Einhaltung benötigter Sicherheitseinstellungen auf den mobilen Endgeräten kontrollieren können.	Plattformsystem	9.2.2, 9.4
Req-5.1.1	Konfiguration Kontext	Wenn die Benutzer von Business-Apps ihre Endgeräte auch privat nutzen, müssen die Business-Apps auf den Endgeräten innerhalb eines Kontexts laufen, den die Unternehmens-IT konfigurieren kann.	Plattformsystem	9.2.2, 9.4
Req-5.1.1.1	Konfiguration virtueller Kontext	Die Unternehmens-IT kann möglicherweise durch Einsatz von Virtualisierungssoftware und „Mobile Device Management Software“ die Konfiguration im Business-App-Kontext auf den Endgeräten kontrollieren.	Plattformsystem	9.2.2, 9.4

- Unternehmen nutzen beim Ausbau ihrer digitalen Strategie hin zu mobilen Applikationen das Mobile Web als endgerätabgreifende kostengünstige Plattform mit hoher Reichweite. Analog wählt die Unternehmens-IT das Mobile Web als Standardplattform, um ihr Intranet auf mobile Endgeräte zu portieren.
- HTML/JavaScript-Entwickler werden von so genannten „HTML-to-native“-Entwicklungswerkzeugen unterstützt, in den Markt nativer Applikationen einzudringen. Mit Hilfe von „HTML-to-native“-Entwicklungswerkzeugen kann die Funktionalität einer Webanwendung in eine native Applikation umgesetzt werden – zumal native Applikationen Funktionalität als Webinhalt kapseln können, was plattformübergreifende Entwicklung erleichtert.

Hinsichtlich der Plattformunabhängigkeit scheinen neben dem bereits angesprochenen Virtualisierungsansatz bereits existierende Technologien des Mobile Web in Frage zu kommen, um bei der Entwicklung von Business-Apps den plattformunabhängigen Anforderungen gerecht werden zu können, ohne dass ein neues Plattformunabhängigkeitskonzept entwickelt werden muss.

**Tabelle 9.7** Abgeleitete Anforderungen Plattformunabhängigkeit

id	Name	Anforderung	Systemkontext	Referenz
Req-6	Plattform-unabhängigkeit Grundfunktion	Wenn eine Business-App plattformübergreifend eingesetzt wird, muss ihre Grundfunktionalität auf den mobilen Endgeräten aller unterstützten Zielplattformen laufen.	Plattformsystem	9.2.2
Req-6.1	Plattform-unabhängigkeit virtuell	Möglicherweise kann der Einsatz von Virtualisierungssoftware einen plattformübergreifenden Einsatz der Business-App fördern.	Plattformsystem	9.4.1
Req-6.2	Plattform-unabhängigkeit Mobile Web	Möglicherweise kann der Einsatz von Technologien des Mobile Web einen plattformübergreifenden Einsatz der Business-App fördern.	Plattformsystem	9.4.1

## 9.5 Zusammenfassung

Mobile Applikationen auf modernen Mobiltelefonen und Smartphones sind in der Lage Daten zu visualisieren, zu übermitteln und zu empfangen. So können aufeinander abgestimmte Komponenten genutzt werden um Geschäftsprozesse zu unterstützen und Geschäftsanwendungen mobil abzubilden. Für ihre Nutzung wird somit kein vollwertiger Computer benötigt.

Zur Untersuchung wurden hauptsächlich Anforderungen von Mitarbeitern, Endkunden und Maschinen, die als Anwender von mobilen Geschäftsanwendungen in Erscheinung treten, berücksichtigt sowie auf die Bedürfnisse der Unternehmens-IT beim Einbinden der Geschäftsanwendung eingegangen. Es ergeben sich Anforderungen an die auf den einzelnen Endgeräten laufenden Applikationen, Anforderungen an die daraus resultierende Geschäftsanwendung als Ganzes und Anforderungen an die Plattform, auf der diese ausgeführt werden. Eine tabellarische Gesamtliste der exemplarisch abgeleiteten Anforderungen mit Referenz auf die entsprechenden Abschnitte befindet sich in der Tabelle 9.8.

Die Komplexität der mobilen Geschäftsanwendungen als System wurde anhand eines Praxisbeispiels („das Befüllen von vernetzten Getränkeautomaten“) erläutert. Dabei ist festzustellen, dass sowohl die Systemgrenze als auch die Kontextgrenze die Hersteller der mobilen Geschäftsanwendungen vor eine große Herausforderung stellt. Ein Unternehmen als Anwender einer mobilen Geschäftsanwendung hat die Wahl zwischen Plattformsystem und Anwendungssystem.

Mit einem Plattformsystem wie z. B. die BlackBerry-Plattform mit der „Mobile Device Management“-Software ist die Integration der mobilen Geschäftsanwendung in die Unternehmens-Infrastruktur kontrollierbar und führt zu einem guten Ergebnis. Es bleibt trotzdem offen, ob der Geschäftsprozess des Unternehmens dadurch unterstützt wird. Das Unternehmen hat sicherlich eine integrierbare Lösung, aber die Plattformunabhängigkeit geht verloren.

**Tabelle 9.8** Abgeleitete Anforderungen Gesamtliste

id	Name	Anforderung	Systemkontext	Referenz
Req-1	Lebenszyklus	Wenn eine Komponente des Anwendungssystems auf einem Smartphone ausgeführt wird, muss dieses den Lebenszyklus mobiler Applikationen implementieren.	Business-App	9.3
Req-1.1	Lebenszyklus „paused“	Die mobile Applikation muss den Wechsel in den Zustand „paused“ implementieren.	Business-App	9.3
Req-1.1.2	Lebenszyklus GUI	Wenn die Applikation den Zustand von „active“ zu „paused“ wechselt, entzieht das Endgerät dem Benutzer die GUI der mobilen Applikation. Der vom System abgebildete Geschäftsprozess muss berücksichtigen, dass ein Benutzer, der z. B. einen eingehenden Anruf annimmt, temporär die Applikation nicht bedienen kann.	Business-App	9.3.1
Req-1.2	Lebenszyklus „active“	Die mobile Applikation muss den Wechsel in den Zustand „active“ implementieren.	Business-App	9.3
Req-1.2.1	Lebenszyklus „active“ Daten	Wenn die Applikation den Zustand von „paused“ zu „active“ wechselt, muss das System dem Benutzer aktuelle Daten visualisieren.	Business-App	9.3.1
Req-1.2.2	Lebenszyklus „active“ Datenänderung	Wenn ein Benutzer Daten in der Business-App verändert (z. B. Zeicheneingabe in leere Felder der Applikation) und der Zustand von „active“ zu „paused“ wechselt, muss diese Veränderung bei dem nächsten Zustandswechsel „paused“ zu „active“ in der Applikation noch vorhanden sein.	Business-App	9.3.1
Req-1.3	Lebenszyklus „destroyed“	Die mobile Applikation muss den Wechsel in den Zustand „destroyed“ implementieren.	Business-App	9.3
Req-1.3.1	Lebenszyklus Datenpersistenz	Wenn die Applikation den Zustand von „active“ zu „destroyed“ oder von „paused“ zu „destroyed“ wechselt, muss das System die Daten, die der Benutzer eingegeben hat, persistent speichern.	Business-App	9.3.1
Req-1.3.2	Lebenszyklus Ressourcen	Wenn die Applikation in den Zustand „destroyed“ wechselt, muss das System die verwendeten Ressourcen (wie geöffnete Verbindungen) innerhalb der Zeit sauber freigeben, die die Zielpлатform auf dem Endgerät zur Verfügung stellt.	Business-App	9.3.1
Req-2	Gegenstelle	Das System muss berücksichtigen, dass die Verbindung zu einer mobilen Gegenstelle jederzeit gestört sein kann.	Anwendungs- system	9.3.2

**Tabelle 9.8** (Fortsetzung)

id	Name	Anforderung	Systemkontext	Referenz
Req-2.1	Gegenstelle Timeout	Wenn eine Aktion einer Komponente des Anwendungssystems eine Reaktion von einer mobilen Gegenstelle erwartet, muss die Aktion nach einem Timeout abgebrochen werden, damit dieses System für weitere Aktionen wieder verfügbar ist.	Anwendungs- system	9.3.2
Req-2.1.1	Gegenstelle Timeout 2	Das System kann vor dem Abbrechen der unbeantworteten Aktion in einen weiteren Timeout laufen, um der Gegenstelle eine weitere Chance zur Reaktion zu geben.	Anwendungs- system	9.3.2
Req-2.1.2	Alternative Gegenstelle	Das System kann nach alternativen verfügbaren Gegenstellen suchen.	Anwendungs- system	9.3.2
Req-3	Datenverlust	Wenn ein Benutzer mit einer Business-App Daten zentral im Anwendungssystem speichert und die Verbindung zum Anwendungssystem gestört ist, muss die Business-App eine Möglichkeit bieten, den Verlust dieser Daten aus Verbindungsmaßnahmen zu verhindern.	Business-App	9.3.3
Req-3.1	Datenverlust lokal	Das System kann die Daten zunächst lokal auf dem mobilen Endgerät des Benutzers speichern.	Business-App	9.3.3
Req-3.1.1	Datenverlust zentral	Das System kann die temporären Daten zentral speichern, wenn die Verbindung zum Anwendungssystem wieder besteht.	Business-App	9.3.3
Req-3.1.1.1	Datenverlust-Synchronisation	Werden temporär gespeicherte Daten von der Business-App eines mobilen Endgeräts zeitversetzt zentral im Anwendungssystem gespeichert, muss das Anwendungssystem die Daten so synchronisieren, dass zwischenzeitlich zentral aktualisierte Daten durch den Speichervorgang nicht überschrieben werden.	Anwendungs- system	9.3.3
Req-4	App-Verteilung	Der Mechanismus zur Verteilung der Business-Apps auf mobile Endgeräte muss auf jeder Zielplattform die Anforderungen der Unternehmens-IT unterstützen.	Plattform- system	9.2.2
Req-4.1	App-Verteilung Gruppe	Die Unternehmens-IT muss auf jeder Zielplattform die Verteilung auf die vorgesehene geschlossene Benutzer- und Endgerätegruppe sicher beschränken können.	Plattform- system	9.2.2

**Tabelle 9.8** (Fortsetzung)

id	Name	Anforderung	Systemkontext	Referenz
Req-5	Endgerätekonfiguration	Die Unternehmens-IT muss sicherheitskritische Konfigurationen auf den mobilen Endgeräten der Anwender auf jeder Zielplattform kontrollieren können.	Plattformsystem	9.2.2
Req-5.1	Version und Sicherheit	Insbesondere soll die Unternehmens-IT die Installation und Aktualisierung von Business-Apps, sowie die Einhaltung benötigter Sicherheitseinstellungen auf den mobilen Endgeräten kontrollieren können.	Plattformsystem	9.2.2, 9.4
Req-5.1.1	Konfiguration Kontext	Wenn die Benutzer von Business- Apps ihre Endgeräte auch privat nutzen, müssen die Business-Apps auf den Endgeräten innerhalb eines Kontexts laufen, den die Unternehmens-IT konfigurieren kann.	Plattformsystem	9.2.2, 9.4
Req-5.1.1.1	Konfiguration virtueller Kontext	Die Unternehmens-IT kann möglicherweise durch Einsatz von Virtualisierungssoftware und „Mobile Device Management Software“ die Konfiguration im Business-App-Kontext auf den Endgeräten kontrollieren.	Plattformsystem	9.2.2, 9.4
Req-6	Plattform-unabhängigkeit Grundfunktion	Wenn eine Business-App plattformübergreifend eingesetzt wird, muss ihre Grundfunktionalität auf den mobilen Endgeräten aller unterstützten Zielplattformen laufen.	Plattformsystem	9.2.2
Req-6.1	Plattform-unabhängigkeit virtuell	Möglicherweise kann der Einsatz von Virtualisierungssoftware einen plattformübergreifenden Einsatz der Business-App fördern.	Plattformsystem	9.4.1
Req-6.2	Plattform-unabhängigkeit Mobile Web	Möglicherweise kann der Einsatz von Technologien des Mobile Web einen plattformübergreifenden Einsatz der Business-App fördern.	Plattformsystem	9.4.1

Mit der Wahl eines Anwendungssystems hat das Unternehmen zwar mehr Flexibilität (Entwicklung ist auf ein offenes Betriebssystem basiert), es hat aber bei der Integration der mobilen Geschäftsanwendung in seine IT-Infrastruktur mit einem großen Aufwand zu rechnen. Laut Ergebnissen einer aktuellen Studie von Vision-Mobile sollen die Plattformen Android, iOS sowie Windows Phone künftig mehr an Bedeutung gewinnen. Weiterhin dürfte auch der Virtualisierungsansatz interessante Perspektiven bieten um den Anforderung der Plattformunabhängigkeit neben den Technologien des Mobile Web gerecht zu werden.

Die große offene Frage bleibt weiterhin die Verfügbarkeit des Netzes, die den Einsatz der mobilen Geschäftsanwendungen unattraktiv macht. Hier sollten die Unternehmen sich für ihre mobile Geschäftsanwendung für die Netzanbieter mit besserer Netzabdeckung entscheiden.

## Literaturverzeichnis

1. Pohl K, Rupp C (2010) S 21 Definition 2-1: Systemkontext. In: Basiswissen Requirements Engineering, 2. Aufl dpunkt.verlag GmbH, Heidelberg
2. (2011) Laufzeitumgebung. Wikipedia. <http://de.wikipedia.org/wiki/Laufzeitumgebung>. Accessed 11 July 2011.
3. Pohl K, Rupp C (2010) S 24 Abb 2-2 Arten von Aspekten im Systemkontext. In: Basiswissen Requirements Engineering, 2. Aufl dpunkt.verlag GmbH, Heidelberg
4. Pohl K, Rupp C (2010) S 24 Definition 2-1: Systemgrenze. In: Basiswissen Requirements Engineering, 2. Aufl dpunkt.verlag GmbH, Heidelberg
5. Pohl K, Rupp C (2010) S 26 Definition 2-3 Kontextgrenze. In: Basiswissen Requirements Engineering, 2. Aufl dpunkt.verlag GmbH, Heidelberg
6. Ribo (2005) Lebenszyklus eines MIDlets. Wikipedia. <http://de.wikipedia.org/wiki/MIDlet>. Accessed 07 June 2011
7. Glahn K, Wirtgen J (2011) MWC Business-Funktionen für Android-Smartphones. [www.heise.de/mobil/meldung/Business-Funktionen-fuer-Android-Smartphones-1190094.html](http://www.heise.de/mobil/meldung/Business-Funktionen-fuer-Android-Smartphones-1190094.html). Accessed 09 June 2011
8. Junginger et al. (2011) S 56-70 Smartphone-Administration. In iX 04/2011, Heise Zeitschriften Verlag GmbH & Co. KG
9. VisionMobile (2011) Developer Economics 2011. [http://www.visionmobile.com/rsc/researchreports/VisionMobile-Developer\\_Economics\\_2011.pdf](http://www.visionmobile.com/rsc/researchreports/VisionMobile-Developer_Economics_2011.pdf). Accessed 10 June 2011
10. Lippert R (2011) App-Entwicklung heute und morgen. Heise mobil. <http://www.heise.de/mobil/meldung/App-Entwicklung-heute-und-morgen-1258282.html>. Accessed 10 June 2011
11. VisionMobile (2011) S 16-17 Winners and losers in the platform race. In: Developer Economics 2011. [http://www.visionmobile.com/rsc/researchreports/VisionMobile-Developer\\_Economics\\_2011.pdf](http://www.visionmobile.com/rsc/researchreports/VisionMobile-Developer_Economics_2011.pdf). Accessed 10 June 2011

---

# **Kapitel 10**

## **Der Browser als mobile Plattform der Zukunft – Die Möglichkeiten von HTML5-Apps**

### **Chancen und Grenzen der Entwicklung mobiler Anwendungen mit Hilfe von Webstandards**

Kristin Albert und Michael Stiller

**Zusammenfassung** Eine mobile Anwendung für die breite Masse an Endgeräten zu entwickeln ist eine zeit- und kostenintensive Arbeit. Denn soll die Anwendung auf den wichtigsten Plattformen vertreten sein, muss sie separat für die jeweilige Plattform entwickelt werden. Grund dafür sind die Unterschiede im Betriebssystem, der Programmiersprache, den Bibliotheken sowie dem Erscheinungsbild und Verhalten der Plattformen. Doch trotz aller Unterschiedlichkeit kann man durchaus eine Gemeinsamkeit bei den mobilen Plattformen entdecken. Alle Geräte bringen eine Software zum Darstellen von Webseiten, also einen Browser, mit. Heute ist der Browser von dem mobilen Endgerät nicht mehr wegzudenken und hat in letzter Zeit durch die großen Fortschritte im Standardisierungsprozess des World Wide Web und den damit verbundenen neuen Möglichkeiten zusätzlich an Bedeutung gewonnen. Dieser Beitrag gibt einen Überblick über die Neuerungen von HTML5, stellt JavaScript APIs für die Web-App-Entwicklung vor und beschreibt die Möglichkeiten zur Optimierung mobiler Anwendungen. Neben einem Überblick über den aktuellen Stand der HTML5-Implementierung durch die mobilen Browser werden die Grenzen mobiler Webentwicklung aufgezeigt. Am Ende wird ein Ausblick in die Zukunft der mobilen Anwendungen gegeben.

---

Kristin Albert  
Fraunhofer ESK, München,  
E-mail: kristin.albert@esk.fraunhofer.de

Michael Stiller  
Fraunhofer ESK, München,  
E-mail: michael.stiller@esk.fraunhofer.de

## 10.1 Der Browser als Plattform

Eine mobile Anwendung für die breite Masse an Endgeräten zu entwickeln ist eine zeit- und kostenintensive Arbeit. Denn soll die Anwendung auf den wichtigsten<sup>1</sup> Plattformen vertreten sein, muss sie separat für die jeweilige Plattform entwickelt werden. Grund dafür sind die Unterschiede im Betriebssystem, der Programmiersprache, den Bibliotheken sowie dem Erscheinungsbild und Verhalten der Plattformen. Selbst für den Vertrieb der fertigen mobilen Applikation gibt es plattformspezifische Besonderheiten.

Doch trotz aller Unterschiedlichkeit kann man durchaus eine Gemeinsamkeit bei den mobilen Plattformen entdecken. Alle Geräte bringen eine Software zum Darstellen von Webseiten, also einen Webbrowser, mit.

Doch das war nicht immer so. Erst die neuartigen Bedienkonzepte der Multi-Touch Smartphones und die Evolution<sup>2</sup> von GSM<sup>3</sup> verschafften dem mobilen Browser seinen Durchbruch.

Heute ist der Browser von dem mobilen Endgerät nicht mehr wegzudenken und hat sich durch die ambitionierte Arbeit der Browserhersteller zu einer ernstzunehmenden Entwicklungsplattform entwickelt. Die neuen Funktionen des Browsers ermöglichen das Rendern von Animationen, das Abspielen von Videos, den Zugriff auf Sensoren und das Speichern von Daten. Der mobile Browser kann heute viel mehr als nur Textinhalte darstellen. Er ähnelt zunehmend einer Unterhaltungsplattform. Spiele, Fotos, Dokumente, Karten, Musik und Videos können allein über den Browser ausgeführt und dargestellt werden.

Möglich wird dies durch eine Kombination verschiedener Sprachen und Technologien, die heute im Web eingesetzt werden. Dazu gehören neben HTML, JavaScript und den Cascading Stylesheets auch die bereitgestellten Browser APIs<sup>4</sup>.

Einmal implementiert laufen die Webanwendungen, dank der Standardisierung von HTML, auf jedem Endgerät, das über einen Webbrowser verfügt.

## 10.2 Die Entwicklung von HTML5

Ursprünglich wurde die Hypertext-Markup-Sprache<sup>5</sup> (HTML) entwickelt, um Informationen einfach und strukturiert auf digitalem Weg zu übertragen. Heute umfassen die eingesetzten Webtechniken jedoch viel mehr als eine Markup-Sprache, die den Inhalt eines Dokumentenformates beschreibt.

---

<sup>1</sup> Dazu zählen iOS, Android, BlackBerry und Windows Phone.

<sup>2</sup> Die Evolution von GSM bezeichnet die Erweiterung des GSM-Netzes um eine paketorientierte Datenübertragung.

<sup>3</sup> Global System for Mobile Communications.

<sup>4</sup> API bezeichnet eine Programmierschnittstelle, die Funktion für andere Softwaresysteme bereitstellt.

<sup>5</sup> Auszeichnungssprache.

Für die Weiterentwicklung und Standardisierung des World Wide Web setzt sich seit 1994 das W3C<sup>6</sup> ein. Regelmäßig werden Empfehlungen für die Weiterentwicklung von HTML und den angrenzenden Webtechniken an die Browserhersteller herausgegeben.

Neben dem W3C gründete sich 2004 die Arbeitsgruppe WHATWG<sup>7</sup>, bestehend aus Mitgliedern der Browserentwickler Mozilla Foundation, Opera Software ASA und Apple Inc., um die HTML-Spezifikation weiter voranzutreiben. Aufbauend auf die Spezifikation von HTML4 veröffentlichte die Arbeitsgruppe ihre Spezifikation unter dem Namen „Web Applications 1.0“, die später unter dem Namen HTML5 bekannt wurde. Nachdem die beiden Gruppen zwei weitere Jahre getrennt voneinander an ihren Entwürfen gearbeitet hatten, entschieden sich die Mitglieder des W3C, die Ideen der WHATWG für die Weiterentwicklung von HTML aufzunehmen. Seitdem arbeiten W3C und WHATWG gemeinsam an der Spezifikation, die von der WHATWG auch als „Living Standard“ bezeichnet wird. Die Bezeichnung soll verdeutlichen, dass es sich hier um eine kontinuierliche Weiterentwicklung der Spezifikation handelt.

Der Begriff HTML5 umfasst heute eine ganze Reihe an clientseitigen Webtechniken. So kann man in HTML5 neben Spezifikationen zum HTML-Markup auch JavaScript-APIs sowie DOM<sup>8</sup>- und CSS-Definitionen finden. Die aktuelle Spezifikation zu HTML5 findet man im W3C-Dokument „HTML5 a vocabulary and associated APIs for HTML and XHTML“<sup>9</sup>. Das Pendant dazu von der WHATWG enthält zusätzlich noch die Themen Microdata, Canvas und Communication und wurde unter „HTML – Living Standard“<sup>10</sup> veröffentlicht.

## 10.3 Was ist eine Mobile Web App?

Vielleicht haben Sie sich auch schon gefragt, warum mobile Webseiten häufig als Mobile Web App bezeichnet werden. Der Begriff Web Application bezeichnet eine oder mehrere Webseiten (XHTML + CSS), die mit server- und clientseitiger Logik ausgeführt werden. Damit wird ein ähnliches Aussehen und Verhalten wie bei einer nativen<sup>11</sup> Anwendung erreicht, nur das die Web App im Browser ausgeführt wird. Web Apps grenzen sich von einer einfachen mobilen Webseite durch höhere Komplexität sowie die Möglichkeiten der Interaktion ab. Dennoch kann jede Web App auch eine für die mobile Nutzung optimierte Webseite sein. Die Grenzen sind dabei oft fließend. In diesem Beitrag wird der Begriff Web App allgemein für eine im Browser ausgeführte Anwendung verwendet. Das Erscheinungsbild der Web

<sup>6</sup> World Wide Web Consortium.

<sup>7</sup> Web Hypertext Application Technology Working Group.

<sup>8</sup> Das Document Object Model beschreibt die Struktur und den Aufbau eines HTML-Dokumentes.

<sup>9</sup> <http://dev.w3.org/html5/spec/Overview.html> (4. Juni 2011).

<sup>10</sup> <http://www.whatwg.org/specs/web-apps/current-work/multipage/> (9. Juni 2011).

<sup>11</sup> Eine Anwendung, die nur speziell für eine bestimmte Endgeräte-Plattform entwickelt wurde.

App kann zum Beispiel einer einfachen mobilen Webseite ähneln, aber auch einer nativen App sehr nahekommen. Die verwendeten Technologien bleiben die gleichen.

## 10.4 Die Stärken von Web Apps

Es gibt viele Gründe, die für eine Entwicklung von Web Apps sprechen. Sie werden mit Hilfe aktueller Webtechnologien entwickelt, haben einen schnellen Entwicklungszyklus und die Anwendung oder Teile davon werden aus dem Web geladen. Das heißt, sie werden nicht auf dem Endgerät installiert, sondern nur im Browser des Endgerätes angezeigt. Vorteil davon ist, dass der Inhalt der Web App ohne große Schwierigkeiten jederzeit aktualisiert werden kann.

Außerdem machen sich Web Apps die drei großen Stärken von HTML zunutze:

### Offenheit

Getrieben von dem W3C und der WHATWG kann man von einer ständigen Weiterentwicklung des Standards ausgehen. Zusätzlich besteht die Möglichkeit aktiv an der Entwicklung teilzunehmen, indem Ideen und Verbesserungsvorschläge über die Mailingliste der WHATWG kommuniziert werden.

### Plattformunabhängigkeit

HTML ermöglicht eine plattformunabhängige Anwendungsentwicklung, da die Webanwendungen in jedem Browser und somit auf verschiedenen Plattformen und Geräten (z. B. Smartphone, Tablet PC, eBook Reader, TV) zum Einsatz kommen können.

### Flexibilität

HTML folgt einer strengen Trennung von Inhalt, Logik und Design. Das HTML-Markup beschreibt den Inhalt, JavaScript kapselt die Logik und CSS beschreibt die eigentliche Darstellung der Inhalte. Durch diese Trennung können große Teile der Anwendung auf verschiedenen Geräten wiederverwendet werden. Nur die Darstellung muss aufgrund unterschiedlicher Bildschirmgrößen für jedes Gerät angepasst oder optimiert werden.

## 10.5 Die Neuerungen von HTML5

Die Überarbeitung des HTML-Standards bringt eine Vielzahl an Neuerungen für die Entwicklung von Webseiten mit. Einige Verbesserungen kommen auch der Entwicklung von mobilen Webseiten zugute.

Mit HTML5 wurden zum Beispiel neue Elementtypen für die Strukturierung von Inhalten einer Webseite eingeführt. Die Verwendung der neuen Elemente vereinfacht den Aufbau und die Gestaltung der Seiten mit CSS.

Weiter wurden Audio- und Videotags eingeführt, die es erlauben, Audio- und Videodateien bei Verfügbarkeit des Codecs abzuspielen. Eine zusätzliche Installation von Browser Plug-Ins ist nicht mehr nötig.

Hinzu kommen außerdem eine Reihe neuer Input-Typen, die vor allem Vorteile auf dem mobilen Endgerät bieten. Da die meisten Geräte über einen Touchscreen verfügen und somit eine virtuelle Tastatur besitzen, kann sich die Tastatur je nach Input-Typ anpassen, was wiederum die Eingabe von Text für den Nutzer erleichtert. So kann für die Eingabe von Telefonnummern zum Beispiel ein Tastaturlayout bestehend aus Zahlen angezeigt werden.

Weiter wurden im Rahmen von HTML5 einige smartphonespezifische JavaScript-Funktionen standardisiert, die heute bereits von den meisten Browserherstellern implementiert wurden. Über diese Funktionen lassen sich zum Beispiel die verfügbaren Netzverbindungen (WiFi, 3G) abfragen, Touch- und Gesten-Events steuern sowie die Ausrichtung des Bildschirms bestimmen.

HTML5 unterstützt außerdem die Verwendung zusätzlicher Javascript APIs, wie zum Beispiel die Device API, die den Zugriff auf weitere Hardwarefunktionen des Gerätes ermöglichen soll. In Zukunft soll es auch die Möglichkeit geben, auf die Kamera und das Mikrofon des Endgerätes über eine API zuzugreifen.

Im Folgenden werden noch einige weitere JavaScript APIs im Detail vorgestellt.

### 10.5.1 Ortskoordinaten auslesen und verarbeiten

Über die Geolocation API stellt der Browser Funktionalitäten zur Standortbestimmung bereit. Der Standort kann mit dem integrierten GPS-Chip oder über das Internet bestimmt werden. Die Bestimmung über den GPS-Chip ist präziser als die Netzwerkbestimmung, dauert dafür aber erheblich länger. Da so gut wie alle modernen Smartphones einen solchen GPS-Chip besitzen, kann dieser zur Lokalisierung verwendet werden.

Für die Ortungsmethode mit GPS-Chip ist keine Internetverbindung nötig und deshalb kann sie in Kombination mit einer Offline-Applikation verwendet werden. Die API bietet nicht nur eine Funktion für das einmalige Auslesen des Standortes, sondern auch die Möglichkeit eine selbst definierte Callback-Methode aufzurufen, sobald sich der Standort verändert hat.

### ***10.5.2 2D Grafiken zeichnen und verändern***

Mit dem HTML-Element Canvas können Bitmap-Grafiken mit Hilfe von JavaScript dynamisch in die Webseite gerendert werden. Das Element stammt aus der Webkit-Implementierung und wurde später in die HTML5-Spezifikation aufgenommen. Für das Zeichnen von geometrischen Formen, Pfaden, Transparenz und Text oder für die Manipulation von Grafiken stellt die Canvas API der HTML5-Spezifikation zahlreiche Funktionen bereit. Ein Plug-In zur Darstellung des Inhalts im Browser wird nicht benötigt.

Bereits vor dem Canvas-Element bestand die Möglichkeit, dynamische Grafiken mit Hilfe von SVG-Grafiken im Browser anzuzeigen. SVG ist ein XML-basiertes Dateiformat, welches zweidimensionale Vektorgrafiken im Browser darstellen kann. Während Canvas mit einem HTML-Element eine Art Leinwand aufspannt, in die mit JavaScript-Funktionen gezeichnet werden kann, bestehen SVG-Grafiken aus einem Baum an Objekten innerhalb des HTML-Markups.

Dadurch können mit SVG zum Beispiel Maus-Events direkt an Elemente des DOM gebunden werden, was bei Grafiken mit hoher Interaktivität von Vorteil ist.

Weniger gut eignen sich SVG-Grafiken für komplexe Animationen weil, das DOM zu langsam für die schnellen Änderungen ist. Das Canvas-Element hat genau in diesem Punkt seinen großen Vorteil und eignet sich deshalb besonders für Spiele und Animationen, während das SVG-Format bei interaktiven und vektorbasierten Grafiken zum Einsatz kommen sollte.

### ***10.5.3 Daten lokal im Browser speichern***

Die Web Storage API bietet mit den Objekten localStorage und sessionStorage zwei Konzepte zur clientseitigen Speicherung von Daten. Damit stellt sie eine Alternative zu dem Konzept der Datenspeicherung mit Hilfe von Cookies bereit.

Die Daten werden beim Web Storage als Namen-Wert-Paare oder in einer SQL-Datenbank im Browser des Clients gesichert und können auf Wunsch temporär, also für die Dauer einer Browsersitzung (sessionStorage) oder dauerhaft (localStorage) abgelegt werden. Vorteile dieser Konzepte gegenüber von Cookies sind zum einen ihr einfacher Zugriff über ihren Namen oder eine SQL-Abfrage, die Sichtbarkeit der gespeicherten Informationen über mehrere Browserfenster hinweg sowie ein größerer Speicherplatz zum Ablegen der Daten.

Zum anderen können Web-Storage-Daten nur von clientseitigen Skripten verändert werden. Die Daten werden somit, wie zum Beispiel bei Cookies der Fall, nicht bei jeder Anfrage erneut an den Server gesendet. Zusammengefasst ist Web Storage also effizienter und sicherer als Cookies.

### 10.5.4 Daten offline verfügbar machen

Neben den bereits vorgestellten Konzepten zur lokalen Speicherung dynamischer Inhalte gibt es außerdem Mechanismen für das Ablegen von konstanten Ressourcen im Application Cache.

Die Offline Application Cache API stellt Funktionen bereit, um Web Apps verfügbar zu machen, wenn keine Verbindung zum Netz besteht. Dazu legt der Entwickler in einer Manifest-Datei fest, welche HTML-Seiten und andere Ressourcen (JavaScript-Dateien, CSS-Dateien oder Bilder) abgespeichert werden sollen. Die Manifest-Datei muss dann nur noch im HTML-Dokument referenziert werden.

Beim Laden der Seite werden die definierten Dateien auf dem Client gespeichert und können dadurch offline verfügbar gemacht werden. Ein Feature, welches besonders bei mobilen Anwendungen, die den Schwankungen des Netzes unterliegen, sehr wichtig ist.

Weiter können die gespeicherten Inhalte das erneute Laden der Seite beschleunigen. Das sogenannte Caching wird zwar von vielen Browsern bereits implementiert, verhält sich jedoch weniger zuverlässig und ist nicht kontrollierbar.

Zu guter Letzt wird die zu ladende Datengröße minimiert, da der Browser nur Daten nachlädt, die sich auch verändert haben.

Die Synchronisation des Application Cache bei einer Veränderung an den Daten erfolgt erst nach einer Änderung in der Manifest-Datei. Bewährt hat sich dabei die Verwendung von Versionsnummern, um eine Änderung der Manifest-Datei zu signalisieren. Weiterhin kann die Aktualisierung mit einer Update-Methode erzwungen werden.

Ein bestehendes Problem ist die Entfernung der Daten aus dem Cache. Laut Spezifikation soll deren Löschung mit dem Zurücksetzen des „normalen“ Browsecache geschehen. Bisher besteht bei den mobilen Browsern noch keine Möglichkeit, den Application Cache aktiv zu löschen.

## 10.6 Optimierung mobiler Web Apps

Bei der Entwicklung von mobilen Anwendungen müssen die Limitierungen des Endgerätes beachtet werden. Neben den unterschiedlichen Bildschirmgrößen können die knapp verfügbaren Hardwareressourcen, die eingeschränkte Bandbreite, verschiedene Eingabemethoden und geringe Akkulaufzeiten von großer Bedeutung sein.

Die Bereitstellung einer Web App für unterschiedliche Endgerätekategorien erfordert eine gesonderte Behandlung der verschiedenen Bildschirmauflösungen, während die Inhalte selbst und die Logik der Anwendung meist gleich bleiben. Mit Cascading Stylesheets ist es möglich, zusätzliche Layoutdefinitionen für un-

terschiedliche Endgeräte bereitzustellen. Die Unterscheidung, wann vom Browser welches Layout verwendet werden soll, kann mit Hilfe von Media Types und Media Queries festgelegt werden. Media Types sind bereits aus CSS2 bekannt und ermöglichen die Angabe unterschiedlicher Stylesheet-Varianten für die Webseite. Verwendet wurde dieses Element häufig um eine Druckansicht einer Seite anzubieten. Durch das Hinzukommen der CSS Media Queries kann nun eine genauere, endgerätespezifische Beschreibung für die unterschiedlichen Styles abgelegt werden. Für die Media Queries stehen eine Reihe an Attributen bereit, um die Zielplattform zu beschreiben. Dazu gehören zum Beispiel die Maße und die Orientierung des Bildschirms. So können zum Beispiel Style-Definitionen für alle Geräte mit einer maximalen Bildschirmbreite von „960px“ hinterlegt werden.

Um das Erscheinungsbild mobiler Seiten weiter zu verbessern ist es sinnvoll, leere Bereiche zu minimieren und Abstände zwischen Layoutboxen zu verkleinern, um so die gesamte Bildschirmfläche zu nutzen. Hover-Definitionen können herausgenommen werden, da sie auf dem Touchscreen nicht sichtbar sind.

Mit dem CSS Box Model kann außerdem die Anzeigereihenfolge der Layoutboxen auf der Seite definiert werden. Dadurch können zum Beispiel Elemente der Seite, die auf dem Desktop nebeneinander angezeigt werden, auf dem mobilen Gerät in gewünschter Reihenfolge untereinander dargestellt werden.

Unter dem Begriff Responsive Layout versteht sich ein neues Webdesignkonzept, welches eine flexible Darstellung des Inhalts abhängig von der Größe des Ausgabemediums ermöglicht. Dabei wird der wesentliche Inhalt auf allen Ausgabemedien dargestellt und je nach Breite um weiteren Inhalt (zum Beispiel in zusätzlichen Spalten) ergänzt. Erreicht werden kann dies durch die Verwendung von Prozentzahlen bei der Größenangabe für Layoutboxen und Bilder sowie der oben beschriebenen Media Queries.

Für die Optimierung der Darstellung auf dem mobilen Endgerät können weitere Parameter eingesetzt werden. Dazu gehört unter anderem das Viewport-Element. Es beschreibt die sichtbare Fläche des Browsers und kann bei mobilen Webseiten für die Darstellung der Seite entscheidend sein. Unter Angabe der Breite, Skalierung und Zoombarkeit kann man dem Browser einige Voreinstellungen mitgeben um eine bessere Benutzung zu ermöglichen.

Um die Usability innerhalb der Anwendung zu verbessern, sollten zum Beispiel Telefon- oder SMS-Funktionen als Verknüpfungen hinter die Zahlen gelegt werden. Damit können die Nutzer mit einem Klick ein Telefonat oder eine SMS mit einem vordefinierten Text initiieren. Gerade auf dem mobilen Endgerät müssen die wichtigsten Funktionen auf kürzestem Weg erreicht werden können.

IOS-spezifische Einstellungen sind außerdem das Anzeigen der Webseite in Fullscreen-Modus oder das Hinterlegen eines Icons für den Home Screen.

Die Optimierung einer Webseite für das mobile Endgerät geht jedoch weit über eine verbesserte Darstellung der Seite auf dem Endgerät hinaus. Es ist natürlich für die Lesbarkeit und Bedienung der Anwendung wichtig, dass Elemente auf dem mobilen Endgerät gut sichtbar, Grafiken skaliert und die Schriftgrößen angepasst sind. Eine wesentliche Einschränkung neben der des kleinen Bildschirms ist jedoch auch die geringe Bandbreite, die dem Endgeräts zur Verfügung steht. Um die Performan-

ce der Anwendung zu verbessern, sollte die Anzahl von HTTP-Anfragen möglichst gering gehalten werden. Das Entfernen von eingebetteten Frames diverser Drittanbieter (z. B. von Facebook) kann ebenso helfen die Anfragen zu minimieren, wie das Löschen von ausgeblendeten Grafiken und nicht benötigten Skript- und CSS-Codeteilen. Außerdem ist eine Komprimierung der verwendeten Skript-Files sinnvoll, um die Größe der Anwendung zu reduzieren.

Um ein schnelleres Starten der App beim zweiten und wiederholten Aufruf zu erreichen, kann man den bereits beschriebenen Application Cache verwenden. Dadurch, dass die in der Manifest-Datei definierten Daten auf dem Client abgelegt werden, müssen sie erst bei einer Änderung erneut aus dem Internet geladen werden. Zu beachten ist hier jedoch, dass auch die Manifest-Datei übersichtlich bleiben muss, also nur die wichtigsten Dateien darin aufzulisten sind.

Ein weiterer Punkt, der Web Apps träge und langsam machen kann, sind häufige und aufwendige DOM-Manipulationen sowie ein überdurchschnittlich großes DOM. Grund dafür ist das vom Browser durchgeführte Reflow, ein wiederholtes Rendern der Seite, das bei jeder kleinen Veränderung im DOM ausgeführt wird.

Die Größe des JavaScript-Codes ist ebenfalls für die initiale Ladezeit der Seite entscheidend. Es gibt einige Techniken und Vorgehensweisen, die das Parsen des Codes beschleunigen können, aber auf die hier in der Kürze nicht eingegangen werden kann.

Als letzte Empfehlung kann noch die Vermeidung von Cookies und die Verwendung von Technologien zur clientseitigen Speicherung von Daten genannt werden.

Das W3C sowie die Webseite HTML5Rocks sprechen weitere Empfehlungen<sup>12</sup> für die Entwicklung von mobilen Apps aus.

## 10.7 Unterstützung durch Micro-Frameworks

Für mobile Endgeräte existieren eine Reihe von optimierten JavaScript-Bibliotheken, welche die Entwicklung von mobilen Web Apps vereinfachen können.

Einige davon bieten zum Beispiel JavaScript-Funktionen und vorgefertigte Stylesheets an, um das Erscheinungsbild von nativen Anwendungen einer Plattform nachzuempfinden. Die Scriptbibliotheken jQTouch<sup>13</sup>, iUI<sup>14</sup> und UIUIKit<sup>15</sup> gehören zu dieser Art von Frameworks und unterstützen vor allem bei der Entwicklung von Web Apps, die den nativen Anwendungen der iOS-Plattform nahe kommen sollen. In Zukunft werden sicher auch solche Frameworks für Android und andere Plattformen zu finden sein.

Allerdings sollte bei der Entwicklung von Anwendungen für multiple Plattformen abgewogen werden, inwieweit das Look & Feel dieser Anwendungen an eine

---

<sup>12</sup> <http://www.w3.org/TR/mwabp> und <http://html5rocks.com/mobile/mobifying.html>.

<sup>13</sup> <http://jqtouch.com/>.

<sup>14</sup> <http://code.google.com/p/iui/>.

<sup>15</sup> <http://code.google.com/p/iphone-universal/>.

bestimmte Plattform angepasst sein sollte. Denn möchte man sich dieser Herausforderung stellen und eine plattformspezifische Oberfläche für jede Plattform anbieten, wird man auf Probleme stoßen, wenn es heißt, bestimmte Interaktionstechniken und Gestaltungselemente einer Plattform auf eine andere abzubilden.

Die Argumentation, dass Web Apps kein natives Aussehen benötigen, sondern mit ihrem universellen Erscheinungsbild auf allen Plattformen gleich aussehen, macht bei den meisten Anwendungen durchaus Sinn. Zu beachten ist dabei, dass die Anwendung so universell sein muss, dass die Nutzer aller Plattformen sie möglichst intuitiv bedienen können.

Eine weitere Gruppe von JavaScript-Bibliotheken für die mobile Entwicklung stellt einen im Vergleich zur Desktopvariante reduzierten Funktionsumfang zur DOM-Manipulation bereit. Häufig unterstützen diese den Entwickler auch beim Umgang mit Touch-Events und bringen eine Reihe mobiler UI-Elemente wie Buttons, Listen, Menüansichten, etc. mit. Zu diesen Bibliotheken gehören unter anderem jQuery Mobile<sup>16</sup>, Zepto.js<sup>17</sup>, YUI Library<sup>18</sup> und XUI<sup>19</sup>.

Außerdem kann man eine Vielzahl von Bibliotheken finden, die kleine Aufgaben bewältigen. Dazu gehört zum Beispiel die Bibliothek Lawnchair<sup>20</sup>, welche Funktionen zum Speichern von Daten im JSON-Format in der Web App bereitstellt.

Viele der hier genannten Bibliotheken werden fortlaufend weiterentwickelt und stehen unter einer MIT- oder GNU-Lizenz. Sie können somit frei bei der Entwicklung von Web Apps verwendet werden.

Zu den kommerziellen Produkten gehört das Sencha Touch Framework<sup>21</sup>, welches ein sehr umfangreiches Framework darstellt. Neben der Bereitstellung einer Vielzahl moderner Bedienelemente, einer guten Unterstützung bei Formularen und seiner Cross-Plattform-Kompatibilität bringt es eine große Anzahl von weiteren Funktionen mit.

## 10.8 Ein Standard – viele Implementierungen

Da der HTML5-Standard einer andauernden Weiterentwicklung unterliegt und die Entwicklung der einzelnen HTML5-Bestandteile nicht parallel verläuft, wird es immer Features des HTML5-Standards geben, die schon sehr weit in der Spezifizierung sind, während andere bisher nur als Entwurf existieren. Diese Entwicklung spiegelt sich so auch in den Browsern wider.

Während einige Browserhersteller aktiv am Standard mitarbeiten und ständig an der Weiterentwicklung interessiert sind, übernehmen andere die Bestandteile des

---

<sup>16</sup> <http://jquerymobile.com/>.

<sup>17</sup> <http://zeptojs.com/>.

<sup>18</sup> <http://developer.yahoo.com/yui/>.

<sup>19</sup> <http://xuijs.com/>.

<sup>20</sup> <http://westcoastlogic.com/lawnchair/>.

<sup>21</sup> <http://www.sencha.com/>.

**Tabelle 10.1** Ergebnisse HTML5-Test

Betriebssystem	Punkte
Windows Phone 7	17
Android OS	182
iOS	206
BlackBerry OS	233

Standards erst nach und nach. Es wird wohl immer unterschiedliche Implementierungen des Standards durch die Browser geben. Dennoch kann man immer den kleinsten gemeinsamen Nenner an Funktionen finden, die von allen Browsern unterstützt werden.

Mit dem HTML5-Test von Niels Leenheer<sup>22</sup> haben wir einige mobile Browser auf Herz und Niere überprüft. Der Test wurde von Leenheer unabhängig vom HTML5-Standardisierungskomitee entwickelt und garantiert keine Vollständigkeit. Dennoch hilft er einen Überblick über den aktuellen Stand der mobilen Browser zu erhalten.

Für unsere Tests haben wir folgende Geräte verwendet: ein Apple iPhone (iOS 4.3), ein Samsung GALAXY S (Android 2.3.3), ein BlackBerry Torch 9800 (BlackBerry 6.4.0) und ein HTC HD7 (Windows Phone 7).

Für den Test wurde die HTML5-Testseite mit dem nativen Browser<sup>23</sup> der eben genannten Betriebssysteme aufgerufen. Der mobile Browser Firefox gehört nicht zu den nativen Browsern und wurde deshalb in diesem Test nicht berücksichtigt. Tabelle 1 zeigt die Ergebnisse des Tests unter Angabe der Plattform. Insgesamt können mit dem Test 400 Punkte erreicht werden. Diese Zahl wird jedoch von keinem auf dem Markt existierenden mobilen oder Desktop-Browser erreicht.

Die Browser der Betriebssysteme iOS und BlackBerry liegen im oberen Bereich, wenn man davon ausgeht, dass die besten Ergebnisse derzeit mit der Betaversion des Google Chromium Desktop Browsers (293 Punkte) erreicht werden können.

Danach kommt der Browser von Android und das Schlusslicht stellt der Windows Phone 7 Browser da. Mit seinen 17 erreichten Punkten wird deutlich, dass bisher keine HTML5-spezifischen Funktionen implementiert wurden. Allerdings soll bereits das Betriebssystem mit der Version 7.5 ein Update für den Browser beinhalten, welches die aktuelle HTML5-Spezifikation unterstützt. Bis das Update verfügbar ist, können eine Reihe an Workarounds verwendet werden, um einige der fehlenden Funktionalität des Browsers auszugleichen. Außerdem besteht die Möglichkeit einen alternativen Browser zu installieren.

Folgende wesentliche Gemeinsamkeiten und Unterschiede in den Implementierungen kann man aus dem Test herauslesen:

1. Die neuen HTML5-Elemente zur Strukturierung des Inhaltes können nur von BlackBerry und iOS interpretiert werden.

<sup>22</sup> <http://www.html5test.com>.

<sup>23</sup> „Nativer Browser“ bezeichnet den von der mobilen Plattform mitgelieferten Webbrowser, der keiner zusätzlichen Installation bedarf.

2. Formularbezogene Elemente werden am besten von BlackBerry unterstützt, gefolgt von iOS und Android.
3. Das Video- und Audio-Element wird von BlackBerry, Android und iOS erkannt. Bonuspunkte bekamen alle drei für die Unterstützung des MP3-Codecs. Zusätzliche Punkte erhielt iOS für die Unterstützung der Codecs MPEG, H.264, PCM und AAC.
4. Das Canvas-Element wird von allen Browsern außer dem vom Windows Phone 7 implementiert.
5. Die Application Cache API, Geolocation API und die Elemente LocalSession und SessionStorage werden ebenfalls von den drei besten Browsern implementiert.
6. WebGL<sup>24</sup> ist noch in keinem der mobilen Browser verfügbar.
7. Die Web Workers API<sup>25</sup> wird bisher nur vom BlackBerry Browser unterstützt.
8. Web Sockets<sup>26</sup> können nur unter BlackBerry und iOS verwendet werden.

Problematisch bleiben die vielen unterschiedlichen Browsersversionen der Endgeräte. Da die Nachlieferung der Updates für ein Betriebssystem in fast allen Fällen vom Hersteller des Endgerätes abhängt, kommt es häufig zu einer verzögerten Auslieferung der Updates. Im schlimmsten Fall werden die Aktualisierungen für den Gerätetyp ganz eingestellt. Entwickler müssen deshalb Fallbacklösungen für alte Browsersversionen implementieren, wenn sie auf allen Geräten gleiche Ergebnisse erzielen wollen.

## 10.9 Die Grenzen mobiler Web Apps

Web Apps unterliegen folgenden Einschränkungen: begrenzter Zugang zu den Hardwareressourcen mobiler Geräte wie Bluetooth, NFC, Dateisystem, Kamera oder Telefonfunktionen und die geringe Performance bei multimedialen Anwendungen bzw. Spielen. Das bedeutet: abhängig von der geplanten Funktionalität der Anwendung ist zu entscheiden, ob eine Web App oder eine native Entwicklung in Frage kommt.

Web Apps eignen sich vor allem für die Präsentation von Medieninhalten (z. B. Vimeo), Firmendarstellungen (z. B. Clearleft), Blogs (z. B. Think Vitamin), Newsportale (Gigaom, Heise) aber auch Mailclients (z. B. Gmail) und Communities (z. B. Google+). Native Anwendungen haben Zugriff auf viele Funktionalitäten des Endgerätes und eignen sich deshalb für rechenintensive Anwendungen wie Spiele oder Fotobearbeitungstools sowie für Anwendungen, die Zugriff auf die Kamera benötigen, wie zum Beispiel ein Barcode-Scanner. Native Anwendungen können viel

---

<sup>24</sup> WebGL ist eine Bibliothek um 3D-Grafiken im Browser zu rendern.

<sup>25</sup> Die Web Workers API dient zur Ausführung von Skripten im Hintergrund.

<sup>26</sup> Web Sockets ermöglichen eine bidirektionale Verbindung zwischen der Web App und dem Server.

intensiver auf die mobile Plattform eingehen und dem Nutzer ein besonderes Erlebnis bei der Benutzung der Anwendung bieten.

An den Ergebnissen des HTML5-Tests wurde deutlich, dass die Implementierungen der Spezifikation sehr unterschiedlich vorangetrieben werden. Das erschwert die Entwicklung einer Web App, die auf allen Plattformen gleich dargestellt werden soll. Bei der Entwicklung einer Web App wird man immer mit der Fragmentierung der Browser umgehen müssen.

Eine weitere Grenze von Web Apps ist ihre Erreichbarkeit. Besteht keine Verbindung zum Internet und liegen die benötigten Daten nicht auf dem Gerät vor, kann die Web App nicht verwendet werden.

Der letzte Punkt ist der Vertrieb von mobilen Anwendungen. Web Apps sind für alle Nutzer über das Internet erreichbar. Um die Web Apps über eine Plattform wie den Android Market oder den Apple App Store zu verkaufen, ist nur ein Umweg über eine hybride Anwendungsentwicklung möglich. Bei einer hybriden Anwendung wird der Teil der Anwendung, der mittels Webtechnologien entwickelt wurde, innerhalb einer nativen Anwendung gekapselt und ausgeführt.

## 10.10 Die Zukunft von Web Apps

Dass mobile Web Apps eine Zukunft haben, ist aus heutiger Sicht unbestritten. Web Apps stellen heute für einen Softwareanbieter die einzige Möglichkeit dar, weitgehend unabhängig von dominierenden Plattformen und Ecosystemen zu agieren. Nach Aussagen der Hersteller Opera und Mozilla<sup>27</sup> ist die Anzahl der Webentwickler mit geschätzt 3 Million wesentlich größer als die Anzahl der Entwickler nativer mobiler Apps mit ca. 53.000. Damit besteht die Chance, dass mobile Web Apps abhängig von der weiteren Entwicklung von HTML5 in den nächsten 3 Jahren eine dominierende Rolle spielen werden.

Die aktuellen mobilen Browser bieten bereits eine gute Unterstützung des HTML5-Standards an. Es ist davon auszugehen, dass sich daran auch in naher Zukunft nichts ändern wird. Bestehende Lücken in der Browserimplementierung bzgl. HTML5 werden sowohl von W3C als auch WHATWG intensiv verfolgt. Wenn die Spezifikation von HTML5 weiter fortgeschritten ist, werden auch die anderen Browser und Plattformen mit der Unterstützung nachziehen. Ist dieser Punkt erreicht, kann man bei HTML5 von einer echten plattformübergreifenden Technologie sprechen. Offen bleibt die Frage nach weitreichenden Zugriffsmöglichkeiten der Web Apps auf die Gerätefunktionen und die Hardware der Endgeräte. In der Roadmap der W3C „Standards for Web Applications on Mobile“<sup>28</sup> wird der aktualisierte Status bzgl. Implementierungsgrad der einzelnen Technologien dargestellt. Dabei kann man feststellen, dass ein Großteil der Zugriffsmöglichkeiten heute bei den Browser- bzw. Plattformherstellern nur begrenzt oder überhaupt nicht implemen-

<sup>27</sup> <http://www.itespresso.de/2011/03/04/cebit-opera-und-mozilla-wollen-web-apps/>.

<sup>28</sup> <http://www.w3.org/2011/05/mobile-web-app-state.html>.

tier ist. Es gibt jedoch Bestrebungen der Endgerätehersteller (Device Specification WAC 2.0)<sup>29</sup>, weitreichende Zugriffsmöglichkeiten auf die Endgeräte zu ermöglichen. Öffnen sich die Hersteller in den nächsten Jahren diesen Anforderungen, erhält das Thema Sicherheit eine neue Qualität, da mit dem Zugriff auf das Endgerät der bisherige Sicherheitsaspekt von mobilen, isoliert ablaufenden Web Apps aufgeweicht wird. Eine weitere Verbreitung von mobilen Web Apps könnte auch durch webbasierte Desktopanwendungen entstehen, da hier zunehmend auch mobile Geräte wie Smartphones oder Tablets eingebunden werden. Eine weitere Frage stellt sich wie sich der aktuelle App Store-Gedanke künftig auf Web Apps übertragen lässt. Aber auch hier gibt es erste Pläne verschiedener Anbieter (u. a. Projekt Spartan von Facebook<sup>30</sup>).

Als Alternative zur reinen Web App ist die Entwicklung von hybriden plattformübergreifenden Apps zu sehen. Hier wird von einer Web App ausgegangen, die mit Hilfe von Entwicklertools in eine plattformabhängige native App umgewandelt wird. Außerdem stellen diese Tools als Erweiterung den Zugriff der App auf Gerätelfunktionen und die Hardwareressourcen bereit. Die Zukunft dieser Variante der App-Entwicklung liegt vor allem in Anwendungen, die zwar eine höhere Performance als Web Apps bzw. ein natives User Interface benötigen, bei denen der Entwicklungsprozess jedoch weitgehend plattformübergreifend geschehen soll. Zudem sind hybride Apps auch in naher Zukunft die einzige Möglichkeit um Web Apps über bestehende App Stores vermarkten zu können.

Mittel- bis langfristig haben Web Apps die Chance, vor allem bei geräteübergreifenden Anwendungen, deren Funktionalität überwiegend in der Cloud liegt, eine wesentliche Rolle zu spielen. Dabei kommen auch neue webzentrierte Betriebssysteme zum Einsatz, wie z. B. Chrome OS. So wird im EU-Forschungsprojekt Webinos der Gedanke der übergreifenden webbasierten Anwendungen auf verschiedenste Gerätetypen (Fahrzeug, Home Entertainment, Smartphones und PC) erweitert.

---

<sup>29</sup> <http://specs.wacapps.net/2.0/jun2011/>.

<sup>30</sup> <http://techcrunch.com/2011/06/15/facebook-project-spartan/>.

---

# Kapitel 11

## Die Grenzen des Browsers durchbrechen

### Hybride Anwendungsentwicklung für mobile Endgeräte

A.R.S. Gerlicher

**Zusammenfassung** Eine der Herausforderungen bei der Entwicklung mobiler Anwendungen ist die Heterogenität der mobilen Plattformen. Um eine möglichst große Zahl an Endgeräten zu unterstützen, ist die Kenntnis verschiedenster Betriebssysteme<sup>1</sup>, Entwicklungsumgebungen, Programmierschnittstellen und Programmiersprachen<sup>2</sup> notwendig. Dies erschwert die Entwicklung und Wartung einer mobilen Anwendung und ist zeit- und kostenintensiv. Sind in einem Unternehmen verschiedene mobile Endgeräte zur Nutzung durch die Mitarbeiter zugelassen und möchte man diese produktiv einsetzen, so ist eine plattformübergreifende Anwendungsentwicklung notwendig. Webbasierte Anwendungen stellen hier eine mögliche Lösung dar. Viele mobile Webbrowser unterstützen bereits neue Funktionen der zukünftigen HTML Version HTML5, wie zum Beispiel W3C Web Storage<sup>3</sup> oder die W3C Geolocation<sup>4</sup> API. Damit ist es möglich, webbasierte Anwendungen zu entwickeln, die ohne bestehende Internetverbindung nutzbar und funktional sowie optisch nur schwer von nativen Anwendungen zu unterscheiden sind. Webbasierte Anwendungen sind durch den Browser des Endgerätes in ihren Möglichkeiten begrenzt. Endgerätfunktionen (wie zum Beispiel der Zugriff auf die Kamera des Geräts), die momentan nicht durch eine standardisierte Browserschnittstelle zur Verfügung stehen, können durch eine webbasierte Anwendung nicht genutzt werden. Sogenannte hybride Apps durchbrechen die durch den Browser vorgegebene Grenze und ermöglichen über zusätzliche Programmierschnittstellen den Zugriff einer webbasierten Anwendung auf native Endgerätfunktionen. Dieses Kapitel widmet sich den Möglichkeiten zur Entwicklung mobiler hybrider Anwendungen als echte Alternative

---

Ansgar Gerlicher

Hochschule der Medien Stuttgart, Studiengang Mobile Medien, Professur Mobile Applications,  
E-mail: gerlicher@hdm-stuttgart.de

<sup>1</sup> Z. B. Symbian OS, iOS, Android, BlackBerry OS, Windows Mobile/Phone 7, WebOS, Bada.

<sup>2</sup> Z. B. Symbian C++, Objective-C, Java, C#.

<sup>3</sup> Web Storage: <http://www.w3.org/TR/webstorage/>, vom 8.5.2011.

<sup>4</sup> Geolocation API Specification: <http://www.w3.org/TR/geolocation-API/>, vom 8.5.2011.

zur nativen Anwendungsentwicklung. Es werden verschiedene Möglichkeiten zur Entwicklung mobiler hybrider Apps vorgestellt und die Vor- und Nachteile gegenüber nativen Anwendungen besprochen.

## 11.1 Hybride mobile Anwendungen

Hybride Anwendungen bestehen aus einer mittels Webtechnologien (HTML, JavaScript und CSS) entwickelten Anwendungsfunktionalität, die innerhalb einer nativen Anwendung gekapselt und ausgeführt wird. Die eigentliche Anwendungslogik liegt dabei nicht im nativ entwickelten Quellcode vor, sondern wird in Form von HTML, JavaScript und CSS implementiert. Mittels einer JavaScript-Abstraktionschicht kann die Anwendungslogik auf native Endgerätfunktionen zugreifen. Dies erlaubt es, bestehende mobile Webanwendungen in native Anwendungen zu verpacken, um sie zum Beispiel mit nativen Funktionen zu erweitern. So erstellte hybride Anwendungen können in die App Stores der verschiedenen Anbieter hochgeladen werden und diesen Vertriebskanal zusätzlich nutzen.

### 11.1.1 Technologie der hybriden Anwendungen

Innerhalb des nativen Anwendungs-Containers einer hybriden Anwendung wird häufig eine native Browserkomponente des Betriebssystems verwendet, um die in HTML, CSS und JavaScript vorliegende Anwendungsfunktionalität bereitzustellen. Browserkomponenten besitzen in der Regel eine HTML- und CSS-Render-Engine sowie einen JavaScript-Interpreter. Die Render-Engine der Browserkomponente ist für das Rendern der grafischen Benutzeroberfläche zuständig. Die Interaktion des Benutzers mit der Anwendung geschieht hauptsächlich über diese, welche meist den kompletten Bildschirm des Endgerätes ausfüllt. Die Anwendungslogik wird in JavaScript implementiert, das vom integrierten JavaScript-Interpreter der Browserkomponente ausgeführt wird. Da es sich bei dieser Art der hybriden Anwendung im Prinzip um eine Webanwendung handelt, können beliebige JavaScript Frameworks<sup>5</sup> wie zum Beispiel Sencha Touch<sup>6</sup>, jQuery Mobile<sup>7</sup> oder The-M-Projekt<sup>8</sup> zur Entwicklung der graphischen Benutzeroberfläche und Anwendungslogik genutzt werden. Der Zugriff auf native Endgerätfunktionen, wie zum Beispiel die eingebauten Kamera, Beschleunigungssensoren, Kompass oder Adressbuch, wird über eine

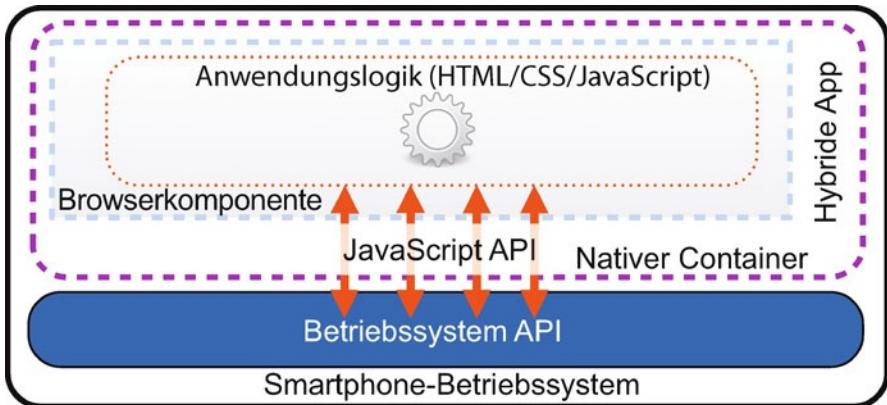
---

<sup>5</sup> Ein Framework, englisch für Rahmenstruktur, ist ein Programmgerüst zur Softwareentwicklung.

<sup>6</sup> JavaScript Framework der Firma Sencha, <http://www.sencha.com/>.

<sup>7</sup> Abkömmling des OpenSource JavaScript Framework Projekts jQuery, <http://jquerymobile.com/>.

<sup>8</sup> MVC basiertes HTML5 JavaScript Framework für mobile Anwendungen, OpenSource-Projekt (MIT Lizenz).



**Abb. 11.1** Schematischer Aufbau einer hybriden Anwendung mit Browerkomponente

Schnittstelle, die von der Anwendung in Form einer JavaScript-Bibliothek eingebunden wird, ermöglicht.

Die JavaScript-Bibliothek abstrahiert den Zugriff auf die native Programmierschnittstelle des Betriebssystems (Application Programming Interface, API), welche je nach verwendeter Browerkomponente und darunter liegendem Betriebssystem unterschiedlich ausgeprägt sein kann. Die Abstraktion der nativen Betriebssystemfunktionen erleichtert eine plattformunabhängige Implementierung der Anwendung. In Abb. 11.1 ist der Aufbau einer hybriden Anwendung mit Browerkomponente schematisch dargestellt.

Neben dem Einbetten einer Browerkomponente in den nativen Anwendungs-Container gibt es auch die Möglichkeit, nur einen JavaScript-Interpreter einzusetzen. In diesem Fall wird nicht nur die Anwendungslogik, sondern auch die grafische Benutzeroberfläche (engl. Graphical User Interface, GUI) der Anwendung in JavaScript (anstelle von HTML und CSS) implementiert. Es wird dazu eine zusätzliche Abstractionsschicht für die grafische Benutzeroberfläche eingeführt, die es ermöglicht über JavaScript auf native GUI-Funktionen des Betriebssystems zugreifen. Dieser Ansatz wird zum Beispiel vom Appcelerator Titanium Mobile Framework verwendet, auf das später in Abschn. 11.3.2 eingegangen wird. Damit ist es möglich, plattformunabhängige Anwendungen für Desktop und mobile Endgeräte in JavaScript zu entwickeln. Durch die Verwendung nativer GUI-Funktionen zur Darstellung der Benutzeroberfläche kann eine bessere Leistung im Vergleich zur HTML- und CSS-basierten Oberflächenprogrammierung erreicht werden. Gründe dafür liegen unter anderem in einer besseren Ausnutzung der Grafikhardware bei nativen GUI-Komponenten. Ein Nachteil ist dabei, dass im Vergleich zur browerkomponentenbasierten Lösung ein zusätzlicher Aufwand für das Abbilden der GUI-Abstractionsschicht auf native GUI-Funktionen entsteht. Da die GUI-Funktionen der verschiedenen Betriebssysteme sich teilweise stark unterscheiden, ist bei der plattformübergreifenden Anwendungsentwicklung eine Fallunterscheidung im JavaScript-Code notwendig. Plattformübergreifende Frameworks zur Ent-

**Tabelle 11.1** Vergleich der Ansätze zur Entwicklung hybrider Anwendungen

Hybrider Ansatz	Vorteile	Nachteile
Browser-komponente	Durch Betriebssystemunabhängige GUI (basiert auf HTML, CSS) einfachere Implementierung für verschiedene (und meist mehr) Plattformen	GUI Leistung abhängig von Render-Engine der Browserkomponente
JavaScript-Interpreter	Verwendung nativer GUI-Komponenten und dadurch plattformabhängige (optisch kein Unterschied zur nativen Anwendung) und hardwareunterstützte GUI	Zusätzlicher Aufwand durch Abbilden der GUI-Abstraktionsschicht auf native GUI-Funktionen und dadurch eingeschränkte Plattformunterstützung.

wicklung hybrider Anwendungen, die auf einem JavaScript-Interpreter basieren, unterstützen daher momentan meist nur eine kleine Anzahl mobiler Plattformen (zum Beispiel iOS und Android). Da bei browserkomponentenbasierten hybriden Anwendungen keine Abbildung der GUI-Funktionen notwendig ist, unterstützen plattformübergreifende Frameworks wie zum Beispiel PhoneGap oft bis zu sechs oder sieben verschiedene mobile Plattformen. Welchen Ansatz man für die Entwicklung einer hybriden Anwendung wählt, hängt davon ab, wie viele Plattformen unterstützt werden sollen und welche Anforderungen an die GUI der Anwendung gestellt werden. Ein Kriterium kann zum Beispiel sein, dass die Anwendung möglichst der GUI der jeweiligen Plattform (z. B. Android, iOS) entsprechen soll oder, im anderen Fall, eine plattformunabhängige GUI erwünscht ist. Auch kann die Leistung der GUI einen großen Einfluss auf die Benutzererfahrung (engl. „User-Experience“) haben und sollte daher bei der Wahl des hybriden Ansatzes mit in Betracht gezogen werden. Eine kurze Übersicht über die Vor- und Nachteile der zwei Ansätze wird in Tabelle 11.1 dargestellt.

### **11.1.2 Kopplung von JavaScript mit nativem Code bei Verwendung einer Browserkomponente**

Um eine Verbindung zwischen JavaScript-Funktionen und der nativen Betriebssystem-API herzustellen, bieten die nativen Browserkomponenten vordefinierte Schnittstellen. Im Folgenden wird eine solche Schnittstelle am Beispiel der Plattformen iOS und Android betrachtet.

#### **11.1.2.1 Beispiel iOS**

Die native Browserkomponente wird unter iOS durch die Klasse *UIWebView* repräsentiert. Eine hybride Anwendung für iOS-Geräte würde zum Beispiel diese *UIWebView*-Komponente einbetten, um den Anwendungsinhalt bereitzustellen. Das

*UIWebViewDelegate*-Protokoll wird verwendet, um auf bestimmte Ereignisse innerhalb dieser *UIWebView*-Komponente zu reagieren. Ein solches Ereignis ist zum Beispiel das Laden einer URL<sup>9</sup>. Um nun aus JavaScript heraus eine native Betriebssystemfunktion aufzurufen, kann man sich des *UIWebViewDelegate*-Protokolls bedienen. Hat der native Anwendungs-Container dieses Protokoll implementiert, so wird er beim Laden einer URL innerhalb der *UIWebView* über den Aufruf der entsprechenden *UIWebViewDelegate*-Methode benachrichtigt. Im nächsten Schritt kann die dabei übergebene URL auf bestimmte Schlüsselwörter geprüft werden. Wird in der URL ein zuvor definiertes Schlüsselwort gefunden, so wird die entsprechende Betriebssystem-Funktion aufgerufen. Der folgende Beispielcode zeigt, wie der Aufruf innerhalb einer JavaScript-Bibliothek aussehen könnte.

**Listing 11.1** Innerhalb der Browserkomponente ausgeführter JavaScript-Code

```
1 var url = "JSLNK://" + funktion + "/" + parameter;
2 window.location.href = url;
```

Zunächst wird hier eine URL initialisiert, die als URL-Schema das selbst definierte Schlüsselwort *JSLNK* enthält. Nach dem Schema folgt der Name der aufzurufenden nativen Funktion und die entsprechenden Parameter. Im nächsten Schritt wird die URL der *href*-Eigenschaft des *location*-Objekts zugewiesen, um diese im *UIWebView* zu laden. Bevor die URL jedoch geladen wird, benachrichtigt der *UIWebView* den nativen Anwendungs-Container durch Aufrufen der entsprechenden *UIWebViewDelegate*-Methode. Wie innerhalb des nativen Teils der Anwendung darauf reagiert wird, zeigt folgender Beispielcode:

**Listing 11.2** Objective-C-Implementierung des *UIWebViewDelegate*-Protokolls

```
1 - (BOOL)webView:(UIWebView *)theWebView
2 shouldStartLoadWithRequest:(NSURLRequest*) request
3 navigationType:(UIWebViewNavigationType)navigationType {
4 NSURL *url = [request URL];
5 if ([[url scheme] isEqualToString:@"JSLNK"]) {
6 // Parsen des URLRequest und Aufrufen der
7 // entsprechenden nativen Funktionen
8 return NO; }
9 else if ([[url isFileURL] ||
10 [[url scheme] isEqualToString:@"http"] {
11 // Falls es sich um eine normale URL handelt,
12 // diese vom UIWebView laden lassen
13 return YES; }
14 }
```

Die Methode *webView:shouldStartLoadWithRequest:navigationType:* wird aufgerufen, sobald im *UIWebView* eine URL geladen wird. An dieser Stelle kann nun

---

<sup>9</sup> URL: Uniform Resource Locator.

die übergebene URL geprüft werden. Entspricht das URL-Schema dem Schlüsselwort *JSLINK*, so wird der Rest der URL geparsst und die entsprechende native Funktion aufgerufen. Der Rückgabewert ist in diesem Fall *NO*, was dazu führt, dass der *UIWebView* nicht versucht, die URL zu laden. Entspricht das URL-Schema einer Datei-URL oder HTTP bzw. HTTPS, so wird *YES* zurückgegeben und damit die URL von der *UIWebView* geladen. Im nächsten Schritt soll nach dem Aufruf einer nativen Funktion (zum Beispiel dem Auslesen des Beschleunigungssensors des Endgerätes) ein Ergebnis an die in der Browserkomponente laufende Anwendungslogik übergeben werden. Dazu wird im JavaScript-Teil der Anwendung meist eine entsprechende Call-Back-Funktion definiert. Diese Call-Back-Funktion kann nun aus dem nativen Anwendungs-Container heraus aufgerufen werden. Die Klasse *UIWebView* bietet dafür die folgende Methode:

**Listing 11.3** *UIWebView*-Methode zur Evaluierung von JavaScript-Code

```
1 - (NSString *)stringByEvaluatingJavaScriptFromString :  
2 (NSString *)script
```

Durch Aufruf dieser Methode ist es möglich, im Kontext des *UIWebView* beliebigen JavaScript-Code auszuführen, der in Form eines *NSString*-Objekts übergeben wird. Es gibt dabei die Einschränkung, dass der JavaScript-Code nicht länger als 10 Sekunden laufen und nicht mehr als 10 MB Speicher allokieren darf. Wird diese Einschränkung nicht eingehalten, so wird der JavaScript-Code sofort vom *UIWebView* beendet. Die Methode liefert einen *NSString* mit den Ergebnissen des ausgeführten Scripts oder im Fehlerfall *nil* zurück. Wie dieses Beispiel zeigt, ist eine Verknüpfung der Anwendungslogik in JavaScript mit nativen Betriebssystemfunktionen über die vorhandenen iOS-Schnittstellen relativ einfach zu realisieren. Andere mobile Betriebssysteme bieten ähnliche Schnittstellen.

### 11.1.2.2 Beispiel Android

Android bietet die Klasse *WebView* als Browserkomponente, die über folgende Methode verfügt, um aus JavaScript heraus nativen Java-Code anzusprechen:

**Listing 11.4** Methode der *WebView*-Klasse zur Anbindung von Java-Objekten an JavaScript

```
1 void addJavascriptInterface ( Object obj ,  
2 String interfaceName );
```

Damit kann ein beliebiges Java-Objekt an den JavaScript-Code gebunden werden und von diesem über den im zweiten Parameter angegebenen Namen angesprochen werden. Alternativ gibt es unter Android die Möglichkeit, die Klasse *WebViewClient*<sup>10</sup> zu nutzen, um über bestimmte Ereignisse wie zum Beispiel das Laden

---

<sup>10</sup> Die Klasse befindet sich im Package: *android.webkit*.

einer URL benachrichtigt zu werden. Damit ist ein ähnlicher Lösungsansatz, wie er für iOS beschrieben wurde, möglich. Durch Ableiten der `WebViewClient`-Klasse und Überladen der Methode `shouldOverrideUrlLoading(WebView view, String url)`; können URLs auf Schlüsselwörter überprüft und es kann dementsprechend im nativen Code darauf reagiert werden. Die Methode `setWebViewClient(WebViewClient client)`; der Klasse `WebView` wird verwendet, um den `WebViewClient` beim `WebView` zu registrieren, damit dieser über entsprechende Ereignisse benachrichtigt wird.

Der Weg zurück von der Java-Seite auf die JavaScript-Seite kann zum Beispiel mittels der folgenden Funktion realisiert werden:

**Listing 11.5** WebView-Methode zum Aufruf einer URL aus Java

```
1 void loadUrl ( String url );
```

Der übergebene String kann dabei JavaScript-Code enthalten wie in folgendem Beispielcode:

**Listing 11.6** Übergabe von JavaScript-Code in der loadURL-Methode

```
1 webview . loadUrl ( " javascript : ( function () { " +
2 " alert (' Call – Back ' ) ; } ) () " );
```

Das Einbetten einer Browserkomponente ist eine elegante Möglichkeit, hybride Anwendungen zu erstellen. Da die meisten Plattformen über eine solche Browserkomponente verfügen, ist es mit relativ wenig Aufwand möglich, plattformübergreifende Anwendungen auf Basis von HTML, CSS und JavaScript zu realisieren. Auf diese Art und Weise arbeiten verschiedene Frameworks für die plattformübergreifende hybride Anwendungsentwicklung. Eines der bekanntesten ist das PhoneGap Framework, das später in Abschn. 11.3.1 besprochen wird.

## 11.2 Plattformspezifische SDKs zur Entwicklung hybrider Anwendungen

Die Entwicklung nativer Anwendungen für mobile Endgeräte wird oft als komplex und aufwändig wahrgenommen. Um eine größere Anzahl an Softwareentwicklern (insbesondere aus der Webentwicklung) für ihre Plattformen begeistern zu können, haben verschiedene Firmen, wie zum Beispiel Nokia (mit Symbian OS) und Research in Motion (mit BlackBerry OS), das Konzept der hybriden Anwendungsentwicklung aufgegriffen und entsprechende Werkzeuge und Anwendungen, so genannte Software Development Kits (SDK) zur Entwicklung hybrider Anwendungen bereitgestellt. Die Firma Palm Inc. ist bei der Entwicklung von WebOS noch einen Schritt weiter gegangen und hat dieses Konzept direkt in das WebOS-Betriebssystem integriert. Im folgenden Abschnitt werden verschiedene plattformabhängige Lösungen zur Entwicklung hybrider Anwendungen kurz vorgestellt.

## 11.2.1 webOS

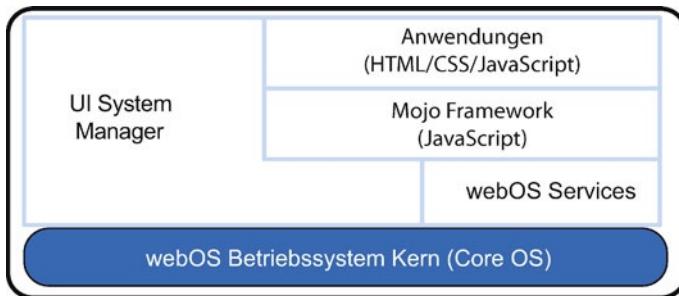
webOS ist ein proprietäres Betriebssystem für mobile Endgeräte, das auf dem Linux Kernel basiert [1]. Es wurde ursprünglich von der Firma Palm Inc. entwickelt und kam im Jahr 2009 mit dem Palm Pre auf den Markt. Im Jahr 2010 wurde Palm Inc. von der Firma Hewlett-Packard (HP) übernommen und webOS wird seitdem von HP weiterentwickelt. Die Endgeräte, auf denen webOS läuft, sind Smartphones. Ab Version 3.0 wurde das Betriebssystem auch für Tablet-Endgeräte wie das HP TouchPad optimiert. Eine Besonderheit von webOS gegenüber anderen mobilen Plattformen ist, dass es von vornherein auf eine möglichst gute Integration von sozialen Netzwerken und Web 2.0-Diensten ausgelegt wurde. Über die sogenannte Synergy-Funktion<sup>11</sup> können Daten auf dem Endgerät direkt mit webbasierten Diensten synchronisiert werden. Um die durchgängige Integration web-basierter Dienste zu unterstützen, wurde ein Konzept entwickelt, bei dem alle Anwendungen der Plattform auf Web 2.0-Standards aufbauen. Anwendungen werden dabei hauptsächlich mittels Webtechnologien wie HTML, CSS und JavaScript umgesetzt.

### 11.2.1.1 Hybride webOS-Anwendungen

Eine der zentralen Komponenten des webOS-Betriebssystems ist der sogenannte User Interface (UI) System Manager. Dieser regelt das Starten und den Lebenszyklus der webOS Anwendungen. Er bietet Funktionen zur Navigation zwischen Anwendungen auf dem Gerät, zum Abfragen des Systemzustands, zur lokalen und webbasierten Suche und zum Rendern von Anwendungen, die in HTML, CSS und JavaScript geschrieben sind. Es handelt sich beim UI System Manager um eine Art Laufzeitumgebung, die auf Standard-Browsertechnologie basiert. Dies ist vergleichbar mit einer hybriden Anwendung, die eine Browserkomponente verwendet, um die Anwendungsinhalte bereitzustellen. Betriebssystemfunktionen wie zum Beispiel der Zugriff auf GUI-Funktionen, auf Gerätehardware wie die Kamera, aber auch auf web-basierte und in das System integrierte Dienste werden über die sogenannten webOS Services und das JavaScript Framework Mojo bereitgestellt. Das Mojo Framework definiert verschiedene Funktionen, die von einer webOS-Anwendung implementiert werden müssen. Es stellt Funktionen zum Erstellen der grafischen Benutzeroberfläche und für die Ereignisbehandlung zur Verfügung. Mit HTML wird das Layout und mit CSS das Aussehen der Benutzeroberfläche definiert. Abbildung 11.2 zeigt den schematischen Aufbau der webOS-Architektur [2].

---

<sup>11</sup> Siehe Palm-Support: Wie funktioniert die Synergy-Funktion: [http://kb.palm.com/wps/portal/kb2/common/article/33923\\_de.html](http://kb.palm.com/wps/portal/kb2/common/article/33923_de.html), vom 17.6.2011.



**Abb. 11.2** Vereinfachte Darstellung der webOS-Architektur [2]

### 11.2.2 Nokia Web Widgets

Unter dem Begriff „Widget“ versteht man kleine Anwendungen, die mittels HTML, CSS und JavaScript implementiert werden. Sie werden meist in eine ZIP-Datei verpackt und können in dieser Form auf verschiedenen Endgeräten oder in manchen Web-Browsern (zum Beispiel Opera) installiert und genutzt werden. Bekannt geworden sind Widgets im Desktop-Bereich als kleine Anwendungen auf dem Mac OS X Dashboard oder seit Windows Vista auch als Sidebar-Gadgets. Seit der Version „Series 60 3rd Edition, Feature Pack 2“ (erschienen Anfang 2007) des Symbian-Betriebssystems besitzt jedes Symbian-Smartphone eine Version der sogenannten Web Runtime Platform (WRT) [3]. Die WRT ist eine in das Betriebssystem integrierte browserbasierte Laufzeitumgebung<sup>12</sup>, die zum einen als Basis für den Symbian Browser dient und es zum anderen ermöglicht, Widgets auf dem Endgerät zu installieren und auszuführen. Die Widgets können dabei über eine JavaScript API auf bestimmte Systemdienste (Platform Services), wie zum Beispiel auf den Kalender, die Kontakte, Sensoren oder Geolocation zugreifen. Da installierte Widgets optisch nicht von anderen Anwendungen zu unterscheiden sind, gibt es aus Sicht des Anwenders keinen Unterschied zwischen einem Widget und einer nativen Anwendung. Eine WRT-Widget-Installationsdatei ist dabei eine ZIP-komprimierte Datei mit der Endung „.wgz“, die entsprechende HTML-, CSS- und JavaScript-Dateien sowie Grafiken und eine Datei mit Metainformationen des Widgets enthält. Nokia war mit der Einführung der Web Widgets einer der Vorreiter hybrider Anwendungen auf mobilen Endgeräten. Mit den W3C<sup>13</sup> Widgets [4] findet momentan eine Standardisierung der Widget-Schnittstellen statt.

<sup>12</sup> WRT basiert auf dem WebKit OpenSource-Projekt, [www.webkit.org](http://www.webkit.org).

<sup>13</sup> W3C: World Wide Web Consortium, [www.w3.org](http://www.w3.org).

### ***11.2.3 BlackBerry WebWorks***

Mit der BlackBerry-WebWorks-Plattform [5] stellt die Firma Research In Motion (RIM) eine web-basierte Programmierschnittstelle für die BlackBerry-Smartphones und das PlayBook zur Verfügung. Ab der Version 5.0 des BlackBerry-Betriebssystems<sup>14</sup> ist die Entwicklung von sogenannten WebWorks-Anwendungen auf Basis von HTML, CSS und JavaScript möglich. Die Anwendungen laufen dabei ähnlich wie die Nokia Web Widgets in einer browserbasierten Laufzeitumgebung<sup>15</sup>, sind aber eigenständige Anwendungen. Sie unterscheiden sich aus Anwendersicht nicht von anderen BlackBerry-Anwendungen und können über den offiziellen BlackBerry AppWorld App-Store vertrieben werden. Zur Entwicklung der WebWorks-Anwendungen stellt RIM das BlackBerry WebWorks SDK [6] kostenlos zur Verfügung. Dieses enthält den WebWorks Packager, mit dem die erstellen Webanwendungen in WebWorks-Anwendungen übersetzt, signiert und Installationsdateien generiert werden können. Für den Zugriff auf Betriebssystemfunktionen, wie zum Beispiel das Adressbuch, den Kalender, Dateien und Telefonfunktionen, kann die BlackBerry WebWorks API [7] genutzt werden.

## **11.3 Plattformübergreifende Frameworks zur Entwicklung hybrider Anwendungen**

Wie in Abschn. 11.1.1 besprochen, erleichtert die JavaScript-Abstraktionsschicht einer hybriden Anwendung die Entwicklung plattformübergreifender Lösungen. Um eine hybride Anwendung für mehrere Betriebssysteme zur Verfügung zu stellen, müssen allerdings entsprechende native Anwendungs-Container erstellt und die abstrakten JavaScript-Funktionen mit den jeweiligen nativen Betriebssystemfunktionen verknüpft werden. Diese Aufgabe haben sich verschiedene Frameworks zur Entwicklung plattformübergreifender hybrider Anwendungen zum Ziel gesetzt. Zwei der prominentesten sind PhoneGap und Appcelerator Titanium Mobile, die in den folgenden Abschnitten beispielhaft besprochen werden.

### ***11.3.1 PhoneGap***

PhoneGap [8] ist ein OpenSource<sup>16</sup>-Projekt der Firma Nitobi Software (Nitobi wurde im Oktober 2011 von der Firma Adobe akquiriert) zur Erstellung plattformübergreifender Anwendungen für mobile Endgeräte. Anwendungen werden dabei in HTML, CSS und JavaScript entwickelt und mithilfe von plattformspezifischen

---

<sup>14</sup> Das PlayBook verwendet das Tablet OS, welches auf QNX basiert.

<sup>15</sup> WebWorks basiert ebenso wie Nokia Web Widgets auf dem WebKit OpenSource-Projekt.

<sup>16</sup> PhoneGap ist erhältlich unter einer modifizierten BSD oder der MIT-Lizenz (2008).

Anwendungsvorlagen in native Anwendungen übersetzt. Es werden aktuell folgende Plattformen unterstützt:

- Android
- Bada<sup>17</sup>
- BlackBerry OS<sup>18</sup>
- iOS
- Symbian OS
- webOS

Die Unterstützung von Windows Mobile bzw. Windows Phone 7 ist in Planung. Damit PhoneGap auch diese Plattformen unterstützen kann, werden bestimmte Browserfunktionen benötigt, welche von den aktuellen Internet-Explorer-Versionen für Windows Mobile bzw. Phone 7 nicht unterstützt werden. Mit dem angekündigten Internet Explorer 9 für die Windows-Phone-Plattform könnte sich das ändern.

Die von PhoneGap bereitgestellten Anwendungsvorlagen beinhalten neben dem plattformspezifischen Anwendungs-Container<sup>19</sup> ein *www*-Verzeichnis. In diesem Verzeichnis liegen die PhoneGap-JavaScript-Bibliothek und eine *index.html*-Datei. Diese *index.html*-Datei ist der Einstiegspunkt in die Anwendungslogik. Um nun eine hybride Anwendung mit PhoneGap zu entwickeln, muss lediglich der Inhalt der *index.html*-Datei angepasst werden und danach die Anwendung unter Verwendung der jeweiligen Anwendungsvorlage in eine native Anwendung der Zielplattform übersetzt werden. Der Zugriff auf native Funktionen, wie zum Beispiel die Beschleunigungssensoren des Endgeräts, geschieht über die mitgelieferte PhoneGap-JavaScript-Bibliothek. Soll die Anwendung später auf allen unterstützten Plattformen laufen, so muss theoretisch<sup>20</sup> in der Anwendungslogik nichts angepasst werden. Um die Anwendung für die Zielplattform zu übersetzen, werden Software-Entwicklungswerzeuge (engl. „Software Development Kit“, SDK) für die jeweilige Plattform benötigt. Für Android benötigt man das Android SDK [9], für Bada das Bada SDK [10], für BlackBerry das BlackBerry WebWorks SDK [6] und für iOS benötigt man Xcode [11]<sup>21</sup>. Die Symbian-OS-Vorlage von PhoneGap basiert auf Symbian WRT. Das heißt, es wird keine native Anwendung, sondern ein Nokia Web Widget erstellt und in eine „.wgt“ Datei verpackt, die dann auf einem Symbian-Gerät installiert werden kann. Für webOS wird das webOS SDK [12]<sup>22</sup> benötigt.

---

<sup>17</sup> Bada: Betriebssystem für mobile Endgeräte der Firma Samsung.

<sup>18</sup> Die unterstützten Funktionen unterscheiden sich je nach BlackBerry-Betriebssystem-Version.

<sup>19</sup> Der Anwendungs-Container liegt dabei in Form von Bibliotheken und Source-Code der jeweiligen Zielplattform vor.

<sup>20</sup> Leider gibt es momentan noch verschiedene Funktionen, die nicht auf jeder Plattform gleich gut funktionieren, daher ist eine Fallunterscheidung an manchen Stellen weiterhin notwendig.

<sup>21</sup> Xcode erfordert einen MacOS X-Rechner. Es gibt PhoneGap-Vorlagen für Xcode 3.5 und für Xcode 4.

<sup>22</sup> Erhältlich in der Version 2.1 für webOS Smartphones und 3.0 für das HP TouchPad, PhoneGap bietet momentan noch keinen offiziellen TouchPad-support.



**Abb. 11.3** Screenshot des PhoneGap-Simulators

Einer der größten Kritikpunkte an PhoneGap als Multi-Plattform-Framework war, dass man die jeweiligen Plattform-SDKs benötigt, um Anwendungen für mehrere Plattformen erstellen zu können. Die Firma Nitobi bietet daher den PhoneGap Build-Dienst [13] an. Mit PhoneGap entwickelte Anwendungen können dabei auf einen zentralen Server geladen und dort automatisiert für alle unterstützten Plattformen kompiliert werden. Dadurch werden die unterschiedlichen SDKs nicht mehr lokal vom Entwickler benötigt und er kann sich voll auf die Entwicklung der Anwendung mittels HTML, CSS und JavaScript konzentrieren. Der PhoneGap Build-Dienst befindet sich momentan noch in der Beta-Phase und wird voraussichtlich für OpenSource-Projekte kostenlos zur Verfügung stehen.

Um den Einstieg in die mobile Anwendungsentwicklung für verschiedene Plattformen zu erleichtern, wurde für das PhoneGap-Projekt ein Simulator auf Basis von Adobe AIR entwickelt, mit dem es möglich ist, bestimmte Teile der PhoneGap API zu testen (siehe Abb. 11.3). Zum Beispiel können damit die Beschleunigungssensoren, die Geräteausrichtung oder Geolocation simuliert werden, um unabhängig vom realen Endgerät eine PhoneGap-Anwendung zu testen. Es werden verschiedene Smartphone „Skins“ wie iPhone, Android G1 oder BlackBerry Storm angeboten, um einen Eindruck davon zu bekommen, wie die Anwendung auf dem realen Endgerät aussehen könnte.

### 11.3.2 Appcelerator Titanium Mobile

Appcelerator Titanium Mobile [14] (im folgenden Titanium genannt) ist ein Open-Source<sup>23</sup>-Projekt [15] der Firma Appcelerator Inc. zur Entwicklung nativer mobiler Anwendungen mittels Webtechnologien. Titanium verwendet im Gegensatz zu PhoneGap keine Browserkomponente<sup>24</sup>, sondern einen JavaScript-Interpreter wie in Abschn. 11.1.1 beschrieben, um Anwendungsinhalte darzustellen. Die Entwicklung geschieht dabei hauptsächlich in JavaScript unter Verwendung der Titanium JavaScript API. Der JavaScript-Code wird dann für die jeweilige Plattform in einen nativen Anwendungs-Container verpackt und zur Laufzeit von der im Anwendungs-Container enthaltenen JavaScript-Engine interpretiert. Titanium unterstützt momentan zwei mobile Plattformen: iOS und Android. Unter iOS wird die JavaScriptCore<sup>25</sup> und unter Android die Mozilla Rhino-Engine zur JavaScript-Interpretation eingesetzt. Funktionen der Titanium API zur Entwicklung der grafischen Benutzeroberfläche werden direkt auf native Betriebssystemfunktionen abgebildet. Durch die JavaScript-Abstraktionsschicht entsteht, im Vergleich zu echten nativen Anwendungen, ein geringer Leistungsverlust bei der Darstellung der grafischen Oberfläche. Alle anderen JavaScript-Funktionen, für die es keine native Entsprechung gibt, werden weiterhin von der jeweiligen JavaScript-Engine interpretiert<sup>26</sup>. Die Abbildung der abstrakten JavaScript-GUI-Funktionen auf ihre native Entsprechung ist nicht immer ohne weiteres möglich. Probleme entstehen zum Beispiel dann, wenn eine bestimmte GUI-Komponente zwar auf der einen Plattform, aber nicht auf der anderen Plattform existiert. Unter diesen Umständen ist eine Fallunterscheidung im Quellcode notwendig.

Wie bei PhoneGap auch werden zur Entwicklung einer Titanium-Anwendung die entsprechenden Plattform-SDKs und für iOS ein Apple-Mac-OS-X-Rechner benötigt. Zur Unterstützung der Anwendungsentwicklung existiert die OpenSource-Anwendung Titanium Developer [16]. Damit ist es möglich, an zentraler Stelle native Anwendungen für beide Plattformen<sup>27</sup> zu übersetzen, im Emulator und auf dem Endgerät zu testen sowie für den jeweiligen App-Store bereitzustellen. Die Programmierung erfolgt dabei außerhalb der Titanium-Developer-Anwendung in einem beliebigen Texteditor. Ein Titanium-Projekt besteht aus einem Verzeichnis, in dem sich eine Konfigurationsdatei mit dem Namen *tiapp.xml* und ein *Resources*-Unterverzeichnis befinden. Die *tiapp.xml*-Datei enthält Metainformationen wie Anwendungsname, Icon-Dateiname, Herausgeber und zielplattformabhängige Infor-

---

<sup>23</sup> Das Projekt steht unter der Apache Public License (Version 2).

<sup>24</sup> Erst seit der Version 1.0. In Vorgängerversionen wurde ebenfalls eine Browserkomponente genutzt.

<sup>25</sup> Der Interpreter ist Teil des WebKit-OpenSource-Projekts.

<sup>26</sup> Der im Safari-Browser integrierte JavaScript-Interpreter nutzt seit iOS Version 4.3 Just-In-Time (JIT) Kompilierung. Externen JavaScript-Interpretern wird derzeit keine JIT-Kompilierung ermöglicht.

<sup>27</sup> Titanium Developer unterstützt zusätzlich bei der plattformübergreifenden Entwicklung von Desktop-Anwendungen für Linux, Max OS und Linux.

mationen, wie zum Beispiel über die unterstützte Geräteausrichtung (engl. „device orientation“). Innerhalb des *Resources*-Verzeichnisses befindet sich die Datei *app.js*. Diese Datei ist der Einstiegspunkt in eine Titanium-Anwendung und wird beim Start der Anwendung vom nativen Anwendungs-Container ausgeführt. Um eine Anwendung mit Titanium zu entwickeln, muss der Inhalt dieser Datei angepasst werden und danach der Build-Prozess für die Zielplattform zum Beispiel mithilfe der Titanium-Developer-Anwendung gestartet werden. Titanium Developer wurde mittlerweile von der auf Eclipse basierten integrierten Entwicklungsumgebung (engl. „Integrated Development Environment“, IDE) Titanium Studio abgelöst. Titanium Studio existiert als kostenpflichtige Variante in einer sogenannten „Professional“- und einer „Indie“-Version. Eine kostenlose „Community“-Version mit eingeschränkten Funktionen wird zusätzlich angeboten.

## 11.4 Zusammenfassung und Ausblick

Ein großer Vorteil der hybriden Anwendungsentwicklung ist, dass Anwendungsentwickler mit HTML-, CSS- und JavaScript-Kenntnissen relativ schnell lauffähige native Anwendungen erstellen können, ohne sich in neue Programmiersprachen einarbeiten zu müssen. Besonders interessant sind hybride Anwendungen dann, wenn es darum geht, die Grenzen des Browsers zu durchbrechen und Endgerätfunktionen zu nutzen, für die (noch) keine Browserschnittstelle existiert. Die dazu notwendigen JavaScript APIs der hier vorgestellten Lösungen zur Entwicklung hybrider Anwendungen sind schnell erlernt. Ein weiterer Vorteil im Vergleich zu mobilen Webanwendungen ist die Möglichkeit, die Anwendungen über die entsprechenden App-Stores vertreiben zu können<sup>28</sup>.

Die hybride Anwendungsentwicklung eignet sich gut für das Rapid Prototyping von datengetriebenen Anwendungen, die keine besonders hohen Ansprüche an die Rechen- oder die Grafikleistung (wie zum Beispiel 3D-Spiele) stellen. Open-Source Frameworks wie PhoneGap und Appcelerator Titanium Mobile bieten durch die Plattformunabhängigkeit einen weiteren Vorteil gegenüber der nativen Anwendungsentwicklung. Mobile Multi-Plattform-Frameworks wie diese sind allerdings noch relativ jung<sup>29</sup> und daher in manchen Bereichen noch nicht ausgereift. Dies zeigt sich zum Beispiel in der teilweise unterschiedlichen Abbildung der JavaScript-Funktionen auf native Funktionen bei verschiedenen Plattformen<sup>30</sup>, wodurch eine Fallunterscheidung im Quellcode notwendig wird. Ein weiteres Problem dieser Lösungen ist die relativ schlechte Debugging-Unterstützung, die sich teilweise auf Fehlerausgaben auf der Konsole beschränkt. In zukünftigen Versionen wer-

---

<sup>28</sup> Anwendungen müssen nicht über einen App-Store verteilt werden, wenn dies zum Beispiel bei firmeninternen Anwendungen nicht erwünscht ist.

<sup>29</sup> PhoneGap entstand Ende 2008 auf dem iPhoneDevCamp in San Francisco, im Juni 2009 wurde die erste Beta von Titanium Mobile veröffentlicht.

<sup>30</sup> Bei Versuchen mit PhoneGap zeigte sich zum Beispiel, dass die Achsen-Werte der Beschleunigungssensoren unter iOS und Android vertauscht waren.

den diese Nachteile mit großer Wahrscheinlichkeit behoben oder in kostenpflichtigen Framework-Versionen angeboten. Die kostenpflichtige Version von Appcelerator Titanium Studio bietet bereits eine verbesserte Debugging-Unterstützung. Für die Entwicklung komplexerer plattformunabhängiger Geschäftsanwendungen sind die kostenlosen Framework-Versionen momentan nur bedingt zu empfehlen<sup>31</sup>. Das Geschäftsmodell der Entwickler von PhoneGap und Titanium beruht daher auf dem Support und der Bereitstellung erweiterter Funktionen, wie zum Beispiel der Debugging-Unterstützung.

Die Idee, HTML, CSS und JavaScript zur Entwicklung von Anwendungen zu nutzen – die auch außerhalb eines Browsers lauffähig sind –, ist sicherlich nicht neu. Nokia hat mit den Web Widgets allerdings als einer der ersten diese Technologie auf die Smartphones gebracht. Damit wurde eine weitere Alternative zur (auf mobilen Plattformen relativ komplexen) nativen Anwendungsentwicklung geschaffen und der Kreis der potenziellen Anwendungsentwickler stark vergrößert. HTML, CSS und JavaScript haben einen sehr großen Bekanntheitsgrad und sind leicht zu erlernen. Diesen Vorteil hat auch Palm Inc. früh erkannt und das webOS-Betriebssystem von vornherein so konzipiert, dass Anwendungen dafür hauptsächlich mit Webtechnologien entwickelt werden. Die junge webOS-Plattform hatte damit sofort eine große Zahl potenzieller Anwendungsentwickler für sich gewonnen. Mittlerweile gibt es eine Vielzahl an Frameworks und Werkzeugen zur Entwicklung hybrider Anwendungen für mobile Endgeräte. Die hier vorgestellten Frameworks sind nur ein Ausschnitt der momentan auf dem Markt erhältlichen Lösungen. Als eine weitere plattformabhängige Lösung für die Entwicklung von iOS-Anwendungen mittels Webtechnologien sei hier NimbleKit [17] genannt. Weitere Lösungen zur plattformübergreifenden Entwicklung hybrider Anwendungen sind zum Beispiel appMobi{!} [18], QuickConnectFamily [19] und Rhomobile [20].

Der Trend zum Einsatz von Webtechnologien in Stand-Alone-Anwendungen wird vermutlich auch durch neue Entwicklungen wie HTML5 unterstützt. Es werden in Zukunft viele Funktionen, die momentan nur über eine native Betriebssystem-API zur Verfügung stehen, auch über den Webbrowsr genutzt werden können. Der Zugriff auf Audio- und Video-Hardware ist nur ein Beispiel. Hierfür existiert momentan ein W3C Working Draft mit dem Titel HTML Media Capture [21]. Mit den Beschleunigungssensoren und der Geräteausrichtung beschäftigt sich der W3C Working Draft Device Orientation [22]. Wenn in Zukunft all diese Funktionen in allen Browsern der Endgeräte unterstützt werden, dann stellt sich die Frage, ob hybride Anwendungen noch notwendig sind. In Zukunft werden wahrscheinlich viele Stand-Alone-Anwendungen auf den meisten Plattformen mit Webtechnologien entwickelt werden, auch ohne dass zusätzlich ein nativer Anwendungs-Container benötigt wird. Zum Beispiel hat Microsoft angekündigt [23], dass Anwendungen für das zukünftige Windows-Betriebssystem (Codename Windows 8) auf HTML5-Technologien basieren werden.

---

<sup>31</sup> Diese Aussage bezieht sich auf Erfahrungswerte aus verschiedenen Projekten an der Hochschule der Medien Stuttgart. Zum Beispiel wurde die „Lange Nacht der Museen Stuttgart App“ mit Titanium Mobile entwickelt.

Anwendungen auf Basis von Webtechnologien machen natürlich nicht überall Sinn. Dort, wo es auf eine hohe Rechen- oder Grafikleistung ankommt, werden native Anwendungen noch länger Bestand haben. Aber auch dort könnten Technologien wie zum Beispiel WebGL [24] mittels Hardwareunterstützung früher oder später die native Anwendungsentwicklung ablösen.

## Literaturverzeichnis

1. Allen M (2009) Palm webOS. O'Reilly Media
2. HP webOS Developer Center ([developer.palm.com](http://developer.palm.com)), Overview of webOS, Palm webOS Architecture, <http://goo.gl/tjmZD>, vom 17.6.2011
3. Forum Nokia, Nokia web tools. <http://www.developer.nokia.com/Develop/Web/>, vom 17.6.2011
4. Widget Packaging and XML Configuration, W3C Working Draft 07.06.2011, <http://www.w3.org/TR/2011/WD-widgets-20110607/>, vom 17.6.2011
5. BlackBerry WebWorks Development. <http://us.blackberry.com/developers/browserdev/>, vom 17.6.2011
6. BlackBerry WebWorks SDK. <http://us.blackberry.com/developers/browserdev/widgetsdk.jsp>, vom 17.6.2011
7. BlackBerry WebWorks API. <http://www.blackberry.com/developers/docs/webworks/api/>, vom 17.6.2011
8. PhoneGap OpenSource Projekt. <http://www.phonegap.com>, vom 15.06.2011
9. Android SDK. <http://developer.android.com/sdk/index.html>, vom 17.6.2011
10. Bada SDK. <http://developer.bada.com/devtools/sdk>, vom 17.6.2011
11. Apple Xcode. <http://developer.apple.com/xcode/>, vom 17.6.2011
12. HP webOS Developer Center. <https://developer.palm.com/>, vom 17.6.2011
13. PhoneGap Build Service. <http://build.phonegap.com/>, vom 17.6.2011
14. Appcelerator Titanium. <http://www.appcelerator.com/products/titanium-mobile-application-development/> vom 15.06.2011
15. Appcelerator Titanium Mobile OpenSource Projekt. [https://github.com/appcelerator/-titanium\\_mobile](https://github.com/appcelerator/-titanium_mobile), vom 15.06.2011
16. Appcelerator Titanium Developer OpenSource Projekt. [https://github.com/appcelerator/-titanium\\_developer](https://github.com/appcelerator/-titanium_developer), vom 15.06.2011
17. NimbleKit Framework. <http://www.nimblekit.com>, vom 15.06.2011
18. AppMobi. <http://www.appmobi.com>, vom 15.06.2011
19. QuickConnectFamily. <http://www.quickconnectfamily.org>, vom 15.06.2011
20. Rhodes Rhomobile. <http://rhomobile.com>, vom 15.06.2011
21. HTML Media Capture, W3C Working Draft. <http://www.w3.org/TR/2011/WD-html-media-capture-20110414>, vom 17.6.2011
22. DeviceOrientation Event Specification, W3C Working Draft. <http://dev.w3.org/geo/api/spec-source-orientation.html>, vom 17.6.2011
23. ZDNet (www.zdnet.de), Microsoft: Sinofsky zeigt Tablet-Interface von Windows 8, <http://goo.gl/MKD4U>, vom 2.6.2011
24. Khronos Group, WebGL Specification. <http://www.khronos.org/registry/webgl/specs/latest/>, vom 17.6.2011

---

# Kapitel 12

## Technologische und marktseitige Unsicherheit bei der Neuentwicklung von Mobile Enterprise Services

Frank Weiß und Matthias Söllner

**Zusammenfassung** Mobile Dienstleistungen, sog. „Mobile Services“, haben in den letzten Jahren stark an Popularität gewonnen. Besonders im Konsumentenmarkt haben sich Applikationen auf mobilen Plattformen zu einem Massenphänomen entwickelt. Mit der wachsenden Mobilität im Arbeitsumfeld und den steigenden Einflüssen aus dem Konsumentenbereich auf das Unternehmensumfeld („Consumerization“) ist ein steigender Bedarf an mobilen Unternehmenslösungen („Mobile Enterprise Services“) festzustellen. IT-Dienstleister, die Mobile Enterprise Services in ihrem Produktpotfolio haben und diese weiter ausbauen wollen, sind mit spezifischen Herausforderungen konfrontiert. Basierend auf einer Reihe von Fallstudien wurden die heterogene, sich schnell ändernde mobile Infrastruktur, hohe Sicherheitsanforderungen und die Notwendigkeit der Integration in Unternehmensinfrastruktur als Rahmenbedingungen identifiziert, die eine technologische Unsicherheit für IT-Dienstleister erzeugen. Starkes Marktwachstum und die Consumerization des Unternehmensumfelds erzeugen marktseitige Unsicherheit.

### 12.1 Thematische Einführung

Mobile Dienstleistungen, sog. „Mobile Services“, haben in den letzten Jahren stark an Popularität gewonnen. Besonders im Konsumentenmarkt haben sich Applikationen auf mobilen Plattformen zu einem Massenphänomen entwickelt. Dies wurde begünstigt und vorangetrieben durch technologische Entwicklungen im Bereich

---

Frank Weiß  
Universität Kassel/Detecon International, München,  
E-mail: weiss@wi-kassel.de

Matthias Söllner  
Universität Kassel, Fachgebiet Wirtschaftsinformatik,  
E-mail: soellner@uni-kassel.de

mobilier Endgeräte, mobiler Funknetze sowie mobiler Applikationen [21]. Im Unternehmensumfeld haben hohe Sicherheits- und Verfügbarkeitsanforderungen die Verbreitung von Mobile Services stark gebremst [21]. Mit der wachsenden Mobilität im Arbeitsumfeld und den steigenden Einflüssen aus dem Konsumentenbereich auf das Unternehmensumfeld („Consumerization“) ist ein steigender Bedarf an mobilen Unternehmenslösungen („Mobile Enterprise Services“) festzustellen.

IT-Dienstleister, die Mobile Enterprise Services in ihrem Produktportfolio haben und diese weiter ausbauen wollen, sind mit einer Reihe von spezifischen Herausforderungen konfrontiert. Ziel dieses Beitrags ist die Identifikation und Analyse der Rahmenbedingungen, die eine technologische und marktseitige Unsicherheit erzeugen. Diese Unsicherheit muss durch IT-Dienstleister im Rahmen der Service-Neuentwicklung beherrscht werden. Die hieraus abgeleiteten Fragestellungen, die im Rahmen dieses Beitrags diskutiert werden, stellen sich wie folgt dar:

- Welche existierenden Rahmenbedingungen erzeugen technologische oder marktseitige Unsicherheit für IT-Dienstleister, im Rahmen der Neuentwicklung von Mobile Enterprise Services?
- Welche Rahmenbedingungen sind spezifisch für den Bereich der mobilen Unternehmenslösungen (im Vergleich zu mobilen Konsumentenlösungen)?

Im Folgenden werden in Abschn. 12.1.1 und 12.1.2 basierend auf dem aktuellen Stand der wissenschaftlichen Literatur die Begriffe Mobile Enterprise Service und Service-Neuentwicklung (Fachbegriff „New Service Development“) definiert. In Abschn. 12.2 wird die der Studie zugrunde liegende Forschungsmethodik beschrieben. Die relevanten Studienergebnisse werden in Abschn. 12.3 detailliert wiedergegeben und in Abschn. 12.4 zusammengefasst und diskutiert.

### ***12.1.1 Mobile Enterprise Services***

Mobile Enterprise Services stellen zum einen spezifische Ausprägungen von IT-Dienstleistungen („IT-Services“) dar und zum anderen repräsentieren sie eine spezifische Klasse von Mobile Services.

Gemäß Wiggers et al. [1] ist ein IT-Service eine identifizierbare, messbare, bestellbare und verrechenbare Dienstleistung, die aus Kundensicht eine benötigte IT-Funktionalität erbringt. Diese Dienstleistung wird von der internen Unternehmens-IT zur Unterstützung der Geschäftsprozesse, über einen internen oder externen Dienstleister, bereitgestellt. IT-Services werden standardisiert oder individuell, auf Kundenspezifika angepasst, erbracht. Unterteilt werden sie in Infrastruktur-, Applikations- und Beratungsleistungen [1].

Losgelöst vom Unternehmensumfeld wird ein Mobile Service in der Literatur als eine Dienstleistung beschrieben, die über ein mobiles Netzwerk und heterogene mobile Endgeräte genutzt wird [2–4]. Karhu [2] legt den Fokus hierbei auf applikationszentrierte Mobile Services und klammert traditionelle mobile Sprach- und SMS-

Dienste bewusst aus. Gemäß Karhu [2] wird ein Mobile Service über die erbrachte Dienstleistung basierend auf einem Zusammenspiel von Applikation, Endgerät und Netzwerkinfrastruktur definiert.

Den Kern von mobilen Lösungen bilden Softwareapplikationen. Mobile Applikationen stellen die Schnittstelle zum Kunden dar und können „online“ (verbunden mit dem Netzwerk) oder „offline“ (getrennt vom Netzwerk) über Endgeräte wie Notebooks, PDAs oder Smartphones bereitgestellt werden [5]. Neuere Forschung legt den Fokus hierbei auf Smartphones als die primäre Endgeräte-Klasse zur Nutzung von mobilen Applikationen [2, 6–9].

Smartphones stellen die Konvergenz von PDA- und Telefonie-Endgeräten dar und agieren als Schnittstelle zur Nutzung von Kommunikationsdiensten sowie datenzentrierten Anwendungen [9]. Im Konsumentenbereich reicht die Funktionalität noch weiter als im Unternehmensumfeld und integriert Anwendungen zur Bereitstellung von Musik, Filmen oder Spielen [9].

Smartphone-Applikationen lassen sich in zwei Generationen unterteilen [9]. Applikationen der ersten Generation („first-generation“) sind u. a. mobile Telefonie, Kurznachrichten, die mobile Bereitstellung von E-Mail und Kalender-Daten oder die mobile Internet-Nutzung. Anwendungen der zweiten Generation („second-generation“) erfordern eine höhere Datenrate [9]. In diese Anwendungsklasse fallen Mobile-Enterprise-Lösungen, aber auch mobile Ticket-Applikationen, mobile soziale Netzwerke oder mobile Handelslösungen [9].

Die ersten Mobile-Enterprise-Anwendungen waren meistens transkodierte, mobile Versionen existierender Unternehmensapplikationen, die Zugang zum E-Mail-Postfach oder zu internen Datenbanken ermöglichen [10]. Neuere Mobile-Enterprise-Anwendungen („next-generation“) gehen hier einen Schritt weiter und machen i. d. R. Gebrauch von Sensoren-Daten (z. B. Lokationsinformationen über GPS) und erhöhen hierdurch den Automatisierungsgrad von Geschäftsprozessen [10].

Ein systematischer Ansatz zur Kategorisierung von Mobile Enterprise Services wird durch Basole [11] bereitgestellt. Er unterteilt in „Business-to-Commerce“ (B2C), „Business-to-Employee“ (B2E) und „Business-to-Business“ (B2B). Den Fokus seiner Studien legt er hierbei auf B2E und B2B und nennt die folgenden zentralen Mobile Enterprise Services [11]:

- Mobile Bereitstellung von E-Mail, Kalender und Kontakt-Daten („Mobile Office and Personal Information“)
- Mobile Einsicht auf Lagerbestände, Bestellung von Gütern oder Anstoßung von Rechnungszahlung („Mobile Enterprise Resource Planning“)
- Mobile Bereitstellung von Kundendaten („Mobile Customer Relationship Management“)
- Mobile Lager- und Bestandsverwaltung („Mobile Supply Chain Management“)
- Mobiler Austausch von Dokumenten und Wissen („Mobile Knowledge Management“)

Basole führt des Weiteren aus, dass Unternehmenslösungen zum einen taktisch definiert werden können, mit Fokus auf Produktivitätssteigerungen und Kostensenkung,

aber auch strategisch, mit Fokus auf die Generierung von Kernkompetenzen und die Erlangung von Wettbewerbsvorteilen.

Für die weiteren Ausführungen in diesem Beitrag und der zugrunde liegenden Studie wird ein Mobile Enterprise Service definiert als ein elektronischer Service (s. [12], der von einem Unternehmen, durch eine Softwareapplikation, über ein mobiles Netzwerk, auf einem Smartphone seinen Mitarbeitern bereitgestellt wird. Die Softwareapplikation kann eine native Applikation, J2ME-Applikation, aber auch ein mobiler Browser für den Zugriff auf eine mobile Web-Applikation sein. Bezogen auf die IT-Service-Definition liegt der Fokus im Rahmen dieser Studie auf der Bereitstellung von mobilen Applikationen und nicht auf Infrastruktur- oder Beratungsleistungen.

### ***12.1.2 New Service Development***

Mit New Service Development bzw. Service-Neuentwicklung wird in der Literatur die Entwicklung eines Angebotes bezeichnet, das für den Kunden in dieser Form noch nicht verfügbar war [13]. Dieses neue Service-Angebot kann aus der Kombination bestehender hervorgehen oder durch radikale oder inkrementelle Innovationen [13].

Eine alternative, ebenfalls weit verbreitete Bezeichnung für die Neuentwicklung von Services ist der Begriff „Service-Innovation“. Gemäß Schumpeter unterteilt man Service-Innovationen in Produkt- (z. B. neues Produkt), Prozess- (z. B. neuer Herstellungsprozess), Markt- (z. B. Erschließung eines neuen Marktes), Input- (z. B. Nutzung einer neuen Rohstoffbezugsquelle) und Organisationsinnovationen (z. B. Implementierung einer neuen Organisationsform) [14].

Eine Analyse diverser Ansätze zur Definition von New Service Development wurde von Tsai et al. [15] durchgeführt. Das Ergebnis dieser Analyse war die Synthese zu einem umfassenden und übergreifenden Ansatz, dem sog. „5-D Framework“. Tsai et al. [15] beschreibt hierbei folgende fünf Phasen der Service-Entwicklung:

1. Entdeckung und Generierung von neuen Ideen für zukünftige Service-Angebote („Discover“)
2. Entwicklung und Test von ausgewählten Service-Konzepten („Define“)
3. Detaillierte Spezifikation der Service-Funktionalitäten sowie der Produktionsysteme und -prozesse in einem Service-Design („Design“)
4. Pilotierung und Markteinführung von neuen Services („Deliver“)
5. Kontinuierliche Auswertung und Verbesserung der eingeführten Services („Debug“)

Das durch Tsai et al. [15] entwickelte Framework bildet die Basis für die in diesem Beitrag verwendete Abgrenzung und Definition der Projektphasen und Aktivitäten, die im Rahmen der Neuentwicklung von Mobile Enterprise Services durchlaufen

werden. Die Untersuchung von Neuentwicklungsaktivitäten basiert auf dem Rahmen, der hierdurch definiert wird.

## 12.2 Forschungsmethodik

Das Forschungsgebiet, das sich mit der Neuentwicklung von Mobile Enterprise Services beschäftigt, ist noch relativ jung, was sich auch durch die nur wenigen wissenschaftlichen Veröffentlichungen bemerkbar macht. Um dieses Thema zu erschließen wurde deshalb ein qualitativer Fallstudienansatz basierend auf Experteninterviews angewandt [16]. Für die Untersuchung von kausalen Beziehungen, wie es im Rahmen der Studie erfolgt, eignet sich die Fallstudien-Methodik in besonderem Maße [17, 18]. In einem iterativen Ansatz werden hier Daten gesammelt, analysiert sowie theoretische Zusammenhänge aufgestellt [16–18].

Die Fallstudien-Interviews wurden im Rahmen der Studie anhand der „Template-Analyse-Methode“ ausgewertet [2, 19]. Jede Fallstudie wurde initial einzeln kodiert. Die gesammelten Ergebnisse wurden dann in Summe analysiert und nach Mustern untersucht [16]. Durch die Anwendung sog. analytischer Generalisierungstechniken werden in der Studie Erkenntnisse gesammelt und der wissenschaftlichen Forschung zugefügt [18].

Zwischen Januar und März 2011 wurden sieben Fallstudien mit IT-Dienstleistern durchgeführt. Die untersuchten Unternehmen adressieren zum einen den deutschen IT-Dienstleistungsmarkt, betätigen sich aber auch international. Im Rahmen der Fallstudien wurde jeweils untersucht, wie einzelne IT-Dienstleister neue Mobile Enterprise Services entwickeln. Es wurde hinterfragt, welche Probleme hierbei auftraten und welche Maßnahmen hiergegen eingesetzt wurden. In jeder Fallstudie bzw. für jeden an der Studie teilnehmenden Dienstleister wurde stellvertretend ein Experte befragt. Diese Experten waren direkt verantwortlich für das Innovationsmanagement und die Neuentwicklung von Mobile Enterprise Services oder zu einem besonderen Maße hierin involviert. Neben der reinen Fallstudien-Betrachtung (retrospektiv) wurden die Experten gebeten zu beurteilen, ob die beschriebenen Herausforderungen und Maßnahmen im Konsumentenbereich stärker oder schwächer ausgeprägt sind. Diese Experteneinschätzung erlaubt eine erste Ableitung von Herausforderungen und Maßnahmen, die eine besondere Ausprägung im Unternehmensumfeld haben, und gibt Anhaltspunkte, ob es Unterschiede zwischen den beiden Märkten gibt.

Die Sammlung und Auswertung von Daten wechselte iterativ im Laufe der Fallstudien-Untersuchung. Zur Erstellung eines Interview-Leitfadens wurden, basierend auf einer initialen Literatur-Recherche, Pilot-Interviews mit Dienstleistern und potentiellen Kunden durchgeführt. Der Leitfaden wurde kontinuierlich, anhand der fortlaufenden Auswertung der Interviews erstellt und erweitert [16]. Anhand des Leitfadens wurden in den sieben Fallstudieninterviews Herausforderungen und eingesetzte Maßnahmen untersucht. Wurden neue Erkenntnisse gewonnen, wurde der Leitfaden angepasst. Für schon durchgeführte Fallstudien wurden diese Aspek-

te in kurzen Telefonaten oder per E-Mail nochmals hinterfragt. Um die Qualität der Fallstudien-Ergebnisse zu erhöhen wurde zusätzlich Dokumentation zu den Mobile-Enterprise-Service-Angeboten der Dienstleister ausgewertet.

Für die hier im Rahmen dieses Beitrags präsentierte Auswertung wird auf den Teil der Studie zurückgegriffen, in dem existierende Rahmenbedingungen untersucht wurden, die Herausforderungen, über eine erzeugte technologische oder marktseitige Unsicherheit, bedingen. Rahmenbedingungen wurden in Folge der Studienauswertung als Unsicherheit erzeugend den Analyseergebnissen hinzugefügt, wenn sie in mindestens zwei Fällen identifiziert wurden. Diese Einschränkung der Analyse dient der Erhöhung der Repräsentativität der Ergebnisse. Wurde ein kausaler Zusammenhang in allen untersuchten Fällen beobachtet, so ist dieser als repräsentativ für die untersuchten Dienstleister zu betrachten [17]. Dies erlaubt gemäß [18] die analytische Ableitung neuer wissenschaftlicher Theorie und unterstützt die externe Validität der Studienergebnisse. Eine statistische Verallgemeinerbarkeit auf die Gesamtpopulation ist aber aufgrund der Stichprobengröße nicht möglich. Neben der Überprüfung der externen Validität wurde für die Fallstudienuntersuchung interne Validität (durch Überprüfung von Mustern), Konstrukt-Validität (durch Nutzung mehrerer Datenquellen und Bildung einer Beweiskette) und Reliabilität (durch Fallstudiendatenbank und -protokoll) getestet und verifiziert.

Die Bewertung, ob eine Rahmenbedingung im Unternehmensumfeld spezifisch bzw. stärker als im Konsumentenmarkt ausgeprägt ist, erfolgte, wenn mindestens 50 % der Befragten (vier) dies so annahmen.

Gemäß Eisenhardt (1989) müssen keine weiteren Fälle einer Fallstudien-Untersuchung hinzugefügt werden, wenn die sog. „Saturation“ erreicht wurde. Dies ist der Fall, wenn durch die Hinzunahme weiterer Fälle keine neuen Erkenntnisse gewonnen werden können [16]. Für die Identifikation von technologischen und marktseitigen Rahmenbedingungen (mind. zweimal identifiziert) wurde die Saturation nach der dritten Fallstudie erreicht.

## 12.3 Fallstudienergebnisse

Als Basis für die folgende Analyse wird ein Ausschnitt der durchgeföhrten Fallstudien und deren Auswertung genutzt. Ziel ist es, qualitative Erkenntnisse über technologische und marktseitige Rahmenbedingungen, die einen Einfluss auf die Neuentwicklung von Mobile Enterprise Services haben, zu erlangen. Es wird hier jeweils untersucht, welche Rahmenbedingungen technologische oder marktseitige Unsicherheit für den jeweiligen IT-Dienstleister erzeugt haben.

Das Konzept der Unsicherheit, welches die Basis für die Untersuchung darstellt, wurde von Van den Ende [20] beschrieben. Als technologische Unsicherheit erzeugend werden technische Probleme, benötigtes technisches Wissen und erforderliche Anpassungen des Entwicklungsprozesses beschrieben [20]. Marktseitige Unsicherheit wird durch das Verhalten von Kunden und Konurrenten sowie den existierenden Substituten bedingt [20].

### **12.3.1 Technologische Unsicherheit**

Alle Dienstleister berichteten, dass die heterogene, sich schnell ändernde mobile Infrastruktur, hohe Sicherheitsanforderungen und die Notwendigkeit der Integration in Unternehmensinfrastruktur technologische Unsicherheit erzeugten. In fast allen Fallstudien konnte darüber hinaus ein Mangel an Ressourcen und Know-how wie auch der fehlende Standard für mobile Applikationen als Unsicherheit verursachend identifiziert werden (siehe Tabelle 12.1).

Betrachtet man die Einschätzung der Ausprägung der aufgeführten Rahmenbedingungen auf dem Konsumentenmarkt, so ist eine dominantere Ausprägung der technischen Heterogenität und Änderungsrate hier anzunehmen. Hohe Sicherheitsanforderungen, Ressourcen- und Know-how-Mangel und die Notwendigkeit der Integration in Unternehmensinfrastruktur hingegen wurden als relevanter und ausgeprägter im Unternehmensumfeld eingestuft. Unterschiedliche Auswirkungen der Abstinenz eines mobilen Applikationsstandards auf Konsumenten- und Unternehmensumfeld konnten basierend auf den Studienergebnissen nicht abgeleitet werden.

In Tabelle 12.1 wird die Häufigkeit der Nennung der einzelnen Rahmenbedingungen in den einzelnen Fallstudien dargestellt. Im Folgenden werden qualitative Erkenntnisse zu den einzelnen Rahmenbedingungen, die in der Fallstudienauswertung gesammelt wurden, kurz zusammengefasst.

#### **Hohe Heterogenität der mobilen Infrastruktur**

Die untersuchten Fälle zeigen eine hohe Heterogenität von mobilen Endgeräten (mit unterschiedlicher Displaygröße und -auflösung) und Betriebssystemen bei Unternehmuskunden. Diese Heterogenität wird, wegen der Notwendigkeit der Integration von privaten Mitarbeiter-Endgeräten und den aus dem privaten Gebrauch kommenden Anforderungen an betriebliche Endgeräte (Consumerization), gemäß der Erwartung der befragten Experten, weiter steigen. Beispielhaft berichteten die

**Tabelle 12.1** Technologische Unsicherheit erzeugende Rahmenbedingungen

Technologische Rahmenbedingungen	Nennungen	Ausprägung im Vergleich zum Konsumentenmarkt
Hohe Heterogenität der mobilen Infrastruktur	7	Schwächer
Hohe Änderungsrate der mobilen Infrastruktur	7	Schwächer
Fehlender Standard für mobile Applikationen	5	?
Hohe Sicherheitsanforderungen	7	Stärker
Ressourcen- und Know-how-Mangel	6	Stärker
Integration in Unternehmensinfrastruktur	7	Stärker

Dienstleister von Unternehmenskunden, die es nicht schafften die Vielfalt der unterstützten mobilen Endgeräte zu limitieren und sich nun zunehmend öffnen. Eine Notwendigkeit, die als hieraus resultierend dargestellt wurde, ist der Bedarf nach Mobile-Device-Management-Lösungen, zur sicheren Administration der Geräte-Vielfalt. Für den Konsumentenmarkt wurde durch einen Experten die Möglichkeit der Fokussierung der Anbieter auf wenige, dominante Plattformen wie iOS und Android gesehen. Eine solche Möglichkeit besteht im Enterprise-Umfeld nur sehr eingeschränkt, da man als Anbieter die Plattformen, die bei den Kunden im Einsatz sind, unterstützen muss.

### **Hohe Änderungsrate der mobilen Infrastruktur**

Neben der hohen Heterogenität wurde auch eine hohe Änderungsrate für die mobilen Endgeräte und Betriebssysteme der Unternehmenskunden berichtet. Es wurde erwähnt, dass Unternehmen bestrebt sind, die Nutzungsdauer von mobilen Endgeräten auszudehnen. Aufgrund der Erwartung der Mitarbeiter (speziell der oberen Managementebenen), dass der Arbeitgeber innovative Endgeräte bereitstellen muss, droht dies in der Regel zu scheitern. Die durch die Experten geschilderte Erwartungshaltung zeigt einen Bedarf an betrieblichen Endgeräten, die den privat genutzten ebenbürtig sein sollen. Gemäß der Einschätzung der Dienstleister wird eine Ausdehnung der Nutzungsdauer zukünftig schwer möglich sein.

### **Fehlender Standard für mobile Applikationen**

Die fehlende Unterstützung von J2ME durch die Betriebssysteme iOS und Android hat einen einst etablierten mobilen Applikationsstandard ad absurdum geführt. Apple und Google verfolgen proprietäre Entwicklungsansätze, wodurch die Bedeutung von Java-basierten J2ME-Applikationen stark abnimmt. Ein unterschiedlich starker Einfluss dieser Rahmenbedingung auf den Konsumenten- oder Unternehmens-Bereich konnte aus den Experteneinschätzungen nicht abgeleitet werden. Eine Tatsache, die genannt wurde, die die Auswirkung dieser Rahmenbedingung auf den Konsumentenmarkt abschwächen könnte, ist die Möglichkeit der Fokussierung auf dominante Plattformen. Darüber hinaus könnte die Nutzung von HTML5 zur Bereitstellung mobiler Applikationen mittelfristig einen neuen Standard setzen, da einzelne Anbieter positive Ergebnisse erster Pilotversuche berichteten.

### **Hohe Sicherheitsanforderungen**

Sicherheitsanforderungen sind typischerweise im Unternehmensumfeld stärker ausgeprägt. Dies zeigt auch die Einschätzung der befragten Experten. Nichtsdestoweniger wurde eine steigende Bedeutung von Sicherheitsaspekten im Konsumentenmarkt gesehen. Eine Sicherheitsmaßnahme, implementiert seitens der Dienstleis-

ter, die im Rahmen der Studie beispielhaft genannt wurde, ist die Vergabe von Sicherheitszertifikaten für Partnerprodukte. Als kritisch für die Bereitstellung einer hohen mobilen Sicherheit in Unternehmen wurde das teilweise Fehlen von Sicherheitsrichtlinien („Policies“) gesehen. Darüber hinaus stellen der ansteigende Consumerization-Trend und die daraus resultierende Notwendigkeit der Integration privater Endgeräte ein zunehmendes Sicherheitsrisiko dar.

### Ressourcen- und Know-how-Mangel

Die an der Studie teilnehmenden IT-Dienstleister berichteten überwiegend, nicht ausreichend Ressourcen und Know-how zur Verfügung zu haben, um mit der hohen Heterogenität und Änderungsrate umgehen zu können. Die aktuell steigende Nachfrage nach mobilen Unternehmenslösungen könnte diesen Mangel weiter verstärken. Der Anbieter, der gemäß der Interviewergebnisse nicht von diesem Mangel betroffen ist, berichtete von speziellen Maßnahmen des Know-how-Aufbaus. Durch die Akquise von Start-Ups und die Nutzung von eigenen Offshore-Kapazitäten wurde hier dieser Mangel beseitigt. Für den Konsumentenmarkt wurde ein Ressourcen- oder Know-how-Problem weniger gesehen, da hier wiederum die Möglichkeit der Fokussierung auf dominante Plattformen besteht.

### Integration in die Unternehmensinfrastruktur

Die Integration von Mobile Services in die Unternehmensinfrastruktur wurde als eine Rahmenbedingung beschrieben, die überwiegend Anbieter im Enterprise-Umfeld bewältigen müssen. Heterogene Backend-Strukturen für z.B. CRM- oder ERP-Systeme stellen eine spezifische Herausforderung für die befragten IT-Dienstleister dar. Beispielsweise wurde berichtet, dass Mobile-Middleware-Konzepte (z.B. Sybase Unwired Platform) eingesetzt werden, um über standardisierte Schnittstellen eine Backend-Integration zu realisieren. Neben der Notwendigkeit der Backend-Anbindung zeigen die Studienergebnisse einen Bedarf der Integration der mobilen Unternehmenslösungen in bestehende Geschäftsprozesse, die als nicht für eine Mobilisierung konzipiert beschrieben wurden.

#### 12.3.2 Marktseitige Unsicherheit

Rahmenbedingungen, die als marktseitige Unsicherheit erzeugend dargestellt wurden, sind in Tabelle 12.2 aufgeführt. Alle Interview-Partner sahen hier ein starkes Marktwachstum sowie einen anhaltenden Consumerization-Trend bei ihren Unternehmenskunden. Des Weiteren wurden heterogene, sich schnell ändernde Kundenanforderungen, der kontinuierliche Eintritt neuer Spieler sowie Service-Angebote, die sich zu Beginn des Lebenszyklus befinden, von den meisten Teil-

**Tabelle 12.2** Marktseitige Unsicherheit erzeugende Rahmenbedingungen

Marktseitige Rahmenbedingungen	Nennungen	Ausprägung im Vergleich zum Konsumentenmarkt
Hohe Heterogenität der Kundenanforderungen	6	Schwächer
Hohe Änderungsrate der Kundenanforderungen	5	Schwächer
Eintritt neuer Spieler in den Markt	6	Schwächer
Hoher Kostendruck auf Service-Neuentwicklung	2	Schwächer
Service-Angebote noch jung	6	Stärker
Starkes Marktwachstum	7	Kein Unterschied
Consumerization des Unternehmensumfelds	7	Nur Unternehmensumfeld
Intoleranz für Service-Instabilität	3	Stärker

nehmern genannt. Weniger verbreitet und daher auch wohl mit geringerem Einfluss auf die marktseitige Unsicherheit scheinen der hohe Kostendruck auf die Service-Neuentwicklung und Intoleranzen für Service-Instabilitäten zu sein.

Eine weitere diskutierte, aber nicht aufgeführte Rahmenbedingung, da nur von einem Dienstleister genannt, ist aggressives Konkurrenzverhalten. Aggressive Markteintrittsstrategien wurden aber verstärkt im Konsumentenmarkt gesehen; Eintritt neuer Spieler, hoher Kostendruck sowie Heterogenität und Kurzlebigkeit von Kundenanforderungen ebenso. Als schwächer ausgeprägt im Konsumentenumfeld wurden Intoleranz für Service-Instabilitäten und die Existenz von relativ jungen Service-Angeboten eingestuft. Der Consumerization-Trend ist per Definition ein Unternehmensphänomen. Keinen Unterschied zwischen den Märkten sah man im Mobile-Service-Marktwachstum.

Im Folgenden werden weitere qualitative Erkenntnisse zu den einzelnen untersuchten Rahmenbedingungen zusammengefasst dargestellt.

### Hohe Heterogenität der Kundenanforderungen

Die Mobile-Service-Angebote der Anbieter stellten sich als überwiegend individuell auf das Geschäft der Kunden zugeschnitten dar. Für den Kern der Services wurden aber homogene Kernanforderungen gesehen, die sich bei einem vergleichbaren Funktionsumfang ähneln. Ein starker Einfluss aus dem Konsumentenbereich auf die betrieblichen Anforderungen wurde auch hier genannt. Gründe für den Einsatz von Mobile Enterprise Services, die von den Dienstleistern aufgeführt wurden, variieren von Produktivitätssteigerung, Mobilitätsunterstützung bis zur Betriebskosten senkung. Mit der wachsenden Produktreife betrieblicher Mobile Services wird eine noch stärkere heterogene Ausgestaltung der Kundenanforderungen durch die befragten Experten erwartet.

## **Hohe Änderungsrate der Kundenanforderungen**

Die Anforderungen der betrieblichen Kunden konnten im Rahmen der Studie als sich schnell ändernd identifiziert werden. Der steigende Bedarf nach Mobilität und Flexibilität, die noch jungen Service-Angebote sowie ein immer stärkerer Consumerization-Trend scheinen dies zu beschleunigen. Im Vergleich zum Konsumentenmarkt wurde diese Änderungsgeschwindigkeit trotzdem als moderater eingestuft, beeinflusst u. a. durch striktere Sicherheitsanforderungen im betrieblichen Umfeld.

## **Eintritt neuer Spieler in den Markt**

Fast alle an der Studie teilnehmenden Dienstleister berichteten von einer Vielzahl neuer Spieler, die in den Mobile-Enterprise-Markt aktuell eintreten. Eine größere Fokussierung auf wenige Services im Produktkatalog wurde als möglicher Grund, hiervon im eigenen Umfeld nicht betroffen zu sein, durch einen Studienteilnehmer genannt. Generell wurden neue Marktteilnehmer überwiegend als mögliche Partner und weniger als Konkurrenten gesehen. Neue Spieler auf dem mobilen Konsumentenmarkt wurden als noch heterogener als im Unternehmensumfeld eingestuft. Speziell Firmen und Start-Ups aus dem Online-Gaming- und Web 2.0-Bereich wurden hier genannt.

## **Hoher Kostendruck auf Service-Neuentwicklung**

Gemäß den Studienergebnissen empfanden nur wenige der befragten Teilnehmer einen hohen Kostendruck bei der Service-Neuentwicklung. Einige Dienstleister berichteten sogar von einer gestiegenen Investitionsbereitschaft bei ihren Kunden. Interessanterweise gaben aber einige befragte Dienstleister an, sich trotz Abstinenz eines Kostendrucks bei der Service-Neuentwicklung stark auf eine Kostenreduktion zu fokussieren, um einem möglichen Kostendruck aktiv zu begegnen. Im Konsumentenmarkt für Mobile Services wurde dieser Druck als deutlich ausgeprägter eingeschätzt. Dies wurde begründet mit der Erwartung der Konsumenten, dass Online Services kostenlos oder sehr günstig bereitgestellt werden müssen, und der höheren Produktreife mobiler Konsumentenservices.

## **Service-Angebote noch jung**

Die meisten Studienteilnehmer stuften ihre eigenen Mobile-Enterprise-Service-Angebote als relativ jung ein. Dies wurde für den gesamten mobilen Unternehmensmarkt, verglichen mit dem Konsumentenmarkt, so gesehen. Services wie z. B. mobile soziale Netzwerke haben schon eine relativ hohe Marktdurchdringung auf dem Konsumentenmarkt erreicht und wurden deshalb als reifer eingestuft.

## **Starkes Marktwachstum**

Das existierende starke Marktwachstum im betrieblichen Mobile-Service-Bereich wurde durch die Studienteilnehmer bestätigt. Es wurde angenommen, dass sich dieses weiter fortsetzen wird. Ähnlich wurde die Situation im Konsumentenmarkt eingeschätzt. Unterschiede konnten hier nicht benannt werden.

## **Consumerization des Unternehmensumfelds**

Eine starke Beeinflussung der Anforderungen betrieblicher Nutzer durch ihre privaten Erfahrungen und Gewohnheiten im Umgang mit mobilen Technologien wurde durch die Studienteilnehmer dargestellt und betont. Hieraus resultierten in den untersuchten Fällen u. a. die private Nutzung von betrieblichen Endgeräten und Anwendungen, die Notwendigkeit der betrieblichen Integration privater Endgeräte und besonders die betriebliche Nachfrage nach dem iPhone. Nachdem Unternehmen in der Vergangenheit immer versucht hatten, ähnliche Tendenzen zu blockieren, berichteten die Studienteilnehmer von einer immer größeren Bereitschaft ihrer Geschäftskunden sich hierfür zu öffnen. Aspekte wie eine Verbesserung der „Work-Life-Balance“ z. B. durch die Verschmelzung von privatem und geschäftlichem Endgerät oder eine höhere Mitarbeiterzufriedenheit wurden als Motivationsgründe genannt.

## **Intoleranz für Service-Instabilität**

Nur wenige IT-Dienstleister nannten Intoleranz für Mobile-Service-Instabilität (z. B. Verbindungsabbrüche oder längere Reaktionszeiten), bei ihren Kunden als eine Rahmenbedingung, die eine marktseitige Unsicherheit erzeugt hat (verglichen mit anderen IT-Services). Der Grund hierfür könnte in der Vertrautheit von Nutzern mit den generellen Einschränkungen und Instabilitäten mobiler Services sein. Auf der anderen Seite wurde die Nutzung betrieblicher mobiler Services im beruflichen Alltag als ein möglicher Grund für eine mögliche höhere Erwartungshaltung gesehen. Die höhere Erwartungshaltung an die betriebliche Stabilität von mobilen Lösungen zeigt sich u. a. in den existierenden SLAs (Dienstgütevereinbarungen) im Unternehmensumfeld. Die Tatsache, dass mobile Konsumentenservices kostenlos bzw. relativ günstig erhältlich sind, wurde als wesentlicher Aspekt für eine höhere Fehlertoleranz bei Privatkunden benannt. Trotzdem wurde durch einen Experten beispielhaft erwähnt, dass speziell im Konsumentenmarkt der erste Eindruck einer mobilen Applikation entscheidend für die weitere Nutzung sein kann.

## 12.4 Zusammenfassung und Diskussion

Die Analyse der Studienergebnisse nach Rahmenbedingungen, die technologische und marktseitige Unsicherheit bei der Neuentwicklung von Mobile Enterprise Services erzeugen, ist in Kapitel 3 dargestellt. Anhand der durchgeföhrten Fallstudien können diverse Rahmenbedingungen identifiziert werden, die potentiell eine Unsicherheit für die Neuentwicklungsaktivitäten kreieren. Untersucht wurde hierbei, ob die Rahmenbedingungen im Umfeld der teilnehmenden Dienstleister existierten und ob sie für diese eine Unsicherheit erzeugt haben.

In allen Fallstudien wurden die heterogene, sich schnell ändernde mobile Infrastruktur, hohe Sicherheitsanforderungen und die Notwendigkeit der Integration in Unternehmensinfrastruktur als technologische Unsicherheit bedingende Rahmenbedingungen genannt. Im Bereich der marktseitigen Unsicherheit waren es das starke Marktwachstum und die Consumerization des Unternehmensumfelds.

Die hier zusammengefassten Implikationen zwischen existierenden Rahmenbedingungen und Unsicherheit im Rahmen der Service-Neuentwicklung wurden aus der Theorie abgeleitet und konnten in allen Fällen repliziert werden. Dies erlaubt gemäß [18] die analytische Ableitung neuer wissenschaftlicher Theorie und unterstützt die externe Validität der Studienergebnisse. Aufgrund der zu geringen Stichprobengröße der durchgeföhrten Studie können diese kausalen Zusammenhänge statistisch nicht auf die Grundgesamtheit der IT-Dienstleister übertragen werden [18]. Dies ist in weiteren Untersuchungen zu prüfen (z. B. anhand von Strukturgleichungsmodellierung).

Basierend auf der Experteneinschätzung der befragten Dienstleister wurden in der Studie darüber hinaus Rahmenbedingungen identifiziert, für die ausgeprägte Spezifika auf dem Enterprise-Markt existieren. Diese Rahmenbedingungen sind die hohen Sicherheitsanforderungen, die Notwendigkeit der Integration in Unternehmensinfrastruktur sowie der Consumerization-Trend. Ein Mangel an Ressourcen, potentiell unausgereifte Service-Angebote und Intoleranz für Service-Instabilität wurden ebenso als ausgeprägter im Unternehmensumfeld eingestuft. Diese Rahmenbedingungen konnten aber nicht in allen Fällen repliziert werden.

Weitere Aspekte, die im Rahmen der zugrunde liegenden Fallstudienauswertung betrachtet, aber für diese Untersuchung nicht berücksichtigt wurden, sind die aus der Unsicherheit resultierenden Herausforderungen sowie die hiergegen eingesetzten Maßnahmen. Diese Aspekte werden in weiterführenden Artikeln untersucht und dargestellt.

## Literaturverzeichnis

1. Wiggers P, Kok H, de Boer-de Wit M (2004) IT performance management. Butterworth-Heinemann Publications, Oxford, Burlington
2. Karhu P (2007) Emerging mobile service innovation markets: The case of the Finnish mobile TV service market. Dissertation Universität St. Gallen. [www1.unisg.ch/www/edis.nsf/.../3297/protectT1\textdollarFILE/dis3297.pdf](http://www1.unisg.ch/www/edis.nsf/.../3297/protectT1\textdollarFILE/dis3297.pdf). Accessed 18 June 2011
3. Lee H, Kim C, Park Y (2008) Evaluation and management of new service concepts: An ANP-based portfolio approach. *Computers & Industrial Engineering* H58(2010):535–543
4. van de Kar E, Muniaf P, Wang Y (2006) Mobile services used in unstable environments: Design requirements based on three case studies. International Conference on Entertainment Computing
5. Kalakota R (2004) Mobile enterprise applications: Deploying second-generation solutions. Intel/SAP Executive Brief
6. Issel K, Mrozik J (2008) A mobile data service framework and its business models. Proceedings of International Conference on Advances in Mobile Computing and Multimedia
7. Jiang XN, Yung KL (2010) A new intelligent system for senior executives to maintain remote control of their company. *Expert Systems with Applications* H38(2011):736–742
8. Martinez M (2010) The mobile phone user: Identifying top mobile applications. SSRN. <http://ssrn.com/abstract=1691210>. Accessed 18 June 2011.
9. Zheng P, Ni L (2006) The next wave of computing. In: Smart phone and next generation mobile computing. Morgan Kaufmann Publishers, San Francisco
10. Basole R (2005) Mobilizing the enterprise: A conceptual model of transformational value and enterprise readiness. Proceedings of ASEM National Conference
11. Basole R (2007) The emergence of the mobile enterprise: A value-driven perspective. International Conference on the Management of Mobile Business
12. Riedl C, Leimeister J M, Kremar H (2010) Why e-service development is different: A literature review. *e-Service J*
13. Johnson P, Menor L, Roth A, Chase R (2000) A critical evaluation of the new service development process. In: Fitzsimmons J, Fitzsimmons M (ed) New service development: Creating memorable experiences. Sage Publications, London, Thousand Oaks, New Delhi
14. Schumpeter J A (1934) The theory of economic development: An inquiry into profits, capital, credit, interest, and the business cycle. Harvard University Press, Cambridge, Massachusetts
15. Tsai W, Verma R, Schmidt G (2008) New service development. In: Loch C, Kavadias S (ed) Handbook of new product development management, 1<sup>st</sup> edn, Butterworth-Heinemann Publications, Oxford, Burlington
16. Eisenhardt K (1989) Building theories from case study research. *Academy of Management Review* H14(1989):532–550
17. Gläser J, Laudel G (2010) Experteninterviews und qualitative Inhaltsanalyse, 4. Aufl. VS Verlag für Sozialwissenschaften, Wiesbaden
18. Yin RK (2009) Case study research: Design and methods, 4<sup>th</sup> edn, Sage Publications, London, Thousand Oaks, New Delhi, Singapore
19. King N (1998) Template analysis. In: Symon G, Cassell C (ed) Qualitative methods and analysis in organizational research: A Practical Guide. Sage Publications, London, Thousand Oaks, New Delhi
20. van den Ende J (2002) Modes of governance of new service development for mobile networks: A life cycle perspective. *Research policy* H32(2003):1501–1518
21. Basole R (2008) Enterprise mobility: researching a new paradigm. *Information Knowledge Systems Management* H7(2008):1–7

**Teil V**

**Technologische Sicht: Sicherheit,  
Datenschutz & Privacy**

---

# Kapitel 13

## Smart Mobile Apps: Enabler oder Risiko?

Claudia Eckert und Christian Schneider

**Zusammenfassung** Kennzeichnend für Smart Mobile Apps ist ihre Eigenschaft, Dienste zu nutzen, die über das Internet oder ein privates Netzwerk bereitgestellt werden. Viele dieser Dienste werden in einer Cloud betrieben und ermöglichen es dem Nutzer, einen konsistenten Datenbestand auf unterschiedlichen mobilen und stationären Geräten zu führen. Eine Smart Mobile App stellt dabei die Client-Seite eines Cloud-Dienstes dar, so dass es für den Nutzer keine Rolle mehr spielt, von wo und mit welchem Gerät der Zugriff auf seine Daten erfolgt. Mit diesen Eigenschaften werden Smart Mobile Apps zunehmend auch für den Geschäftsbereich attraktiv. Als Business-Apps werden sie zu einem integralen Bestandteil von Geschäftsprozessen. Die Frage der Datensicherheit bei der Nutzung von Smart Mobile Apps tritt somit spätestens bei deren flächendeckender Nutzung in geschäftlicher Umgebung zu Tage. Um die Frage zu beantworten, ob Smart Mobile Apps in der heutigen Zeit als reine Business-Enabler zu betrachten sind, oder ob damit in erster Linie neue Risiken verbunden sind, wird zunächst die Bedrohungs- und Risikolandschaft bei der Nutzung von Smart Mobile Apps aufgezeigt. Den dargestellten Sicherheitsbedrohungen werden anschließend Lösungsansätze gegenübergestellt, wie sie heutige im Einsatz befindliche Systeme anbieten. Es wird deutlich, dass es noch keine zufriedenstellenden Ansätze gibt, mit denen die im Bedrohungsteil aufgezeigten Probleme überzeugend gelöst werden können. Wir zeigen deshalb abschließend erfolgversprechende Arbeiten aus der Forschung auf, deren konsequente Weiterverfolgung und Umsetzung in die Praxis dazu beitragen würden, Smart Mobile Apps zu wirklichen Enablers im Future Internet werden zu lassen und die dabei auftretenden Risiken zu minimieren.

---

Claudia Eckert

Fraunhofer AISEC, Garching bei München, E-mail: eckert@sec.in.tum.de  
Technische Universität München, Lehrstuhl für Sicherheit in der Informatik,  
Garching bei München, E-mail: eckert@sec.in.tum.de

Christian Schneider

TU München, Fakultät für Informatik, Lehrstuhl für Sicherheit in der Informatik,  
Garching bei München, E-mail: chrschn@sec.in.tum.de

## 13.1 Einleitung

Apps sind bereits heute im privaten Umfeld, im Consumer-Bereich sehr beliebt und weit verbreitet. Auch wenn derzeit noch ein großer Teil der Apps als rein lokale Anwendungen auf mobilen Geräten ausgeführt wird, so ist die Entwicklung hin zu Smarten Mobilen Apps deutlich zu sehen. Kennzeichnend für diese fortgeschrittenen Apps ist ihre Eigenschaft, Dienste zu nutzen, die über das Internet oder ein privates Netzwerk bereitgestellt werden. Viele dieser Dienste werden in einer Cloud betrieben und ermöglichen es dem Nutzer somit, einen konsistenten Datenbestand auf unterschiedlichen mobilen und stationären Geräten zu führen. Eine Smart Mobile App stellt dabei die Client-Seite eines Cloud-Dienstes dar, eine Art „Fenster in die Cloud“, so dass es für den Nutzer keine Rolle mehr spielt, von wo und mit welchem Gerät der Zugriff auf seine Daten erfolgt.

Mit diesen Eigenschaften werden Smart Mobile Apps zunehmend auch für den Geschäftsbereich attraktiv. Als Business-Apps werden sie zu einem integralen Bestandteil von Geschäftsprozessen, die u. a. unternehmenskritische Prozesse bearbeiten oder auf kritische Geschäftsdaten in privaten, hybriden oder Community-Clouds zugreifen. Die Vielzahl der Geräte, auf denen die mobilen Apps zur Ausführung kommen können und über die auf die Daten in der Cloud zugegriffen werden kann, deckt ein breites Spektrum ab, so dass die Daten potentiell jederzeit, von beliebigen Geräten zugreifbar sind. Eine Smart Mobile App ist auf die jeweilige Gerätekasse zugeschnitten, so dass die Benutzerschnittstelle und Repräsentation der Daten den Möglichkeiten des Geräts gerecht wird. Auch dies ist für einen breitflächigen Einsatz der Smarten Apps insbesondere im Geschäftsumfeld ein wichtiger Erfolgsfaktor.

Die Frage der Datensicherheit bei der Nutzung von Smart Mobile Apps tritt somit spätestens bei deren flächendeckender Nutzung in geschäftlicher Umgebung zu Tage. Jedoch sind auch im Consumer-Umfeld viele sicherheitskritische Aktivitäten von der Nutzung von Mobilen Apps betroffen. Smart Mobile Apps sind in besonderem Maße anfällig für Angriffe [1], die darauf abzielen, Daten auszuspähen, sich unter einer falschen Identität Zugriffe auf Daten zu verschaffen, Daten zu manipulieren oder die Verfügbarkeit von Daten-Diensten zu behindern. Im Zusammenhang mit der Nutzung mobiler Endgeräte spielen Geodaten und deren Sicherheitsprobleme eine bedeutende Rolle. Lokalisierungsdienste eröffnen lukrative Geschäftsmodelle über ortsbasierte Dienste, wie etwa das Finden des nächsten Hotels, die Abfahrtszeiten am lokalen Bahnhof oder auch Googles „Suchen in der Nähe“. Gleichzeitig können aber die Geo-Daten, die zur Positionsbestimmung Dritten zur Verfügung gestellt werden, oder aber auch die Analyse der Suchbegriffe und der Ableitung der sich daraus ergebenden Präferenzen des Nutzers, zu erheblichen Verletzungen der Privatsphäre und des Datenschutzes führen. Damit wären sowohl die Nutzer als direkt Betroffene als auch die Anbieter, die gegebenenfalls Datenschutzauflagen des Gesetzgebers missachten, von solchen Sicherheitsproblemen betroffen.

Um die Frage zu beantworten, ob Smart Mobile Apps in der heutigen Zeit als reine Business-Enabler zu betrachten sind, oder ob damit in erster Linie neue, noch nicht zufriedenstellend beherrschte Risiken verbunden sind, gehen wir zunächst etwas detaillierter auf die Bedrohungslage ein, wobei wir uns auf die Betrachtung der Client-Seite, also des mobilen Endgeräts beschränken. Cloud-basierte Dienste weisen ihrerseits eine Vielzahl von Sicherheitsbedrohungen auf (siehe [2]), deren Betrachtung jedoch den Rahmen des vorliegenden Beitrags sprengen würde. Ein ausführlicher Blick auf vorhandene und in Produkten umgesetzte Lösungsansätze zeigt, dass es noch keine zufriedenstellenden Ansätze gibt, mit denen die im Bedrohungsteil aufgezeigten Probleme überzeugend gelöst werden können. Wir zeigen deshalb abschließend erfolgversprechende Arbeiten aus der Forschung auf, deren konsequente Weiterverfolgung und Umsetzung in die Praxis dazu beitragen würden, Smart Mobile Apps zu wirklichen Enablers im Future Internet werden zu lassen und die dabei auftretenden Risiken zu minimieren.

## 13.2 Sicherheitsbedrohungen für Smart Mobile Apps

Smartphones und Tablets werden von Nutzern zunehmend als eine Art persönliche Informationszentrale verwendet. Die Geräte sind „always on“ und werden von vielen Benutzern stets mitgeführt. Dabei steigt die Bedrohungslage für (oder auch durch) Smart Mobile Apps im gleichen Maße wie die Menge und Relevanz der damit verarbeiteten Daten wächst. Um die mit Mobilen Apps verbundenen Risiken besser einschätzen zu können, ist eine Analyse der Bedrohungen aus verschiedenen Blickwinkeln und für unterschiedliche Bereiche sinnvoll.

Von zentraler Bedeutung für die Sicherheit der Anwendungen, und damit natürlich auch der mobilen Apps, ist die Systemsicherheit der mobilen Plattform, auf der die Anwendungen zur Ausführung gelangen. Hierbei geht es um die Frage, inwieweit das mobile Gerät als Kombination aus Hardware und mobilem Betriebssystem eine solide Basis stellt, um einen angemessenen Schutz für die darauf laufenden Smart Mobile Apps zu gewährleisten.

Der zweite sicherheitskritische Bereich wird durch die Anwendungsebene definiert. Sie bildet die direkte Ausführungsumgebung für die Smart Mobile Apps. Hier gilt es einerseits zu hinterfragen, wie aus Sicht der Ausführungsumgebung der mobilen App vertraut werden kann. Das heißt, es sind die Bedrohungen zu analysieren, die sich für die Plattform und damit für den Nutzer und seine Umgebung ergeben können, wenn die mobile App Schadsoftware beinhaltet oder Sicherheitsschwachstellen aufweist, die als Einfallstore auf die mobile Plattform missbraucht werden könnten. Andererseits sind auch die App selbst sowie ihre Daten vielfältigen Bedrohungen durch eine unsichere Plattform oder möglicherweise andere, bösartige Apps ausgesetzt, die gleichzeitig in der Anwendungsebene zur Ausführung gelangen.

Als dritten sicherheitskritischen Aspekt betrachten wir den Bereich der Datensicherheit. Hierbei geht es nicht nur um die Problematik des möglichen Datenverlustes beim Diebstahl eines mobilen Geräts, sondern auch um die mögliche Gefähr-

dung der Privatsphäre durch die Aggregation persönlicher Daten und die Erstellung von Bewegungsprofilen durch ortsbasierte Dienste. Diese aufgezeigten drei Bereiche werden im Folgenden detailliert betrachtet.

### ***13.2.1 Bedrohungen der mobilen Plattformen***

Mobile Plattformen unterliegen einer höheren Gefährdung als stationäre. Sie werden an verschiedenen Orten und damit in unterschiedlich vertrauenswürdigen Umgebungen eingesetzt. Sie gehen schneller verloren oder werden gestohlen als etwa ein PC. Gleichzeitig kann sich ein Unbefugter Zugang zu einem mobilen Gerät verschaffen, wenn es einen Moment unbeaufsichtigt ist. Falls keine entsprechenden Gegenmaßnahmen getroffen wurden, sind in diesen Fällen alle Daten gefährdet, die über die Apps auf dem Gerät zugreifbar sind, gleichgültig ob diese lokal gespeichert sind oder über Cloud-Dienste erreicht werden.

Die kritischste Bedrohung für mobile Plattformen ist jedoch, dass solche personalisierten Geräte oftmals nicht oder nur unzureichend von einer unternehmensweiten Sicherheitspolitik erfasst werden. Meistens werden sie von den Benutzern selbst administriert. Nur selten besteht die Möglichkeit der Fernwartung durch die IT-Abteilung. Die Erfahrung aus der Welt der Desktop-Betriebssysteme lehrt jedoch, dass noch immer vielen Anwendern ein hinreichendes Sicherheitsbewusstsein fehlt: Zugriffssperren und Passwortschutz werden deaktiviert, da sie als störend empfunden werden, Sicherheitswarnungen werden ignoriert, Software-Updates werden, wenn überhaupt, sehr unregelmäßig eingespielt. In solchen Fällen vergrößert sich entsprechend das bereits vorhandene Bedrohungspotential mit einer unnötig großen Angriffsfläche der mobilen Plattform.

Für die Verbesserungswürdige Update-Politik ist aber nicht alleine die Nachlässigkeit der Anwender verantwortlich. Die Hersteller der Betriebssysteme selbst stehen hier in der Verantwortung, sichere, einfach zu nutzende Update-Mechanismen zur Verfügung zu stellen, bzw. diese direkt in die Plattformen zu integrieren. Hier sind derzeit noch erhebliche Mängel vorhanden. Viele mobile Betriebssysteme müssen für das Einspielen von Updates mit einem Rechner verbunden werden. Sogenannte „Over-the-Air“ (OTA)-Updates, also Firmware-Updates ohne PC-Verbindung, sind bisher nur für Android, Windows Phone 7 und Symbian 3 verfügbar, unter gewissen Voraussetzungen auch für BlackBerry OS. Ältere Windows-Mobile- und Symbian-Versionen sowie Apples iOS<sup>1</sup> bieten bisher nur Betriebssystem-Updates mit Unterstützung eines PCs bzw. Macs. Entsprechend erfahren viele Smartphone-Besitzer bestenfalls dann von der Verfügbarkeit eines Updates, wenn sie das Gerät mit dem Rechner verbinden, im schlechtesten Fall müssen sie aktiv nach neuen Versionen suchen.

---

<sup>1</sup> Apple hat inzwischen OTA-Updates für iOS 5 angekündigt, das im dritten Quartal 2011 erscheinen soll.

Googles Android-Plattform, die laut Gartner im ersten Quartal 2011 mit 36 Prozent Marktanteil die Führerschaft der Smartphone-Betriebssysteme übernahm [3], bietet zwar OTA-Updates, leidet aber unter einem anderen Problem: Mit den schnellen Release-Zyklen von neuen Android-Versionen durch Google können viele Hersteller von mobilen Geräten nicht mithalten. Sie stellen deshalb nur für einen kurzen Zeitraum Firmware-Updates für ihre Produkte zur Verfügung, und das auch nur mit deutlicher Verzögerung zur Android-Version [4]. Erschwerend kommt hinzu, dass die Mobilfunkbetreiber viele Smartphones mit speziellen Anpassungen und eigenem Branding vertreiben. Diese benötigen jedoch ihrerseits speziell angepasst Firmware-Updates, die in der Regel wiederum noch später als die Updates der Hardware-Hersteller zur Verfügung gestellt werden. Die Konsequenz ist, dass auf den meisten Android-Geräten veraltete System-Software installiert ist. Beispielsweise wurde Android 2.3 im Dezember 2010 veröffentlicht, doch nach Angaben von Google war ein halbes Jahr später auf 64,6 Prozent aller Geräte noch Version 2.2 installiert. Im Mai 2011 liefen insgesamt nur 8,1 bzw. 0,3 Prozent aller Android-Geräte mit der dahin aktuellen Version 2.3.4 für die Smartphone-Klasse beziehungsweise Version 3.1 für die Tablet-Klasse.<sup>2</sup> Mit anderen Worten, über 90 Prozent aller Android-Geräte verwenden eine veraltete Betriebssystemversion, die möglicherweise bekannte aber noch nicht geschlossene Sicherheitslücken enthält.

### 13.2.2 Bedrohungen der Anwendungsebene

Die meisten Apps werden über einen speziellen App-Marktplatz bekannt gemacht und vertrieben. Solche Marktplätze werden inzwischen von allen großen Herstellern von Smartphone-Betriebssystemen betrieben. Für die mobilen Endgeräte von Apple (iPhone, iPad und iPod) ist Apples AppStore sogar der einzige Weg, um neue Anwendungen zu beziehen und auf den Geräten zu installieren. Der Nutzer muss also der Qualitätskontrolle der Marktplatzbetreiber vertrauen; er muss darauf vertrauen, dass die über den Marktplatz angebotenen Apps keine Sicherheitslücken oder Schadfunktionen aufweisen. Da die entsprechenden Kontrollverfahren nicht offengelegt sind und Apps keine standardisierte Zertifizierungsprozedur durchlaufen müssen, kann über die Qualität der durchgeführten Kontrollen keine verbindliche Aussage getroffen werden. Dass diese Kontrollen mitunter versagen, belegt der aktuelle Malware-Report von Kaspersky Lab [4]. Demnach wurden allein im ersten Quartal 2011 über 50 Apps mit trojanischen Funktionalitäten aus Googles Android-Market entfernt. Laut diesem Bericht ist es nur eine Frage der Zeit, bis mobile Botnetze im großen Stil auftauchen, wie wir sie bisher nur aus der Welt der Desktops kennen.

Die Problematik des Vertrauens in Closed-Source-Software besteht zwar im Prinzip auch bei Software, die auf Desktop-Betriebssystemen installiert wird. Auf einer

---

<sup>2</sup> Aktuelle Daten verfügbar unter:

<http://developer.android.com/resources/dashboard/platform-versions.html>

mobilen Plattform wird dieses Problem jedoch dadurch verschärft, dass beispielsweise Smartphones viele verschiedene Datenquellen aggregieren und diese über eine einheitliche Schnittstelle leicht für andere Anwendungen zugänglich machen. Darüber hinaus können schadhafte Apps ihrem Besitzer direkt beträchtliche Kosten verursachen, indem etwa SMS-Nachrichten an Mehrwertdienste gesendet oder Sonderrufnummern angerufen werden. Wenn sich das Smartphone durch die NFC-Technologie<sup>3</sup> in Zukunft auch noch als mobile Geldbörse etabliert, wie es derzeit von Google mit großen Partnern aus dem Finanzsektor als „Google Wallet“ für seine Android-Plattform forciert wird [5], wird sich diese Problematik voraussichtlich weiter verschärfen.

Eine App kann aber umgekehrt auch nicht zuverlässig feststellen, ob sie in einer sicheren, nicht modifizierten Umgebung ausgeführt wird. Eine durch Schadcode manipulierte Ausführungsumgebung könnte die Mobile App missbrauchen, um gezielt Daten auszuspähen, zu manipulieren oder über die Mobile App Zugriffe auf die Daten anderer Benutzer, andere Geschäftsprozesse etc. zu gelangen. Besonders für sicherheitskritische Anwendungen wären hierfür Techniken aus dem Trusted-Computing-Umfeld interessant. So könnte etwa der Server nur dann den Zugriff auf die Unternehmensdaten für eine mobile App gewähren, wenn die Plattform zuvor ihre Vertrauenswürdigkeit attestierte hat. Die Trusted Computing Mobile Working Group hat zu diesem Zweck bereits eine Referenzarchitektur entworfen [6], jedoch haben diese oder vergleichbare Ansätze bisher noch keinen Einzug in die Welt der mobilen Betriebssysteme gefunden.

### **13.2.3 Bedrohungen für verarbeitete Daten**

Smart Mobile Apps greifen über das Internet auf Daten in der Cloud zu. Entsprechend sind alle Überlegungen zur Sicherheit von Cloud-Computing auch für Smart Mobile Apps relevant (u. a. [2, 7]). Da diese Probleme jedoch nicht spezifisch für mobile Zugriffe sind, soll hierauf im Folgenden nicht weiter eingegangen werden.

Für mobile Apps stellt sich vielmehr die Frage, inwieweit die Daten einer Anwendung vor unberechtigten Zugriffen durch eine andere Anwendung geschützt sind. Hier ist zunächst die Zugriffskontrolle des mobilen Betriebssystems zu betrachten. Je nach Hersteller sind hier sehr starke Unterschiede zu verzeichnen. Die Größe der zu schützenden Einheit, also die Granularität der Kontrolle, spielt dabei eine wesentliche Rolle. Je geringer die verfügbare Granularität, desto größer sind die zu schützenden Einheiten und desto mehr Zugriffsrechte erhält eine App, auch wenn sie diese nicht im vollen Umfang benötigen würde. Da neue Software von unbekannten Autoren heutzutage auf einfachste Weise aus dem App-Marktplatz auf die mobilen Geräte gelangt, sollte die Isolation der Apps untereinander und die Beschränkung der Rechte auf das Nötigste oberstes Gebot sein.

---

<sup>3</sup> Near-Field-Communication

Aber auch legitime Zugriffe können missbraucht werden und damit unerwünschte Folgen haben. Beispielsweise gleicht die offizielle Facebook-App, die für alle populären mobilen Plattformen verfügbar ist, die Kontaktliste auf Smartphones mit den Freunden auf Facebook ab, wozu die App natürlich Zugriff auf die Kontakte des Geräts benötigt. Allerdings lädt die App auch alle Kontakte in das „Phonebook“ hoch, die *nicht* bei Facebook sind, ohne dass der Benutzer dies unterbinden kann. Auf diese Weise gelangt Facebook an viele Kontaktdata von Nicht-Mitgliedern und kann diese über die Telefonnummern als Identifikatoren verknüpfen, um das soziale Netzwerk auch über die Mitglieder hinaus zu erweitern. Dieses Beispiel verdeutlicht, warum die Aggregation verschiedener Datenquellen auf einem Gerät und die Verfügbarkeit dieser Daten über eine systemweite Schnittstelle eine besondere Bedrohung darstellt.

Nahezu alle modernen, smarten Endgeräte können ihre aktuelle Position bestimmen, sei es über die Mobilfunkzellen, die WLAN-Kennungen oder natürlich durch GPS-Module. Diese Positionsdaten werden für alle Apps verfügbar gemacht, wobei hierfür – je nach Zugriffsregeln des jeweiligen Betriebssystems – eine spezielle Berechtigung oder Zustimmung des Benutzers erforderlich sein kann. Werden diese Positionsdaten durch eine Anwendung an den Server übermittelt, um die Dienste ortsbezogen zu optimieren, kann dadurch Server-seitig leicht ein Bewegungsprofil des Nutzers erstellt werden. Schreibt die Anwendung die Positionsdaten sogar in Datensätze auf dem Server, so ist deren Auswertung auch noch nachträglich möglich, unter Umständen auch durch andere Nutzer mit Zugriff auf diese Daten. Somit ist die Privatsphäre des Nutzers ganz erheblich bedroht. Erschwerend kommt hinzu, dass sich der Nutzer eventuell nicht einmal im Klaren darüber ist, dass seine Bewegungen durch die Verwendung der App nachvollziehbar werden.

## 13.3 Existierende Lösungsansätze

Die Hersteller von mobilen Betriebssystemen, Endgeräten und klassischen Sicherheitslösungen sind sich der zuvor beschriebenen Bedrohungen natürlich bewusst und versuchen diesen auf unterschiedliche Weise zu begegnen. Im Folgenden geben wir einen Überblick über die verschiedenen Ansätze einiger derzeit erhältlicher Lösungen, ohne jedoch einen Anspruch auf Vollständigkeit zu erheben.

### 13.3.1 Sicherheit mobiler Betriebssysteme

Google folgt auf seiner Android-Plattform der Philosophie, dass den Apps und ihren Autoren grundsätzlich nicht vertraut werden kann. Entsprechend restriktiv sind hier die Vorgaben für die Zugriffsrechte, so dass jede App zunächst nur ihre eigenen Daten lesen und schreiben darf. Um auf systemweite Daten wie etwa die Kontakte oder den Kalender zuzugreifen, muss eine App die dafür vorgesehene Schnittstelle zum

Android-System verwenden. Allerdings unterliegen diese Schnittstellen einem speziellen Kontrollmechanismus. Damit der App diese Zugriffe vom System gewährt werden, muss sie schon bei der Installation angegeben haben, dass sie diese Berechtigung wünscht. Android definiert annähernd 200 verschiedene Rechte, die den Zugriff auf die persönlichen Daten regeln, oder aber auch auf Sensoren wie Mikrofon, Kamera und GPS-Empfänger. Daneben vergibt Android Rechte zur Verwendung von Systemfunktionen, wie das Senden von SMS-Nachrichten und das Initiieren von Telefongesprächen. Diese Berechtigungen muss der Benutzer bestätigen, bevor die App installiert wird. Dieses Vorgehen schafft eine gewisse Transparenz, verlagert aber damit das Problem auf den Benutzer, der ein entsprechendes Problembewusstsein haben muss, um eine sinnvolle Entscheidung treffen zu können.

Apples iOS verfügt nicht über solch feingranulare Berechtigungen, sondern verfolgt eine andere Strategie zur Zugriffskontrolle. Anders als auf der Android-Plattform erlaubt Apple die Installation von Apps ausschließlich über den Apple AppStore. Dadurch gibt es für iOS-Geräte einen zentralen Punkt, an dem Apple regulierend eingreifen kann. Alle Programme, die in den AppStore eingestellt werden, müssen sich zunächst einem Review-Prozess unterziehen. Dabei analysiert Apple die verwendeten Systemfunktionen und testet das Programm rudimentär. Verwendet eine App Systemfunktionen, die sie zur Erbringung ihrer Funktionalität aber gar nicht benötigt, wird das Programm abgelehnt und nicht im AppStore aufgenommen. Hier übernimmt also der Hersteller in gewisser Weise die Verantwortung, dass eine App nur die Zugriffsrechte erhält, die sie benötigt. Bei Android-Geräten wird diese Entscheidung dem Benutzer überlassen. Dem bereits erwähnten Problem bei dieser Vorgehensweise steht als Vorteil gegenüber, dass ein Benutzer die volle Kontrolle über die Vergabe feingranularer Zugriffsrechte erhält und sich nicht auf einen Hersteller verlassen muss.

Die kanadische Firma Research in Motion (RIM) fokussiert sich mit ihren BlackBerry-Produkten besonders auf den Business-Bereich. Da die Sicherheit der Unternehmensdaten immer einen hohen Stellenwert hat, bieten die Unternehmenslösungen von RIM viele Funktionen zum Schutz der Daten und Geräte. So kann ein Unternehmen zu den BlackBerry-Smartphones einen „Enterprise-Server“ betreiben, mit dessen Hilfe sich die Smartphones aller Mitarbeiter zentral verwalten lassen. Damit lassen sich etwa Software-Updates aus der Ferne installieren oder alle Daten löschen, sollte ein Gerät abhanden kommen. Darüber hinaus können auf dem Enterprise-Server Sicherheits-Policies definiert werden, die automatisch auf allen Geräten umgesetzt werden. So kann beispielsweise dem Benutzer untersagt werden, selbst Apps auf seinem Smartphone zu installieren oder gewisse Einstellungen zu ändern, wie das Deaktivieren von Sicherheitsfunktionen auf dem Gerät. Dadurch behält ein Unternehmen auch über seine mobilen Geräte noch weitgehend die Kontrolle und kann so eine unternehmensweite Sicherheits-Policy besser durchsetzen.

Eine Smartphone-Plattform mit einem Sicherheitsniveau, das auch für behördliche Verschlussachen ausreicht, wurde von T-Systems im Projekt SiMKo 2 entwickelt. Die Entwicklung hat sich an dem Protection Profile für mobile Endgeräte des Bundesamts für Sicherheit in der Informationstechnik (BSI) orientiert. Das BSI hat den SiMKo-2-Smartphones dann auch eine Empfehlung für den Einsatz in Um-

gebungen mit hohem Schutzbedarf ausgesprochen [8]. Kern dieser Architektur ist ein Hardware-basierter Sicherheitsanker in Form einer Kryptokarte im microSD-Format. Alle Daten auf dem Gerät werden durchgängig verschlüsselt und für den Benutzer nur durch Eingabe seiner PIN entschlüsselt. Die Verbindung erfolgt ausschließlich über das Mobilfunknetz, wobei das Smartphone automatisch einen VPN-Tunnel zum entsprechenden Behörden- oder Unternehmensnetz aufbaut. Als Betriebssystem kommt hierbei ein speziell gehärtetes Windows Mobile 6.5 zum Einsatz, wobei Manipulationen daran durch kryptographische Prüfsummen, die beim Startvorgang erzeugt werden, erkannt werden können. Um die Sicherheit des Gerätes nicht durch den Benutzer zu gefährden, führt das Betriebssystem nur Programme aus, die über eine spezielle digitale Signatur verfügen, außerdem sind Kamera, GPS-Empfänger, WLAN und Bluetooth dauerhaft deaktiviert.

### ***13.3.2 Sicherheitslösungen für mobile Anwendungen***

Die skizzierten Maßnahmen der handelsüblichen Mobiltelefone wie Zugriffskontrollen und Transparenz verhindern jedoch nicht, dass Apps mit Schadfunktionen auf ein mobiles Endgerät gelangen können. Das haben auch die Hersteller von Sicherheits-Software erkannt und bieten mobile Varianten ihrer Security-Suiten an, wie etwa Kaspersky, McAfee, Norton oder AVG. Der Funktionsumfang dieser Produkte reicht vom obligatorischen Viren-Scanner über ein entferntes Löschen der Daten (Remote-Wipe) bei Verlust oder Diebstahl des Endgeräts bis hin zu Rufnummernsperrlisten und SMS-Spam-Filtern.

### ***13.3.3 Virtualisierung auf mobilen Geräten***

Die Funktionen der RIM-Produkte oder die SiMKo-Smartphones von T-Systems weisen zwar ein zufriedenstellendes Sicherheitsniveau auf, sind allerdings aus Sicht des Anwenders sehr restriktiv und wenig komfortabel. Oftmals führen Benutzer solcher Geräte deshalb zwei Gerät mit sich, ein dienstliches und ein privates. Über Virtualisierungslösungen, wie sie von VMware mit dem Hypervisor für Android-Smartphones verfügbar sind, könnten diese Restriktionen komfortabel gelockert und gleichzeitig ein hohes Sicherheitsniveau für einzelne Anwendungen beibehalten werden.

Der Hypervisor für Android erlaubt es beispielsweise, zwei komplett voneinander isolierte Instanzen des Android-Betriebssystems auf einem Smartphone laufen zu lassen. In einer Instanz kann der Benutzer dann auf seine privaten Daten zugreifen und beliebige Apps installieren. Die andere Instanz repräsentiert sein Arbeitsumfeld und enthält die Smart Mobile Apps des Unternehmens, die Zugriff auf die Unternehmensdaten haben. Durch die Isolationseigenschaft des Hypervisors sind die Daten einer Instanz nicht für die andere zugreifbar, eine bösartige App

im privaten Bereich kann also keinen Zugriff auf die Geschäftsdaten bekommen. Per Knopfdruck kann der Benutzer jederzeit zwischen den Instanzen wechseln und ist dabei trotzdem immer auf beiden Telefonnummern erreichbar. Da sich auch der Homescreen-Bereich bei diesem Wechsel ändert, ist für den Benutzer immer klar ersichtlich, welche Instanz er gerade verwendet.

### ***13.3.4 Sicheres mobiles Payment***

Die Nahfunktechnik NFC wird im Bereich der Smartphones zunehmend populärer und findet immer weitere Verbreitung. Mit NFC können zwei mobile Geräte direkt Daten miteinander austauschen, vergleichbar etwa mit Bluetooth. Der Industrieverband GlobalPlatform<sup>4</sup> spezifiziert auch bereits eine Sicherheitsarchitektur für NFC-Geräte, mit deren Hilfe sie eine kontaktlose Smartcard emulieren können [9]. Dadurch kann das Smartphone mit entsprechenden Anwendungen als elektronisches Ticket eingesetzt werden, als Zahlungsmittel für Micro-Payments, aber auch als vollwertige Kreditkarte.

Die Architektur fordert, dass in dem Gerät ein sogenanntes „Secure Element“ vorhanden sein muss, also ein vertrauenswürdiger, hardware-basierter, sicherer Speicher, der vom Rest des Gerätes zuverlässig abgeschottet ist. Dadurch soll verhindert werden, dass bei einer Kompromittierung des Betriebssystems gespeicherte sensible Daten, wie der Geldbetrag oder die Kreditkartennummer, manipuliert werden können. Ein Secure Element kann zu diesem Zweck entweder als separater Chip oder in die SIM-Karte des Netzbetreibers integriert werden.

Obwohl diese Technik an sich nicht neu ist, wird erst in jüngster Zeit das NFC-basierte mobile Payment sehr stark vorangetrieben. Besonders Google engagiert sich neuerdings stark auf diesem Gebiet: Durch die Integration der entsprechenden Sicherheitsarchitektur in die Android-Plattform und die Vorstellung des Referenz-Smartphones „Nexus S“ in Kooperation mit Samsung konnte Google die City-Group und MasterCard als Partner für seine Payment-Lösung „Google Wallet“ gewinnen. Weltweit kooperieren auch viele Mobilfunkanbieter mit Finanzdienstleistern, um NFC-basierte Zahlungen mit dem Handy zu ermöglichen. Nach einer aktuelle Studie starten bis Ende 2012 entsprechende Programme in über 20 Ländern, die meisten davon in Europa und Nordamerika, das Transaktionsvolumen wird für 2014 auf weltweit 50 Milliarden US-Dollar geschätzt [10]. Eine Umsetzung der Sicherheitsarchitektur mit Secure Elements ist somit dringend auch für andere Einsatzszenarien und Systeme geboten. Hier besteht jedoch noch erheblicher Nachholbedarf, da die existierenden Secure Elements noch zu wenig flexibel sind und nicht skalieren.

---

<sup>4</sup> Mitglieder sind neben Unternehmen aus dem Bereich Telekommunikation und Sicherheitstechnik auch Vertreter der Kreditwirtschaft, unter anderem Visa, MasterCard und American Express.

## 13.4 Lösungsansätze aus der Forschung

Der Blick auf aktuell eingesetzte Sicherheitslösungen hat gezeigt, dass die bestehenden Systeme sehr unterschiedliche Konzepte zur Absicherung ihrer Geräte und Anwendungen anbieten. Zusammengefasst kann man festhalten, dass die umgesetzten Konzepte entweder sehr restriktive Vorkehrungen treffen, die vom Hersteller kontrolliert werden und den Nutzern wenig Kontrollmöglichkeiten und Flexibilität einräumen, oder aber sie verfolgen die entgegengesetzte Philosophie. Dem Benutzer werden völlige Freiheiten bei der eigenständigen Umsetzung von Sicherheitsmaßnahmen gewährt, mit den entsprechenden Einbußen beim Komfort und ggf. sogar bei der Sicherheit aufgrund von unsachgemäßem Einsatz dieser Konzepte. Notwendig sind somit neue Ansätze, die sowohl durch systemintegrierte Maßnahmen auf nachprüfbare Weise die Vertrauenswürdigkeit der mobilen Plattform erhöhen, gleichzeitig aber auch darauf abzielen, Sicherheit als Komfort-Funktion as-a-Service anzubieten.

Im Folgenden werden beispielhaft einige aktuelle Forschungsarbeiten skizziert, die sich den beiden oben aufgeworfenen offenen Fragestellungen widmen. Es werden Ansätze vorgestellt, die darauf abzielen, das Vertrauen in die mobile Plattform zu stärken, so dass die dorthin übermittelten Daten sicher verarbeitet und gespeichert werden. Für die zweite Fragestellung wird untersucht, wie das Cloud-Paradigma durch Security-as-a-Service-Ansätze dabei helfen kann, mobile Geräte besser abzusichern.

### 13.4.1 Vertrauenswürdige mobile Geräte

Im Bereich stationärer Rechner oder auch Laptops finden zunehmend Hardware-Plattformen Verbreitung, die das so genannte Trusted Computing unterstützen. Der Grundgedanke beim Trusted Computing ist, dass ein System einem anderen System, mit dem es in irgendeiner Weise kooperieren möchte, zuverlässig und nachweisbar attestieren kann, dass es sich in einem als vertrauenswürdig angesehenen Zustand befindet. Befindet sich ein System in einem vertrauenswürdigen Zustand, dann kann das andere System davon ausgehen, dass sich sein Kooperationspartner erwartungsgemäß verhält. Die Gefahr, mit einem manipulierten Gerät zu interagieren, wird dadurch erheblich reduziert. Beispielsweise könnte ein Server einem Client erst dann kritische Daten übergeben, wenn sich der Server davon überzeugt hat, dass der Client ein Kontrollsysteum aktiviert hat, so dass die Daten entsprechend der vergebenen Regeln des Servers auf dem Client verarbeitet werden. Auf diese Weise kann beispielsweise die Gefahr des unautorisierten Kopierens und Weitergebens von Daten, die auf dem Client verarbeitet werden, verringert werden. Erste rudimentäre Beispiele solcher Vorgehensweisen sind bereits seit langem bekannt; sie sind unter dem Namen DRM (Digital Rights Management) für Multimediadaten im Einsatz. Auch wenn das DRM-Konzept als solches auf große Widerstände bei den Verbrauchern gestoßen ist, so ist doch die dahinter liegende Idee, eine abgeschlossene Ausfüh-

rungsumgebung zu schaffen, um auf Dritt-Systemen vertrauenswürdig ein Regelwerk umzusetzen, das vom Besitzer der Daten festgelegt wurde, sehr interessant für den Anwendungskontext von Smart Mobile Apps.

Kernelement dieser Trusted-Computing-Architektur ist ein Secure Element, das so genannte Trusted-Plattform-Module (TPM), das im Wesentlichen aus einem sicheren Speicher mit eigenem Krypto-Prozessor besteht. Solche TPMs werden seit langem in vielen PCs und Laptops eingesetzt, zunehmend auch in Servern. Für mobile Geräte gibt es seit 2008 eine analoge Spezifikation für ein Secure Element, das sogenannte Mobile Trusted Module (MTM) [11–13]. Allerdings wurde diese Spezifikation bisher noch in keinem Gerät in nennenswerter Stückzahl implementiert.

Die möglichen Anwendungsfälle für hardware-basierte Sicherheitsmodule, die Secure Elements, auf mobilen Geräten sind jedoch vielfältig, so dass weitere Forschungsarbeiten hierzu erforderlich sind. Ein Secure Element kann den Zustand des Betriebssystems beim Booten überwachen und dadurch eine Art Selbstdiagnose durchführen. Wurden Komponenten des Systems durch eine Malware manipuliert, so kann dies hierdurch aufgedeckt werden. Der Firmen-Server würde dann gegebenenfalls den Zugriff auf die Unternehmensdaten verweigern, wenn das Gerät kompromittiert wurde. Und als sicherer und vertrauenswürdiger Datenspeicher würde ein Secure Element eine ideale Grundlage für sicherheitskritische Apps wie Payment-Anwendungen bieten.

Aktuelle Forschungs- und Entwicklungsarbeiten im Forschungsbereich Embedded Security am Fraunhofer AISEC<sup>5</sup> in München zielen deshalb darauf ab, skalierende, flexibel anpassbare Hardware-Sicherheitsmodule als solche Secure Elements zu entwickeln. Dazu wird basierend auf einem FPGA ein Sicherheitsmodul mit speziellen Kryptoprozessoren, sicheren Bibliotheken als Schnittstelle zum Betriebssystem und speziellen, integrierten Verfahren zur Abwehr von Angriffen (Sicherheit by Design) entwickelt. Es wird eine FPGA-basierte, gehärtete Kryptobibliothek entstehen, die modulare Bausteine zur Verfügung stellt, die einzeln oder als Ganzes in Systemen integriert werden können. Besonderer Wert wird dabei auf die Resistenz gegen Fehler- und Seitenkanalangriffe gelegt, und geeignete Gegenmaßnahmen werden in den Modulen umgesetzt.

### ***13.4.2 Security-Services für mobile Geräte***

Cloud-Computing ermöglicht schon jetzt viele Anwendungen auf mobilen Geräten, die ohne die Speicher- und Rechenkapazität der Cloud so nicht möglich wären. Da liegt die Überlegung nahe, auch Sicherheitsdienste in die Cloud auszulagern [14]. Eine Idee besteht beispielsweise darin, ein Replikat eines mobilen Betriebssystems in der Cloud laufen zu lassen [15]. Dabei werden alle Verbindungen des mobilen Geräts über einen Proxy geleitet und werden dabei auch an das Replikat in der Cloud gespiegelt. Die Ereignisse der Benutzerinteraktion sendet das Gerät eigenständig

---

<sup>5</sup> Vgl. <http://www.aisec.fraunhofer.de>

an sein Replikat. Dadurch sind Gerät und Replikat immer in einem konsistenten Zustand.

Das Replikat in der Cloud unterliegt nicht den starken Ressourcenbeschränkungen wie das mobile Gerät selbst. Folglich kann dort der Zustand des Betriebssystems mit aufwendigeren Methoden auf mögliche Kompromittierungen hin untersucht werden, die auf dem Gerät spürbare Latenzen erzeugen würden, etwa durch die Interaktionen mit dem Betriebssystem. Da hier auch nicht auf die Akku-Laufzeit geachtet werden muss, können dabei auch rechenintensive Verfahren zur Anwendung kommen. Dadurch kann ein möglicher Angriff zwar nicht verhindert werden, er wird aber frühzeitig erkannt, so dass schnell weitere Maßnahmen ergriffen werden können.

Am Fraunhofer AISEC und am Lehrstuhl für Sicherheit in der Informatik an der TU München werden Lösungen zur Malware-Erkennung sowie zur sicheren Integration von mobilen Geräten entwickelt. Damit können mobile Endgeräte sowohl als sicherer Anbieter von Diensten und Apps, als auch als App-Nutzer in die Cloud eingebunden werden. Dazu werden mobile Plattformen (u. a. auf Android-Basis) weiterentwickelt, die virtualisierte Umgebungen zur Ausführung von Diensten auf mobilen Endgeräten bereitstellen und die Managementkomponenten integrieren, um eine nutzerzentrierte Verwaltung von sicherheitskritischen Informationen zu ermöglichen. Die am AISEC bereits entwickelte Lösung zur nutzerkontrollierbaren Weitergabe personenbezogener Daten wird schrittweise um weitere Dienste erweitert, wobei ein Schwerpunkt auf der Entwicklung von Services zum Management von digitalen virtuellen Identitäten (IdM) liegen wird. Dazu wird eine Identity-as-a-Service-Lösung konzipiert, die ein Framework zur Integration unterschiedlichster Identitätsmanagementlösungen, wie den neuen Personalausweis (nPA), die SIM-Karte, OpenID, Microsoft InfoCards, beliebige Smartcards, als modularisierte, wiederverwendbare Dienste anbietet. Ergänzend dazu werden Monitoringtools entwickelt, die es erlauben, den Sicherheitsstatus von Apps, die in der Cloud laufen, permanent zu überwachen. Ein solcher Cloud-Leitstand ist ein komfortabler Kompromiss zwischen nutzerorientierter Kontrollierbarkeit der Sicherheit von Apps und systemintegrierten, nicht umgehbbaren Kontrollen auf Seiten des App-Betreibers.

Im Bereich Malware-Erkennung werden Ansätze entwickelt, um mit Methoden der Künstlichen Intelligenz und einer Analyseplattform Android-basierte Geräte kontinuierlich auf ihre Sicherheit zu prüfen. Ziel ist es, mittels verschiedener Sensoren, die beispielsweise Dateideskriptoren oder die CPU-Auslastung überwachen, Apps im Hinblick auf ihren Sicherheitszustand zur Laufzeit zu analysieren und Angriffe sowie unerwünschte Informationsflüsse anhand anomalen Verhaltens zu erkennen. Über die Erweiterung der Dalvik VM einer Android-Plattform um Sensorfunktionalität können beispielsweise Meldungen des Kernels ausgewertet werden, aber auch das Monitoring des Netzverkehrs ist möglich. Wegen der limitierten Rechenkapazität der mobilen Plattform wird die Analyse in die Cloud ausgelagert. Dadurch wird auch ein Datenabgleich mit bereits beobachteten Angriffen möglich und es können Daten unterschiedlicher Geräte korreliert werden, so dass ein umfassenderes Lagebild erstellt werden kann.

## 13.5 Zusammenfassung und Ausblick

Apps sind bereits heute im privaten Umfeld weit verbreitet. Auch wenn derzeit noch ein großer Teil der Apps als rein lokale Anwendungen auf mobilen Geräten ausgeführt werden, so ist die Entwicklung hin zu Smarten Mobilen Apps deutlich zu sehen. Kennzeichnend für Smart Mobile Apps ist ihre Eigenschaft, Dienste zu nutzen, die über das Internet oder ein privates Netzwerk bereitgestellt werden. Viele dieser Dienste werden in einer Cloud betrieben und ermöglichen es dem Nutzer somit, einen konsistenten Datenbestand auf unterschiedlichen mobilen und stationären Geräten zu führen. Als Business-Apps werden derartige Apps in Zukunft ein integraler Bestandteil von Geschäftsprozessen sein.

Die Frage der Datensicherheit bei der Nutzung von Smart Mobile Apps tritt somit spätestens bei deren flächendeckender Nutzung in geschäftlicher Umgebung zu Tage. Smart Mobile Apps sind in besonderem Maße anfällig gegen Angriffe, die darauf abzielen, Daten auszuspähen, sich unter einer falschen Identitäten Zugriffe auf Daten zu verschaffen, Daten zu manipulieren oder die Verfügbarkeit von Daten-Diensten zu behindern.

Der Blick auf aktuell eingesetzte Sicherheitslösungen hat gezeigt, dass die bestehenden Systeme sehr unterschiedliche Konzepte zur Absicherung ihrer Geräte und Anwendungen anbieten. Zusammengefasst kann man festhalten, dass die umgesetzten Konzepte entweder sehr restriktive Vorkehrungen treffen, die vom Hersteller kontrolliert werden und den Nutzern wenig Kontrollmöglichkeiten und Flexibilität einräumen, oder aber sie verfolgen die entgegengesetzte Philosophie, dem Benutzer viele Freiheiten bei der eigenständigen Umsetzung von Sicherheitsmaßnahmen zu gewähren, mit den entsprechenden Einbußen beim Komfort und ggf. sogar bei der Sicherheit aufgrund von unsachgemäßem Einsatz dieser Konzepte. Notwendig sind somit neue Ansätze, die sowohl durch systemintegrierte Maßnahmen auf nachprüfbare Weise die Vertrauenswürdigkeit der mobilen Plattform erhöhen, gleichzeitig aber auch darauf abzielen, Sicherheit als Komfort-Funktion as-a-Service anzubieten.

Aktuelle Forschungsarbeiten zielen darauf ab, in diesen Problembereichen Fortschritte zu erzielen. Neben der Integration von Virtualisierungskonzepten werden insbesondere hardware-basierte Secure Elements entwickelt, die notwendige Vertrauensanker in Plattformen unterschiedlicher Ausprägungen bereitstellen würden. Sicherheit als spezieller Service, der u. a. auch durch die Cloud erbracht werden kann, ist ein interessanter Ansatz. So konnten dedizierte Malware-Erkennungsverfahren offline in der Cloud durchgeführt werden und Nutzern mobiler Apps frühzeitig Hinweise auf Sicherheitsprobleme vermitteln. Mit vereinheitlichten Identitätsmanagementservices könnten die Probleme, die sich aus Identitätsdiebstählen in der mobilen Welt vielfältig ergeben, substanzial verringert werden. Erst wenn die Aspekte, also Komfort, Flexibilität und Sicherheit, für Smart Mobile Apps geeignet Berücksichtigung finden, können die Sicherheitsrisiken beherrschbar und das erhebliche Potential mobiler Apps kann auch im Geschäftsumfeld genutzt werden.

## Literaturverzeichnis

1. Eckert C (2009) IT-Sicherheit: Konzepte – Verfahren – Protokolle, 6. Auflage, Oldenbourg
2. Steinberger W, Ruppel A (2009) Cloud Computing Sicherheit – Schutzziele. Taxonomie. Marktübersicht. Technischer Bericht des Fraunhofer SIT, <http://www.sit.fraunhofer.de/presse/texte-studien/Cloud-Security-Studie.jsp> 15. Juli 2011
3. Reisinger D (2001) *Gartner: Android leads, Windows Phone lags in Q1.* [http://news.cnet.com/8301-13506\\_3-20064223-17.html](http://news.cnet.com/8301-13506_3-20064223-17.html). 15. Juli 2011
4. Namestnikov Y (2011) *Malware-Report, erstes Quartal 2011.* Kaspersky Lab <http://www.viruslist.com/de/analysis?pubid=200883738>. 15. Juli 2011
5. Google Inc. (2011) Google Wallet. <http://www.google.com/wallet/>. 15. Juli 2011
6. TCG Mobile Reference Architecture Version 1.0 Revision 1. Trusted Computing Group. (2007) [http://www.trustedcomputinggroup.org/resources/mobile\\_phone\\_work\\_group\\_mobile\\_reference\\_architecture](http://www.trustedcomputinggroup.org/resources/mobile_phone_work_group_mobile_reference_architecture). 15. Juli 2011
7. Meiko J, Schwenk J, Gruschka N, Iacono LL (2009) On Technical Security Issues in Cloud Computing. In: 2009 IEEE International Conference on Cloud Computing, IEEE, pp 109–116
8. SiMKo 2 – eine Lösung für die sichere mobile Kommunikation. Bundesamt für Sicherheit in der Informationstechnik. [https://www.bsi.bund.de/DE/Themen/weitereThemen/MobileSecurity/SiMKo2/simko\\_node.html](https://www.bsi.bund.de/DE/Themen/weitereThemen/MobileSecurity/SiMKo2/simko_node.html). 15. Juli 2011
9. GlobalPlatform System Messaging Specification for Management of Mobile-NFC Services v1.0 (2011) <http://www.globalplatform.org/specificationdownload.asp?id=7585>. 15. Juli 2011
10. Wilcox H (2011) NFC Retail Marketing & Mobile Payments – Markets, Forecasts & Strategies 2011–2016. Juniper Research
11. TCG Mobile Trusted Module Specification Version 1.0 Revision 6 (2008) Trusted Computing Group. [http://www.trustedcomputinggroup.org/resources/mobile\\_phone\\_work\\_group\\_mobile\\_trusted\\_module\\_specification](http://www.trustedcomputinggroup.org/resources/mobile_phone_work_group_mobile_trusted_module_specification). 15. Juli 2011
12. Ramón C und Sailer R (2006) Trusted Mobile Computing. In: Proceedings of IFIP Workshop on Security and Privacy in Mobile and Wireless Networks
13. Pisko E, Rannenberg K, Roßnagel H (2005) Trusted Computing in Mobile Platforms. Datenschutz und Datensicherheit, 29:526–530
14. Chun B-G und Maniatis P (2009) Augmented smartphone applications through clone cloud execution. In: Proceedings of the 12th conference on Hot topics in operating systems, HotOS'09, Seiten 8–8, Berkeley, CA, USA. USENIX Association.
15. Portokalidis G, Homburg P, Anagnostakis K, Bos H (2010) Paranoid Android: versatile protection for smartphones. In: Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10, New York, NY, USA. ACM, pp 347–356.

---

# Kapitel 14

## Das Smartphone als sichere Burg

### Sind die Sicherheitsfunktionen für Apps auf Smartphones ausreichend?

Patrick Wildt und Ralf Meister

**Zusammenfassung** Smartphones sind populär, da sie die Funktionen eines mobilen Telefons mit denen eines „Personal Digital Assistant“ (PDA) vereinen. Die Möglichkeit des Hinzufügens von Software, den Apps, erfordert Schutzfunktion auf dem Smartphone. Dieser Artikel stellt zwei populäre Smartphone-Betriebssysteme vor und diskutiert ihre Schutzfunktionen. Es zeigt sich, dass wegen teilweise ineffektiver Sicherheitsfunktionen kein lückenloser Schutz gewährleistet ist. Daraus ergibt sich der Bedarf an einem Sicherheitskonzept, welches alle Komponenten des Smartphones einbezieht.

#### 14.1 Einleitung

Dieser Artikel diskutiert, inwieweit die Sicherheitsfunktionen für Apps auf Smartphones ausreichend sind. Aber was ist das überhaupt, ein Smartphone? Der Besitzer eines solchen Smartphones wird zur Beantwortung dieser Frage sein Smartphone zücken, die Wikipedia App starten, und dann die folgende Charakterisierung finden: „Ein Smartphone kann auch als ein kleiner transportabler Computer (PDA) mit zusätzlicher Funktionalität eines Mobiltelefons verstanden werden.“ [1].

Smartphones werden von ihren Besitzern oft nur als mobile Telefone angesehen. Die Geräte haben jedoch in den letzten Jahren einiges an Leistung und Komponenten gewonnen, sodass man sie eher als mobilen Computer betrachten müsste. Sie basieren zum Großteil auf denselben Techniken wie typische PCs. Diese mobilen Rechner sind durch die Funkschnittstellen GSM oder UMTS mehr oder weniger

---

Patrick Wildt  
GeNUA mbH, Kirchheim bei München,  
E-mail: [Patrick\\_Wildt@genua.de](mailto:Patrick_Wildt@genua.de)

Ralf Meister  
GeNUA mbH, Kirchheim bei München,  
E-mail: [Ralf\\_Meister@genua.de](mailto:Ralf_Meister@genua.de)

dauerhaft mit dem Internet verbunden, mit all den damit einhergehenden Vor- und Nachteilen.

Eine herausragende Eigenschaft der Smartphones ist ihre Erweiterbarkeit in Hinblick auf die installierte Software, die in diesem Umfeld App, eine Abkürzung für Applikation, genannt wird.

Apps spielen heutzutage eine wichtige Rolle im Alltag der Smartphone-Besitzer. Darunter ist in der letzten Zeit auch eine nicht unerhebliche Menge an sicherheitskritischen Applikationen, wie beispielsweise zum Homebanking, hinzugekommen. Es wird in den nächsten Jahren auch weiterhin ein Zuwachs solcher Apps auftreten [2].

Sicherheitskritische Applikationen wie Bankingsoftware oder der Mail-Zugriff in die Firma – all dies kann ohne Probleme eingerichtet und benutzt werden. Auf die Sicherheit der eingesetzten Verfahren und Programme wird dabei kaum geachtet. Nicht nur Apps sind sicherheitsbedenklich: Die Betriebssysteme und unterliegenden Komponenten sind auch ein wesentlicher Faktor. Es stellt sich die Frage, ob allen Komponenten Vertrauen geschenkt werden darf. Nur eine lückenlose Kette von der Hardware über das Betriebssystem bis zur Software in der Form der Apps kann die Sicherheit des gesamten Systems garantieren.

In diesem Beitrag soll aufgezeigt werden, welche Sicherheitsmechanismen in aktuellen Smartphones vorhanden sind und wo Verbesserungsbedarf besteht. Die beiden betrachteten Systeme sind das iPhone von Apple und Android-basierte Systeme, die aufgrund ihrer Popularität ausgewählt wurden.

### ***14.1.1 Apple iPhone***

Bei dem iPhone von Apple handelt es sich um ein eher geschlossenes System. Erste Versionen vom iPhone kamen 2007 auf den Markt. Momentan ist das Apple iPhone 4 die neueste Hardware-Generation (Stand 13. Juli 2011). Das Betriebssystem bezeichnet Apple als iOS, welches auf Mac OS X basiert. Die zur Zeit aktuelle Version ist 4.4.3.

Der Marktanteil liegt im vierten Quartal 2010 bei 16,0 %, die Verkaufszahlen in demselben Quartal liegen bei 16,2 Millionen Geräten [3].

### ***14.1.2 Android***

Android ist im Vergleich zu Apple iPhone ein sehr offenes System. Ursprünglich ist Android eine Entwicklung des gleichnamigen Unternehmens Android von Andy Rubin. Im Jahr 2005 kaufte Google das Unternehmen Android und brachte Android im Jahr 2007 in die Open Handset Alliance [4] ein. Android basiert auf einem modifizierten Linux Kernel. Die Benutzerprozesse in Android unterscheiden sich aber

stark von einem „normalen“ Linux und sind in Hinblick auf die Sicherheitsbedürfnisse einer mobilen Plattform abgeändert.

Von Android gibt es zur Zeit zwei aktuelle Versionen, die Versionen 2.3.4 „Gingerbread“ für Smartphones und die Version 3.1 „Honeycomb“ für Tablett-Computer (Stand 13. Juli 2011).

Der Marktanteil wird im vierten Quartal 2010 mit 32,9 % angegeben, die Anzahl der verkauften Geräte wird in demselben Quartal mit 33,3 Millionen angegeben [3].

## 14.2 Prinzipieller Aufbau eines Smartphones

Smartphones haben nicht nur auf der Hardwareebene starke Ähnlichkeit zu typischen PCs, auch die Software ist ähnlich strukturiert. Die Apps greifen hierbei nicht direkt und untereinander unkoordiniert auf die Hardware zu. Stattdessen ist eine zusätzliche Schicht vorhanden – das Betriebssystem –, welches zum einen die Zugriffe auf die Hardware koordiniert und kontrolliert an die Apps weitergibt. Der Zugriff auf die Hardware geschieht über Geräteknoten, welche als Kommunikationsendpunkte zum Ansprechen der Hardware bezeichnet werden können. Zum anderen implementiert das Betriebssystem Zugriffsrechte und kann gezielt bestimmten Apps den Zugriff auf die Hardware erlauben oder auch verbieten.

Die Zugriffsrechte spielen auch in einer weiteren Komponente, dem Dateisystem, eine Rolle. Das Dateisystem ist der Ort, an dem sowohl der Programmcode als auch die Daten der Apps hinterlegt werden.

Der prinzipielle Aufbau ist in der Abb. 14.1 skizziert. In dieser Abbildung sind die einzelnen Schichten mit Bezeichnungen für Ebenen versehen. Die Nummerierung der Ebenen orientiert sich an dem OSI/ISO-Schichtenmodell [5], bei denen die Hardwareebene auf der Schicht 1 angeordnet ist. Das Betriebssystem ermöglicht die Abstraktion von Geräteknoten, Dateien oder auch TCP-Strömen und ist daher auf Schicht 2–4 anzusiedeln. Die Anwendungen befinden sich auf den höheren Schichten 5–7.

Obwohl der Benutzer über die physikalischen Schnittstellen mit den Apps kommuniziert, wirkt es für ihn so, als ob er direkt mit der App interagiert. Daher wird man den Benutzer der nächsthöheren Schicht auf Ebene 8 zuordnen.

## 14.3 Schutzbedarf

Das Smartphone ist Träger verschiedenster Daten seines Benutzers. Aufgrund des mobilen Charakters des Smartphones ist dieses leicht zu transportieren und dementsprechend klein und kann somit leicht abhanden kommen. Daher besteht zusätzlicher Schutzbedarf für den Fall, dass das Smartphone entwendet oder verloren gegangen ist.

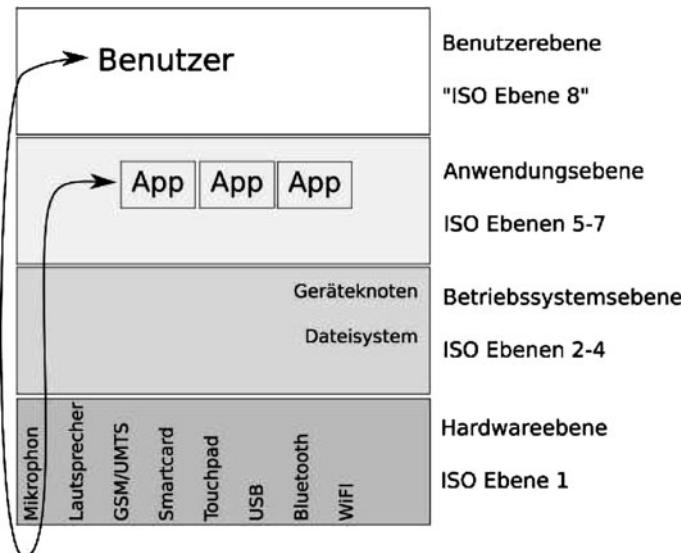


Abb. 14.1 Logischer Aufbau eines Smartphones

Als Kombination aus mobilem Telefon und PDA sind zum einen die Daten zu schützen, die für den Verwendungszweck als Telefon notwendigerweise vorhanden sind. Zum anderen sind aus der Funktion als PDA die Daten der Apps zu schützen.

Daten, die für den Verwendungszweck als Telefon anfallen, sind z. B.:

- die Sprachdaten,
- die Anruflisten,
- das Adressbuch oder auch
- versandte und empfangene SMS.

Für den Zugriff auf das Funknetzwerk des Betreibers sind

- Zugangsdaten auf der SIM-Karte

gespeichert, die geschützt werden müssen.

Zudem fallen aber auch technische Daten wie z. B.

- die Funkzelle oder
- die GPS-Position

an, aus der die nähere Position des Benutzers ermittelt werden kann.

Beispiele für Apps, bei denen personenbezogene und damit schützenswerte Daten anfallen, sind z. B.:

- E-Mails oder E-Mail-Zugänge
- Terminkalender
- Kontoinformationen
- VPN-Schlüssel
- Zugangsdaten zu (Web-)Applikationen.

## 14.4 Ungewollte Preisgabe der zu schützenden Daten

Unter welchen Umständen können diese zu schützenden Daten ungewollt preisgegeben werden?

An der Nutzung eines mobilen Telefonnetzes sind drei Personengruppen beteiligt. Dies sind der Benutzer, der Hersteller des Smartphones und der Betreiber des mobilen Telefonnetzes. Abhängig vom Smartphone kann sich der Hersteller noch in weitere Personengruppen aufspalten.

Im Fall der ungewollten Preisgabe der zu schützenden Daten kommt eine weitere Personengruppe hinzu, welche unautorisiert auf die Daten zugreift. Diese Personengruppe wird nachfolgend Angreifer genannt.

In den folgenden sechs Abschnitten werden Methoden vorgestellt, mit denen der Angreifer Zugriff auf die zu schützenden Daten erlangen kann.

### 14.4.1 Physikalischer Zugriff

Der physikalische Zugriff auf das Smartphone ermöglicht den Zugriff auf das Dateisystem und damit auch den Zugriff auf die zu schützenden Daten. Der Angreifer wird daher versuchen, das Smartphone zu entwenden. Ein ähnlicher Fall tritt ein, falls das Smartphone verloren gegangen ist. Auch in diesem Fall möchte der Benutzer nicht, dass der Finder Zugriff auf seine Daten hat.

Wenn der physikalische Zugriff nicht möglich ist, muss der Zugriff aus der Ferne erfolgen. Die nächsten fünf Abschnitte befassen sich damit, mit welchen Methoden dies erfolgen kann.

### 14.4.2 Herausgabe der Daten durch den Benutzer

Die einfachste Methode ist es, den Benutzer mit einzubeziehen. Diese erfolgt, indem der Benutzer kontaktiert wird, was z. B. durch „Phishing“ (Vorspielen einer falschen Identität im Internet) oder fingierte Anrufe geschehen kann. Diese Art des Angriffs wird auch als „Social Engineering“ bezeichnet. Dies ist ein Angriff auf die Benutzer-Ebene 8 in Abb. 14.1.

### 14.4.3 Kommunikation mit der Außenwelt

Das Smartphone verbindet sich im Falle eines Telefonats oder bei Zugriffen in das Internet per Funk mit der nächsten erreichbaren Funkzelle. Die Übertragung über Funk wird oftmals auch als die Luftschnittstelle des Telefons bezeichnet. Das Abhö-

ren der Datenübertragung über die Luftschnittstelle („over the air“) kann prinzipiell nicht unterbunden werden.

Zudem besteht im Internet die altbekannte Gefahr, sich durch Manipulationen im Namensdienst DNS mit falschen Servern zu verbinden. Hiermit kann man dem Anwender dann z. B. eine gefälschte Web-Seite vorgaukeln. Aber diese Methode kann auch die Grundlage für sogenannte „man in the middle“-Angriffe sein, mit denen der Angreifer den Netzwerkverkehr mitschneiden kann.

Diese Angriffe, die man als passives oder auch aktives Abhören charakterisieren kann, finden auf den Ebenen 0 bzw. 1 bis 4 statt.

#### ***14.4.4 Ausnutzung installierter Apps***

Der Angreifer kann die installierte Software auf dem Smartphone ausnutzen, um Zugriff auf die Daten zu erhalten. So wird in der Regel ein Webbrower auf dem Smartphone installiert sein. Durch das Anbieten fingierter Seiten eines Webservers kann der Angreifer nun versuchen, mit Hilfe des bereits installierten Webbrowsers auf die Daten des Benutzers zuzugreifen.

Eine andere Möglichkeit ist es, die installierten Apps zu manipulieren, um Zugriff auf die Daten zu erhalten. Der Programmcode der Apps liegt im Dateisystem des Smartphones als Dateien vor. Diese Dateien können von anderen Apps, die man z. B. nur testweise installiert hat, ungewollt manipuliert werden. Daher kann auch in eine vertrauensvolle App Schadcode eingefügt werden.

#### ***14.4.5 Ausnutzung von Softwareschwachstellen***

Andere Möglichkeiten bestehen in der Ausnutzung von Schwachstellen in der Software. Dies betrifft sowohl die Applikationsebene als auch die Betriebssystemebene. Je nach App kann die Schwachstelle natürlich sehr unterschiedlich ausfallen. Die Softwareschwachstelle kann sich in der Logik der App befinden, z. B. bei web-basierenden Apps.

Die Softwareschwachstelle kann sich aber auch durch die Wahl einer Programmiersprache ergeben, bei der z. B. Sicherheitsüberprüfungen nicht automatisch durchgeführt werden. Stattdessen sind diese Überprüfungen vom Programmierer zu kodieren und werden dann ggfs. vergessen. Eine solche Sicherheitsschwachstelle ist z. B. ein „Buffer-Overflow“. Wenn von einem Angreifer durch einen Buffer-Overflow der Programmfluss bzw. die Programmlogik zu seinen Gunsten abgeändert werden kann, spricht man von einem „Exploit“.

Leider ist die hardwarenahe Sprache C eine solche unsichere Programmiersprache. Wegen der Nähe zur Hardware wird C gerne für den Betriebssystemkern selber und die Schnittstellenanbindung an das Betriebssystem benutzt. Daher ist der Schutz vor Buffer-Overflow lohnenswert.

### **14.4.6 Trojanische Pferde**

Eine weitere Möglichkeit ist es, den Anwender gezielt dazu zu bringen, ein „trojanisches Pferd“ in Form einer App zu installieren. Dieser App stehen dann alle Möglichkeiten offen, die das Betriebssystem ihm lässt.

Das trojanische Pferd kann auch, wie oben beschrieben, nach seiner Deinstallation noch Auswirkungen haben.

## **14.5 Schutzmechanismen und deren eventuelle Unzulänglichkeit**

Es gibt mehrere Maßnahmen, mit denen die in den vorhergehenden Abschnitten beschriebenen Bedrohungen abgewendet werden können. Da einzelne Maßnahmen gleichzeitig Schutz vor mehreren Bedrohungen bieten können, gibt es leider keine Eins-zu-eins-Abbildung der Schutzmaßnahmen zu den Bedrohungen.

In den folgenden Beschreibungen der Schutzmechanismen wird zudem dargestellt, wie die beiden betrachteten Plattformen Apple iPhone und Android den Schutz jeweils realisieren, sowie eine Einschätzung der Autoren über die Hinlänglichkeit des Schutzes gegeben.

### **14.5.1 Schutz vor der Datenherausgabe durch den Benutzer**

Gegen die Herausgabe der Daten durch den Benutzer helfen technische Methoden nur bedingt. Da die Einbeziehung des Benutzers oftmals durch Mails erfolgt, können SPAM-Filter hilfreich sein. Die wichtigste Maßnahme besteht in der Aufklärung des Benutzers über diese Schwachstelle.

Einschätzung: Die Herausgabe der Daten durch den Benutzer kann allein mit technischen Methoden nicht zuverlässig gelöst werden.

### **14.5.2 Schutz der Kommunikation auf der Luftschnittstelle**

Da prinzipiell das Abhören auf der Luftschnittstelle nicht unterbunden werden kann, hilft an dieser Stelle nur das Verschlüsseln der Daten. Bei der Konzeption von GSM bzw. UMTS wurde diese Verschlüsselung auch implementiert. Aber reicht diese Technologie auch heute noch aus?

Es sind Angriffe auf das sog. Baseband bekannt geworden. Das Baseband stellt für das Smartphone die Schnittstelle zu dem Provider dar. Mittels GSM und UMTS werden darüber hauptsächlich Telefonate übertragen und Verbindungen in das Internet ermöglicht. Der Code, der auf den aktuellen Chips läuft, stammt teilweise aus den 90ern und wurde in der Zwischenzeit nicht nachgebessert. Angriffe auf

die Chips laufen über Funk ab, indem eine Mobilfunkbasisstation in der Nähe des Ziels positioniert wird. Dadurch, dass die Basisstation des Angreifers durch seine Nähe einen leicht besseren Empfang aufweist, wird sich das Baseband dort einwählen. Auf diesem Weg kann man, vor allem durch die verwendeten veralteten Krypto-Operationen, den Audiostream mitschneiden sowie auch Datenverbindungen protokollieren. Diese Angriffsmethode wurde vor allem nach dem Vortrag „The Baseband Apocalypse“ von Ralf-Philipp Weinmann publik [6]. Glücklicherweise sind diese Angriffe derzeit relativ selten, obwohl die dafür benötigten Ressourcen sogar für den Privatanwender sehr günstig in der Anschaffung sind.

Einschätzung: Die Luftschnittstelle ist nur noch schwach geschützt.

### ***14.5.3 Schutz der Kommunikation im Internet***

Apps wie die vorher genannte Banking-Software sowie auch Instant-Messenger oder Mail-Programme benötigen eine Verbindung ins Internet, um Informationen nachzuladen. Da es möglich ist, diese Daten mitzuschneiden, muss ihr Transport geschützt werden. „Transport Layer Security“ (TLS, früher „Secure Socket Layer“, SSL) ist als hybrides Verschlüsselungsprotokoll momentaner Standard um gesicherte Netzwerkverbindungen herzustellen. Hierbei sollte mit einer Public-Key-Infrastruktur (PKI) gearbeitet werden, um die Authentizität des Verbindungspartners überprüfen zu können.

Dabei reicht es nicht, die Validität des Zertifikats zu überprüfen, da es mehrere nicht vertrauenswürdige ausländische Zertifizierungsstellen gibt, die solche erstellen könnten. Im Falle der AusweisApp, einer Software für den PC, mittels derer die elektronische Authentisierung des neuen Personalausweises genutzt werden kann, hat diese nur die Validität des Zertifikats geprüft. Eine Überprüfung, ob das Zertifikat zu den Servern des Herstellers passt, fand mit dem Softwarestand vom November 2010 nicht statt [7]. Doch nur wenn der Public-Key des Verbindungspartners der App von vornherein bekannt ist, kann die App die Authentizität des Partners prüfen.

Einschätzung: Die Umsetzung einer funktionierenden PKI ist komplex, insbesondere die Verwaltung und Verteilung zurückgezogener Zertifikate. Man muss sich auf die Zertifizierungsstellen verlassen. Im April des Jahres 2011 ist zudem eine als „SSL-GAU“ bezeichnete Kompromittierung einer Zertifizierungsstelle geschehen, die zeigt, wie anfällig das System der Zertifizierungsstellen ist [8].

### ***14.5.4 Schutzmechanismen des Betriebssystems***

Der prinzipielle Aufbau eines Smartphones in Betriebssystem- und Anwendungsebene ist ebenfalls als Schutzmechanismus zu verstehen, da mit der Betriebssys-

temsnittstelle Zugriffsbeschränkungen auf die Dateien und Geräteknoten durchgesetzt werden können.

Im Falle eines Betriebssystems für Smartphones wird bereits auf Seiten des Herstellers davon ausgegangen, dass böswillige Apps die Dienste des Betriebssystems kompromittieren wollen. Deshalb wurden in aktuellen Smartphones verschiedene Schutzkonzepte verwirklicht:

Der Linux-Kernel eines Android-Systems erlaubt den Apps keinen Zugriff auf App-fremde Daten oder Gerätefunktionen. Dies wird durch Zuweisung einer eigenen User-ID pro App und konsequente Ausnutzung der Zugriffsrechte im Dateisystem und den Geräteknoten erzwungen. Die Zugriffsrechte können jedoch bei der Installation erweitert werden. Hierzu wird der Benutzer bei der Installation gebeten zu bestätigen, dass er zusätzliche Rechte vergeben will. Zwar können beispielsweise Trojaner ausgeführt werden, jedoch wird aufgrund der Rechte definiert, inwieweit Zugriff auf Informationen möglich ist [9].

Im Vergleich zu Android verfügt der iOS-Kernel über einen „Sandbox“-Mechanismus, jedoch wird dieser nur angewandt, wenn eine App explizit damit eingesperrt werden soll. Programme können sich jedoch auch selber einsperren. Die Sandbox arbeitet mit einer Black-List, das heißt sie kann nichts ermöglichen, sondern sperrt nur ein. Auf Dateisystem-Ebene haben alle Apps Zugriffsrechte auf Dateien anderer Apps, da alle Apps mit derselben User-ID laufen. Falls ein Fehler in der Sandbox auftritt, kann dies ausgenutzt werden, um auf Dateien anderer Apps zuzugreifen. Die Apps verfügen nicht über direkten Zugriff auf Gerätefunktionen, sondern müssen mit Hilfe von APIs darauf zugreifen. Zusätzlich dazu können nur Apps ausgeführt werden, die von Apple signiert wurden [9].

Die Android-Implementation scheint hier im Vergleich Vorteile zu haben. Jedoch kann jede unsignierte App installiert werden. Das macht es zwar für den Benutzer einfacher, eigene Apps auf das System zu bekommen, es können dadurch aber auch Apps installiert werden, die nicht auf Schädlichkeit überprüft wurden. Apple hat mit dem AppStore ein Monopol und blockiert so die Aufnahme mancher Apps. Die Hauptmotivation ist, dass der Konzern durch den Aufzwang ihrer Signatur diese Apps zuerst untersuchen kann, um Schaden beim Benutzer zu verhindern.

Ein anderer Aspekt des AppStore von Apple ist seine finanzielle Seite. Für jede App, die über den AppStore gekauft wurde, wandert ein Teil der Gebühren an Apple. Durch die Bindung der Anwender an den Hersteller des Smartphones spricht man in diesem Zusammenhang auch von einem eigenen Ökosystem.

Einschätzung: Die Prüfverfahren von Apple zur Aufnahme in den AppStore sind nicht offengelegt und können nicht verifiziert werden. Zudem hat Apple ein finanzielles Interesse durch die Einnahmen des AppStore. Bei der Abschottung der Prozesse untereinander muss man sich auf das Betriebssystem verlassen. Die Frage der Zulässigkeit erweiterter Zugriffsrechte der Android Apps wird dem Benutzer überlassen.

### ***14.5.5 Schutz vor Buffer-Overflows***

Buffer-Overflows basieren auf unzureichender Prüfung des zur Verfügung stehenden Speicherbereichs. Wenn aufgrund der Architektur garantiert ist, dass die Prüfung immer stattfindet, können keine Buffer-Overflows entstehen.

Bei der Plattform Android werden Apps in der Programmiersprache Java geschrieben. Zur Ausführung des Java-Bytecodes kommt eine modifizierte Java Virtual Machine, die DalvikVM, zum Einsatz. Aufgrund dieser Architektur sind Buffer-Overflows in der beschriebenen Form nicht möglich. Allerdings ist auch im Falle von Android das Betriebssystem selbst nicht vor Buffer-Overflows geschützt.

Kann man Buffer-Overflows auch für unsichere Programmiersprachen verhindern? Das Problem eines Buffer-Overflows besteht darin, dass dieser es durch einen Exploit ermöglicht, den Programmlauf zugunsten des Angreifers zu verändern. Die dazu benötigte Programmlogik kann zum einen direkt vom Anwender kommen, indem er sie in dem Datenpaket schickt, welches den Buffer-Overflow auslöst. Die veränderte Programmlogik kann sich aber zum anderen auch schon irgendwo im Programmcode der App selbst befinden.

Ein Schutz für den Fall, dass der Angreifer den gewünschten Programmcode in den Daten des Buffer-Overflows schickt, besteht darin, das Speichersegment, in dem diese Daten gespeichert werden, als nicht-ausführbar zu kennzeichnen. Die Kennzeichnung von Speicherbereichen als ausführbar und nicht-ausführbar kann als Abwendung der von-Neumann-Architektur gesehen werden, Daten und Programmcode in einem gemeinsamen Speicher zu halten. Hierzu muss der verwendete Prozessor das sogenannte NX-Bit (no execute) in seiner Speicherverwaltungseinheit unterstützen.

Für den Fall, dass sich die veränderte Programmlogik schon im Programmcode der App befindet, hilft die „Address Space Layout Randomization“ (ASLR), bei der der Programmcode bei jedem Start an anderen, zufällig gewählten Adressen eingeblendet wird. Hierdurch hat der Angreifer keine feste Einsprungsadresse für seine Programmlogik mehr.

Aktuelle Versionen von Apple iPhone und Android benutzen sowohl ASLR und NX Bit.

Beide Methoden, ASLR und NX-Bit sind mit „return-oriented programming“ [10] zu umgehen, bei dem sich die abgeänderte Adresse in einem Codesegment befindet.

Einschätzung: In Android sind Exploits im Kernel möglich, bei Apple iPhone sind sie auf Applikationsebene und Betriebssystemebene möglich.

### ***14.5.6 Schutz der installierten Software***

Selbst wenn mittels symmetrischer Kryptographie die Verbindung abgesichert wurde, ist die App als Datei im Dateisystem ungeschützt gegen Veränderung durch Dritte. Im einfachsten Fall werden Angreifer den Maschinencode der App abändern.

Das führt dazu, dass die Logik der App verändert wird und beispielsweise weitere Funktionen aufgerufen werden. Ausspähen von persönlichen Daten, Abfangen von Passwörtern oder anderen Eingaben ist mit etwas Fachwissen problemlos möglich. Um sich dem zu widersetzen ist die einfachste Möglichkeit, die eigenen Apps zu signieren und diese mit einem fest eingebauten Public-Key zu verifizieren. Das ist nur beschränkt wirkungsvoll, denn der Public-Key kann auch ersetzt werden. Bei extern gespeicherten Schlüsseln kommt dieselbe Problematik auf: Ein Angreifer muss nur den Teil der Logik ändern, der den Schlüssel von dem externen Medium besorgt. Als weiteren Einstiegspunkt für Angriffe kann auch der Arbeitsspeicher dienen: Im Gegensatz zum Ändern des auf dem Dateisystem gespeicherten Programms wird in diesem Fall die Programmlogik im Code-Speicher des laufenden Prozesses geändert.

Prozesse können sich kaum vor solchen Veränderungen schützen, sondern sind gezwungen sich auf den Schutz des Betriebssystems zu verlassen. Daher ist es nötig, Schnittstellen zum Betriebssystem zu nutzen, damit dieses die Überprüfung ausführt.

Das Apple iOS bietet eine solche Schnittstelle: Programme können nur gestartet werden, wenn sie von Apple signiert wurden. Android hingegen ist weniger restriktiv: Unsignede Apps können ausgeführt werden, die Installation ist jedoch nur möglich, wenn im Konfigurationsmenü eine Einstellung angepasst wurde.

Ein Exploit kann jedoch den Schutz durch signierte Apps umgehen. So ist es dem IT-Sicherheitsspezialisten „Comex“ gelungen, einen Angriffsvektor für iOS 4.3.3 zu bilden, obwohl das iOS mit ASLR und NX-Bit geschützt ist [11].

Einschätzung: Die Signatur von Software kann umgangen werden.

### **14.5.7 Schutz bei physikalischem Zugriff**

Weil das Problem der Entwendbarkeit auch schon für mobile Telefone besteht, werden die bei mobilen Telefonen benutzten Schutzmechanismen auch für Smartphones verwendet. Hier sind zum einen Geräteworte zu nennen, ohne die das Smartphone den Betrieb verweigert, zum anderen die PIN, mit der die Zugangsschlüssel des Netzwerkbetreibers geschützt sind.

Ein weiterer Aspekt bei physikalischem Zugriff ist der Zugriff auf das Dateisystem. Wie können sensitive Daten, wie zum Beispiel gespeicherte Passwörter gesichert werden, falls Zugriff auf das Dateisystem besteht?

Apple bietet zwar eine Datenbank zur Passwortsicherung, die mit Hilfe des Passcodes, mit dem der Benutzer das Gerät entsperrt, verschlüsselt ist. Bei der Standardlänge von 4 Ziffern stellt dies jedoch keine große Hürde dar. Die Bankapplikation iOutbank der Firma stoeger it GmbH hingegen speichert die Kontodaten des Anwenders beispielsweise in einer separaten Datenbank. Hierzu wird der Benutzer beim ersten Start der App aufgefordert, ein Passwort einzugeben. Zusätzlich bewertet die Applikation die Sicherheit des Passwortes. Dies ist ein guter und lobenswerter Ansatz, aber leider ein Einzelfall und außerdem nicht ganzheitlich genug: Ein

Auslesen des Schlüssels oder auch ein Abfangen der Eingabe des Passwortes wäre bei einem kompromittierten System trotzdem möglich.

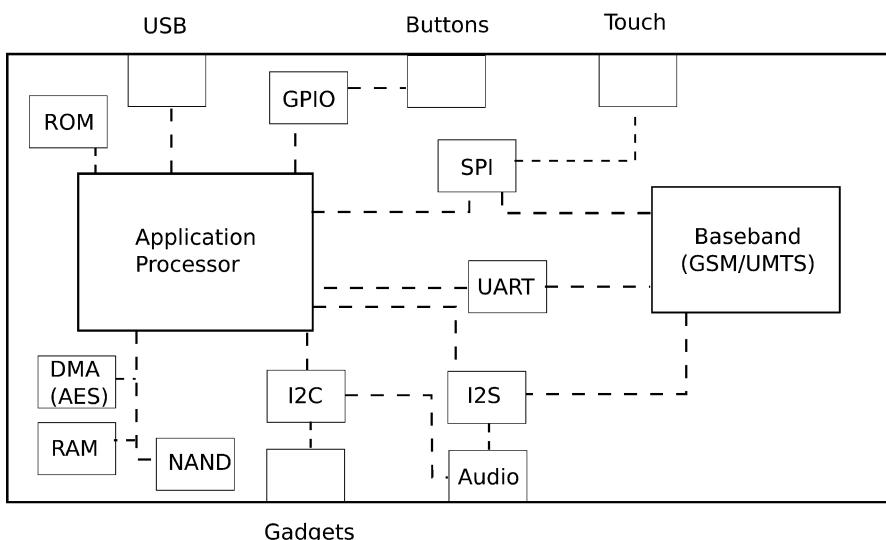
Einschätzung: Der Schutz durch Gerätewörter ist für einen hinreichend versierten Fachmann kein Hindernis.

### 14.5.8 Schutz der Hardwareplattform

Genauso wie bei den herkömmlichen Computern ist auch bei Smartphones die Hardware einer der elementarsten Bestandteile. Sie dient als Grundpfeiler für alles weitere, was auf ihr aufbaut. Jedes Betriebssystem muss perfekt an die Hardware angepasst sein, und das erfordert viel Aufwand auf der Seite der Betriebssystementwickler. Oft werden Hardwaretreiber vom Hersteller der Komponente erstellt und größtenteils vom Smartphonehersteller ohne Änderungen übernommen.

Mittlerweile basieren die meisten Smartphones auf der ARM-Prozessorarchitektur, dazu kommen Schnittstellen beispielsweise zur Kommunikation mit Computern, Einschübe zur Speicherplatzerweiterung oder sogar Video-Ausgänge mittels MHL (Mobile High-definition Link, sozusagen HDMI über bspw. Micro-USB), vgl. Abb. 14.2.

Eine der wichtigsten Schnittstellen, die zur Anbindung von Peripheriegeräten wie externen Festplatten, RS232-Konvertern und auch Smartphones benutzt wird, ist USB. Diese Flexibilität kann jedoch gefährlich werden. Mittels eines Computers und eines Datenkabels kann die 4. Hardwaregeneration des Apple iPhone durch einen Buffer-Overflow im Bootrom übernommen werden. Das Bootrom bietet einen



**Abb. 14.2** Schematischer Aufbau einer Smartphone-Hardware

Wiederherstellungsmodus, in welchem man mittels USB von Apple signierte Dateien ausführen kann. In dessen Treiber zur USB-Unterstützung ist jedoch ein Fehler in der Behandlung der darüber übertragenen Pakete. Nach Ausnutzen der Schwachstelle ist es möglich, eigenen Programmcode auf dem Prozessor laufen zu lassen. Das Smartphone ist dadurch ungeschützt vor Veränderungen.

Ähnliche Möglichkeiten gibt es auch bei Android-Smartphones, bei denen das sogenannte Rooten (erhalten von Administrator-Rechten) nicht ohne Umwege möglich ist. HTC will jedoch nun andere Wege gehen und wird in zukünftigen Geräten die Bootloader-Sperre entfernen. Viele Kunden wünschten sich die Möglichkeit, ihr Smartphone-Betriebssystem modifizieren zu können. Daraufhin ging der Hersteller auf diesen Wunsch ein.

Um die Daten vor Zugriff zu schützen, gibt es seit Android 3.0 „Honeycomb“ die Möglichkeit, die Daten zu verschlüsseln. Den Quellcode hat Google bisher noch nicht freigegeben. Dadurch ist es schwer nachzuvollziehen, wie die Daten wirklich gesichert werden.

Apple hat sich als Zugriffsschutz etwas Ausfalleneres überlegt: eine komplette Verschlüsselung des NAND-Flashspeichers. Die Geräte verfügen über einen AES-Chip, der fester Bestandteil des Direct Memory Access (s. Abb. 14.2) ist. Der Chip enthält zwei Krypto-Schlüssel. Ein Schlüssel (UID) ist einzigartig und nur auf dem Gerät vorhanden, der zweite Schlüssel (GID) ist pro Produktreihe einheitlich. Der UID-Key wird hierfür bei Datenverschlüsselung zur Generierung verschiedener Schlüssel benutzt, die in verschiedenen logischen Schichten des Datenträgers gespeichert sind. Diese Bereiche sind wiederum auch verschlüsselt. Die Metainformationen des von Apple benutzen HFS+-Dateisystems sind mit einem der generierten Keys verschlüsselt, die Dateien selber mit einem anderen. Hier ist noch die Besonderheit, dass pro Datei ein anderer Schlüssel verwendet wird. Die „Extended Attributes“ in den Metainformationen der Datei enthält den Schlüssel. Dieser ist das Produkt eines AES Key Wrap mit einem der generierten Schlüssel und einer Zufallszahl.

Trotz der Komplexität des Verfahrens, welches von Apple selbst nicht veröffentlicht wurde, war es für eine Gruppe von Security-Spezialisten möglich dieses Verfahren zu erforschen. Jean-Baptiste Bédrune sowie Jean Sigwald Sogeti haben sich damit befasst und ihre Ergebnisse im Vortrag „iPhone Data Protection in Depth“ zusammengefasst [12]. Nun kann man mit einer Open Source Software (OpeniBoot [13]) die Verschlüsselung ohne weitere Voraussetzungen ausschließen und alle Dateien auslesen. Der größte Vorteil des Verfahrens ist es, dass man nur einen Bereich mit gespeicherten Keys löschen muss, um ein Auslesen zu verhindern. Das wird als Remote Wipe im mobileMe Service von Apple angeboten.

Einschätzung: Selbst verschlossen gehaltene Sicherheitsmaßnahmen bieten keinen absoluten Schutz. Zudem können verschlossen gehaltene Sicherheitsmaßnahmen nicht verifiziert werden. Remote Wipe funktioniert nur, wenn das iPhone Netzempfang hat. Der Angreifer kann einfach den Netzempfang unterbinden und hat damit genügend Zeit, das Dateisystem auszulesen.

## 14.6 Fazit

Im Titel dieses Beitrags wurde das Smartphone mit einer Burg verglichen. In einer mittelalterlichen Burg gibt es zwei ineinander verschachtelte Sicherheitsbereiche, die äußere und die innere Burg. Angreifer können versuchen einzelne Steine aus der Burg auszuhebeln. Aber die Angreifer können auch versuchen, die zum Betrieb der Burg notwendigen Verbindungen zur Außenwelt auszunutzen. Die Geschichte zeigt, dass durch technische Neuerungen das Konzept der Burg überholt ist.

Smartphone Security besteht aus mehreren Komponenten, die abgesichert werden müssen. Aktuell gibt es zwar verschiedene Konzepte, die in aktuellen Geräten verwendet werden. Diese sind jedoch noch nicht ausgereift und bieten Angriffsvektoren.

Zukünftig sollten mehr Ressourcen zur Absicherung von Smartphones zur Verfügung gestellt werden. In diesem Bereich sind umfangreiche Untersuchungen möglich und nötig, um zukünftige Anwendungen sicher zu gestalten. Ein Sicherheitskonzept, welches alle Bestandteile eines Smartphones einbezieht, ist nötig.

Hierbei spielen auch Hersteller der Hardwarekomponenten eine Rolle. Aktuell werden Treiber und Datenblätter einzelner Komponenten hauptsächlich an die Hersteller der Smartphones verteilt. Da dadurch nur eine geringe Anzahl von Personen Einblick in den Quellcode hat, werden Fehler kaum entdeckt. Zudem können die Konzepte nicht verifiziert werden. Einsehbare Konzepte und ein offener Quellcode spornen zur Kollaboration an. Offenheit vermindert nicht die Sicherheit, sondern hilft, die Sicherheit zu erhöhen.

## Literaturverzeichnis

1. Charakterisierung eines Smartphones bei Wikipedia.  
<http://de.wikipedia.org/wiki/Smartphone>, aufgerufen am 14. Juli 2011
2. Wachstumsprognose für Apps.  
<http://www.idc.com/research/viewdocsynopsis.jsp?containerId=225668>, aufgerufen am 14. Juli 2011
3. Marktübersicht Smartphone. <http://www.canalys.com/pr/2011/r2011013.html>, aufgerufen am 13. Juli 2011
4. Open Handset Alliance. <http://www.openhandsetalliance.com/>, aufgerufen am 13. Juli 2011
5. OS/ISO Schichtenmodell, ISO 7498-1, Information technology – Open Systems Interconnection – Basic Reference Model: The basic model. [http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269\\_ISO\\_IEC\\_7498-1\\_1994\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip), aufgerufen am 13. Juli 2011
6. Weinmann R-P (2010) The Baseband Apocalypse. Vortrag von dem 27. Chaos Communication Congress. Berlin, Deutschland
7. Bundesamt für Sicherheit in der Informationstechnik (2010) Update: Neue Version der AusweisApp wird überprüft. [https://www.bsi.bund.de/ContentBSI/Presse/Pressemitteilungen/Presse2010/AusweisApp\\_181110.html](https://www.bsi.bund.de/ContentBSI/Presse/Pressemitteilungen/Presse2010/AusweisApp_181110.html). aufgerufen am 16. Juni 2011.
8. SSL-GAU.  
<http://blog.mozilla.com/security/2011/03/22/firefox-blocking-fraudulent-certificates>, aufgerufen am 14. Juli 2011

9. Heider J (2011) Gegenüberstellung der technischen iPhone- und Android-Sicherheit. Vortrag von der 54. DFN-Betriebstagung. Berlin, Deutschland.
10. Buchanan E, Roemer R, Savage S (2008) Return-Oriented Programming: Exploits Without Code Injection, Given at Black Hat USA 2008 Briefings. <http://cseweb.ucsd.edu/~hovav/talks/blackhat08.html>, aufgerufen am 14. Juli 2011
11. Ankündigung des iOS 4.3.3 root-Exploits.  
<https://twitter.com/comex/status/87245401899536384>
12. Bédruine J-B, Sogeti J S (2011) iPhone Data Protection in Depth. Vortrag von der HITB-SecConf 2011. Amsterdam, Niederlande.
13. OpeniBoot. <http://www.idroidproject.org>, aufgerufen am 14. Juli 2011

---

# Kapitel 15

## Hochsichere Smartphones mit L4Android

Steffen Liebergeld, Matthias Lange, Adam Lackorzynski und Alexander Warg

**Zusammenfassung** Smartphones vermitteln uns das Gefühl rund um die Uhr vernetzt zu sein. Für viele Anwender sind Smartphones das primäre Kommunikationsgerät geworden. Große Firmen investieren mittlerweile viel Geld, um mit Smartphones neue Geschäftsfelder zu erschließen. Die neuen Anwendungen, wie z. B. Mobile Payment, haben hohe Anforderungen an die Gerätesicherheit. Vorhandene Smartphonebetriebssysteme sind diesen Anforderungen nicht gewachsen. In dieser Arbeit stellen wir L4Android vor. L4Android ist ein Betriebssystemframework, welches das Erstellen sicherer Smartphonearchitekturen ermöglicht. Wir verwenden einen modernen Mikrokern und erlauben das Ausführen von nicht sicherheitskritischer Software in einer virtuellen Maschine.

### 15.1 Einführung

Smartphones sind eine neue Klasse von mobilen Endgeräten. Mit ihren vielfältigen Schnittstellen vermitteln sie das Gefühl ständiger Erreichbarkeit. Viele Nutzer ver-

---

Steffen Liebergeld  
Security in Telecommunications  
Technische Universität Berlin und Deutsche Telekom Laboratories  
E-mail: steffen@sec.t-labs.tu-berlin.de

Matthias Lange  
Security in Telecommunications  
Technische Universität Berlin und Deutsche Telekom Laboratories  
E-mail: mlange@sec.t-labs.tu-berlin.de

Adam Lackorzynski  
Technische Universität Dresden, Professur für Betriebssysteme,  
E-mail: adam@os.inf.tu-dresden.de

Alexander Warg  
Technische Universität Dresden, Professur für Betriebssysteme,  
E-mail: warg@os.inf.tu-dresden.de

wenden Smartphones als ihr zentrales Kommunikationsgerät für Telefonie, E-Mail, SMS, MMS und zum Surfen im Internet. Zudem lassen sich Smartphones durch das Installieren sogenannter Apps an die Bedürfnisse der Anwender anpassen und erweitern.

Große Unternehmen wie Google, Microsoft und Apple haben das Potenzial von Smartphones für die Erschließung neuer Geschäftsfelder erkannt. Diese Firmen investieren viel Geld in die Erforschung neuer Anwendungsgebiete. Der sich abzeichnende Trend geht dabei hin zu Anwendungen, die besondere Anforderungen an die Sicherheit des Smartphones stellen. Beispiele hierfür sind Mobile Payment, Ticketing sowie Banking Apps.

Üblicherweise ist es die Aufgabe des Betriebssystems, für die Sicherheit des Systems zu sorgen. Alle aktuellen Smartphonebetriebssysteme implementieren im Grunde dieselbe Architektur. Betrachtet man den Marktanteil, dann ist Android inzwischen das populärste Smartphonebetriebssystem [1]. Es wird von einer Allianz aus Google und der Open Handset Alliance gemeinsam als Open-Source-Projekt entwickelt. Damit besteht allgemeiner Zugang zu den Android-Quelltexten, was tiefgreifende Analysen erlaubt. Darum haben wir uns entschieden, Android exemplarisch zu betrachten. Unsere Lösung ist allgemeingültig und auf andere Smartphonebetriebssysteme anwendbar.

Leider ist Android von zahlreichen Sicherheitsproblemen geplagt. So müssen beispielsweise Sicherheitsupdates durch die Gerätehersteller zur Verfügung gestellt werden, welche dies, wenn überhaupt, meist nur mit deutlicher Verzögerung tun. Des Weiteren schützt das Rechtemodell nur unzureichend vor aktueller Schadsoftware.

Vielfältige wissenschaftliche Arbeiten versuchen die Sicherheit von Android zu erhöhen [2–8]. Alle vorgeschlagenen Lösungen können jedoch umgangen oder außer Kraft gesetzt werden, wenn es einem Angreifer gelingt, den Betriebssystemkernel zu übernehmen. In diesem Fall spricht man davon, dass das Gerät gerootet wird. Rooten kann dabei durch den Anwender direkt geschehen. Der Kernel wird gegen eine angepasste Version ausgetauscht, um zum Beispiel weitere Funktionen freizuschalten. Rooten kann aber auch durch Schadsoftware geschehen, welche eine Sicherheitslücke des Kernels ausnutzt. Dass dies eine reale Gefahr ist, zeigt eine kürzlich durchgeführte Studie von Coverity, bei der in einem Android-Kernel 88 kritische Sicherheitslücken gefunden wurden [9].

In diesem Beitrag stellen wir ein Framework vor, mit dessen Hilfe sich sichere Softwareumgebungen auf Smartphones erstellen lassen. Dabei wird Android als unsichere Komponente angesehen, welche gekapselt in einer virtuellen Maschine ausgeführt wird. Sicherheitskritische Anwendungen werden von Android isoliert direkt auf dem Mikrokern ausgeführt. So kann ein kompromittiertes Android die Integrität, Vertraulichkeit und Verfügbarkeit von sicheren Anwendungen nicht beeinträchtigen. Unser Framework bildet die Basis für zukünftige sicherheitskritische Anwendungen von Smartphones.

## 15.2 Technische Grundlagen

In diesem Abschnitt erläutern wir zunächst die Grundlagen der Mikrokerntechnologie. Für die folgenden Abschnitte ist es zudem wichtig, die grundlegende Arbeitsweise eines modernen Smartphones zu verstehen, welche wir in Abschn. 15.2.2 beschreiben.

### 15.2.1 Mikrokerne

Der Betriebssystemkern spielt eine besondere Rolle. Er läuft in einem privilegierten Modus, dem sogenannten *Kernelmodus*, des Prozessors. Anwendungen hingegen werden mit weniger Privilegien im *Benutzermodus* ausgeführt.

Da der Kernel mit den höchsten Privilegien läuft, müssen Funktionen, die unmittelbare Auswirkungen auf die Sicherheit haben, im Kernelmodus implementiert werden. Zu den grundlegenden Mechanismen des Kernels zählt das Erzeugen von Schutzdomänen (Adressräumen), das Zuweisen von Prozessorzeit an Aktivitäten (Scheduling) und ein Kommunikationsmechanismus zwischen Schutzdomänen (Inter-Prozess-Kommunikation, IPC).

Implementiert ein Betriebssystemkern zusätzlich zu den oben genannten Mechanismen weitere Funktionen wie zum Beispiel Dateisysteme, Protokollstacks und Gerätetreiber, spricht man von einem *monolithischen* Kern. Linux ist ein typisches Beispiel für diese Kernel-Architektur. Der Nachteil dieser Architektur ist, dass aufgrund der Vielzahl an Funktionen die Abhängigkeiten innerhalb des Kernels eine nicht zu unterschätzende Komplexität ergeben. Innerhalb des Kernelmodus gibt es zwischen einzelnen Komponenten keine Isolationsmöglichkeiten. Enthält eine Kernelkomponente einen Fehler, welcher von einem Angreifer ausgenutzt werden kann, so versetzt sie den Angreifer in die Lage, den Speicher aller anderen Kernelkomponenten manipulieren zu können. Der Angreifer kann zum Beispiel Sicherheitsmechanismen wie *Mandatory Access Control* (MAC) außer Kraft setzen, oder direkt kryptografisches Schlüsselmaterial von Nutzerprogrammen stehlen.

Im Gegensatz zu monolithischen Betriebssystemkernen implementieren Mikrokerne nur die oben erwähnten grundlegenden Mechanismen. Gerätetreiber, Dateisysteme und Protokollstacks werden als Anwendungen im Benutzermodus implementiert. Dadurch lässt sich die Komplexität des Kernels drastisch reduzieren. Mit Hilfe von Adressräumen werden die Anwendungen voneinander isoliert. Erfolgreiche Angriffe können damit auf die betreffende Komponente beschränkt werden.

### 15.2.2 Architektur eines Smartphones

Smartphones bestehen üblicherweise aus zwei individuellen Subsystemen. Diese sind physisch voneinander isoliert und nur über wenige wohldefinierte Schnittstellen

len miteinander verbunden. Eines dieser Systeme ist der sogenannte Anwendungsprozessor. Auf ihm laufen das Smartphonebetriebssystem und die Anwendungen. Das zweite System ist das Baseband, welches die Schnittstelle zum Mobilfunknetzwerk implementiert.

Der Anwendungsprozessor wird als sogenannter System-on-a-Chip (SoC) verbaute. Hierbei werden Prozessor, Speichercontroller, Interruptcontroller, Timer und weitere Funktionseinheiten integriert auf einem gemeinsamen Siliziumchip verbaut. Diese Bauweise spart zum einen Kosten, und zum anderen lassen sich so ausgeklügelte Stromsparmechanismen implementieren. Zu den auf dem SoC integrierten Funktionseinheiten gehören ein Graphikprozessor, Einheiten für die Audio- und Videoverarbeitung sowie Controller für die zahlreichen Systembusse wie zum Beispiel SPI, I2C und USB. Über die Systembusse werden weitere Peripheriegeräte wie zum Beispiel der GPS-Sensor und Speicherkarten an das System angeschlossen.

## 15.3 Bedrohungen für das Android-Betriebssystem

Das Android-Betriebssystem ist verschiedenen Bedrohungen ausgesetzt, die es zum Ziel von Angriffen durch Schadsoftware machen. Im Vergleich zu Desktopbetriebssystemen wie zum Beispiel Windows ist die Anzahl an Angriffen und Schadsoftware für Android momentan noch gering. Jedoch spricht vieles dafür, dass mit weiterer Verbreitung von Smartphones auch die Attraktivität für Angriffe auf diese Systeme steigen wird [10].

Wir haben vier Bereiche identifiziert, in denen Android angreifbar ist. Diese stellen wir in den folgenden Abschnitten jeweils kurz vor.

### 15.3.1 Komplexe Softwarebasis

Ein Maß, mit dem die Sicherheit von Software bewertet werden kann, ist die sogenannte Trusted Computing Base (TCB). Darunter versteht man den Teil des Softwarestacks, dem man als Anwender für die sichere und korrekte Ausführung einer Anwendung vertrauen muss. Der Betriebssystemkern zählt aufgrund seiner speziellen Berechtigungen immer zur TCB.

Android basiert auf dem Linux-Kernel. Aktuelle Versionen des Linux-Kernels bestehen inzwischen aus mehr als 15 Millionen Zeilen Code (Source Lines Of Code, SLOC). Bei dieser Komplexität sind Implementierungsfehler nicht zu vermeiden. Einen Hinweis auf die Fehleranfälligkeit des Android-Kernels liefert eine aktuelle Studie von Coverity Inc. [9], welche mit einer automatisierten statischen Analyse 88 sicherheitskritische Fehler gefunden hat.

Für Android verwendet Google nicht den allgemeinen Linux-Kernel. Die Anforderungen von mobilen Endgeräten bezüglich Stromsparmechanismen, Speicherverbrauch und Netzwerksicherheit werden durch den allgemeinen Linux-Kernel nur

unzureichend erfüllt. Daher hat sich Google entschieden, eine abgeleitete Variante zu entwickeln, die die genannten Probleme löst. Diese Anpassungen sind bislang nur teilweise in den Hauptentwicklungszweig von Linux zurückgeflossen. Daher muss Google allein die Anpassungen für neuere Linux-Kernel pflegen.

Hinzu kommt, dass Hersteller von mobilen Endgeräten den Android-Kernel für ihre Geräte weiter anpassen. Um sich voneinander zu differenzieren, verbauen die Hersteller unterschiedliche Peripheriegeräte, für die sie eigene Treiber entwickeln. Oftmals wird der Code für die Treiber nicht an die Gemeinschaft der Linux-Entwickler zurückgegeben, so dass diese Treiber auch nicht den Begutachtungsprozess der Entwicklergemeinde durchlaufen. Daher weisen diese Treiber oft eine schlechte Qualität auf. Zudem sind damit wieder die Hersteller gefordert, Treiber für neuere Linux-Versionen nachzuportieren. Wenn das Gerät einmal verkauft ist, wird dieser zeit- und kostenintensive Vorgang häufig vermieden, wobei die Anwender mit veralteter und potentiell fehlerträchtiger Software zurückbleiben müssen.

### ***15.3.2 Verzögertes Schließen von Sicherheitslücken***

Die Komplexität und die damit verbundenen Probleme des Linux-Kernels sind nur ein Aspekt für die Anfälligkeiten von Android. Android ist ein Open-Source-Projekt. Neben von Google speziell für Android entwickelten Komponenten werden viele unterschiedliche Open-Source-Projekte, wie zum Beispiel WebKit, in Android verwendet. Für Open-Source-Projekte ist es charakteristisch, dass sie von einer weltweit verteilten Gemeinschaft entwickelt werden. Die Kommunikation zum Entwicklungsprozess läuft über öffentlich zugängliche Wikis, Code-Repositories, Bug-Tracker und Mailinglisten.

Der Zeitraum zwischen dem Bekanntwerden einer Sicherheitslücke bis zum Einspielen des Patches, und damit dem Schließen der Lücke auf dem Endsystem, ist kritisch für die Systemsicherheit, denn während dieses Zeitraums sind die Systeme angreifbar. Üblicherweise ist die Zeitspanne zwischen dem Bekanntwerden einer Sicherheitslücke und der Bereitstellung eines Patches bei Open-Source-Projekten recht kurz. Viele große Linux-Distributionen haben einen schnellen Update-Mechanismus integriert, wodurch die Zeitspanne vom Bereitstellen bis zum Einspielen des Patches minimiert wird. Einen solchen Mechanismus gibt es für Android nicht. Damit bleiben einmal entdeckte Sicherheitslücken unnötig lange offen. Ein Angreifer kann sich aus den oben erwähnten Code-Repositories und Bug-Trackern aktuelle Sicherheitslücken erschließen und kann darauf bauen, dass die Mehrzahl der Geräte den entsprechenden Sicherheitspatch nicht eingespielt hat.

Das Problem wird durch den unregelmäßigen Entwicklungszyklus von Google noch verschärft. Android wird von Google hinter verschlossenen Türen entwickelt. Die Quellen für Updates werden nur zu bestimmten Zeitpunkten veröffentlicht. Da viele Hersteller, um sich von Wettbewerbern zu differenzieren, spezifische Anpas-

sungen, wie zum Beispiel spezielle Benutzeroberflächen, für ihre Geräte implementieren, müssen diese Anpassungen jedes mal aufwändig an neue Android-Versionen angepasst werden.

Problematisch ist auch, dass Subsysteme von Android nicht individuell aktualisiert werden können. Für Aktualisierungen muss ein vollständiges System-Image geladen und installiert werden. Viele Nutzer empfinden diese Unterbrechung bei der Nutzung ihres Geräts als störend. Daher neigen Hersteller dazu, unterschiedliche Updates zu kumulieren und dann gebündelt bereitzustellen.

Verschärfend ist, dass es teilweise für ältere Geräte überhaupt keine Updates mehr gibt, da Google die Entwicklung älterer Android-Versionen bereits eingestellt hat. Das ist insofern problematisch, weil neuere Android-Versionen meist auch erhöhte Hardwareanforderungen haben und somit neue Smartphones voraussetzen. Aktuelle von Google erhobene Marktdaten [11] zeigen, dass die im Dezember 2010 veröffentlichte Android-Version 2.3 erst von weniger als 5 Prozent der Anwender eingesetzt wird. Mehr als 90 Prozent der Benutzer verwenden weiter die Versionen 2.1 und 2.2, welche bereits mehr als ein Jahr alt sind.

### **15.3.3 Bekannte Sicherheitsprobleme von Android**

In Android ist ein Zugriffskontrollsystem in Form eines Rechte mechanismus implementiert. Bei der Installation einer Applikation kann diese die Berechtigungen für den Zugriff auf bestimmte Systemressourcen wie zum Beispiel Ortsinformationen oder Zugang zum Internet vom Benutzer erfragen. Android zeigt einen Dialog mit den erfragten Zugriffsrechten an, und der Nutzer kann entweder dieser Auswahl zustimmen und mit der Installation fortfahren oder den Installationsvorgang abbrechen. Das gezielte Aus- und Abwählen einzelner Rechte ist nicht möglich. Daher akzeptieren viele Benutzer blind die gewünschte Rechteauswahl, ohne sich über die Auswirkungen Gedanken zu machen.

Ein weiteres Problem mit dem Rechtesystem ist, dass es nicht feingranular genug ist [12]. Eine Anwendung, die auf das Internet zugreifen kann, kann mit jedem beliebigen Server kommunizieren. Hat diese Anwendung zum Beispiel auch Zugriff auf das interne Adressbuch von Android, so kann nicht verhindert werden, dass diese Anwendung die Kontaktdatenbank an einen beliebigen Server im Internet verschickt.

In der Arbeit von Enck et al. [4] zeigten zwei Drittel der untersuchten Anwendungen verdächtiges Verhalten bei der Verwendung von privaten Daten. Eine statische Codeanalyse deckte bei weiteren Anwendungen potentielle Datenlecks auf [13].

Ein Großteil der für Android verfügbaren Anwendungen wird über den Google Marketplace vertrieben. Der Marketplace soll unter anderem die Verbreitung von Malware verhindern, da Google dort eingestellte Anwendungen zumindest rudimentär auf Unbedenklichkeit testet. Trotzdem schafften es in letzter Zeit vermehrt Anwendungen mit Schad routinen in den Marketplace [14, 15].

### 15.3.4 Rooten der Geräte

Unter *Rooten* versteht man den Prozess, bei dem der Benutzer zusätzliche Zugriffsrechte auf seinem Telefon erhält. Da hierbei zahlreiche Sicherheitsmechanismen außer Kraft gesetzt werden, stellen geroottete Telefone ein ernst zu nehmendes Sicherheitsrisiko dar. Die Hürde für die Verbreitung von Trojanern oder Botnetzen ist damit erheblich niedriger. Das ist insbesondere bedeutend, weil zwei namhafte Hersteller die Funktion zum Rooten explizit als Vorteil ihrer Geräte bewerben [16, 17].

Das Rooten kann auf zweierlei Weise geschehen. Zum einen absichtlich durch den Benutzer, der zum Beispiel zusätzliche Anwendungen installieren möchte. Hierbei wird üblicherweise eine modifizierte Firmware, inklusive Kernel, installiert, die die gewünschten Funktionen freischaltet. Zum anderen können Geräte durch Schadsoftware wie zum Beispiel DroidDream [18] geroottet werden. Dazu wird eine bekannte Sicherheitslücke im Kernel ausgenutzt und damit die Sicherheitsbarriere des Android Betriebssystems gebrochen.

## 15.4 Softwarearchitektur für sichere Smartphones

Die monolithische Architektur von Android ist die Hauptursache für die in Abschn. 15.3 vorgestellten Sicherheitsprobleme. Die Softwarearchitektur unseres Frameworks fußt daher auf dem Prinzip des „Teile und herrsche“. Kleine isolierte Komponenten implementieren jeweils einen Dienst. Umfangreichere Funktionalität wird mit Hilfe mehrerer miteinander kommunizierender Komponenten implementiert.

Die Aufgabe, ein vorhandenes monolisches System, wie z. B. Android, abzusichern, indem man es in kleine, unabhängige und isolierte Komponenten zerlegt, ist eine umfangreiche Aufgabe. Vorhandene Komponenten haben komplexe Abhängigkeitsbeziehungen untereinander, welche sich schwierig auflösen lassen. Stattdessen akzeptieren wir die Sicherheitseigenschaften der existierenden Systeme, da sie für existierende Anwendungen ausreichend sind. Für Anwendungen mit höheren Sicherheitsanforderungen schaffen wir eine eigene Ausführungsumgebung. Dazu verwenden wir in unserem Framework virtuelle Maschinen (VM), um existierende Smartphonebetriebssysteme zu kapseln. Sicherheitskritische Anwendungen laufen außerhalb der VM. Im Falle eines erfolgreichen Angriffs auf eine VM bewahren diese Anwendungen ihre Integrität.

### 15.4.1 Isolation von Komponenten

Ziel einer sicheren Softwarearchitektur ist es, die unterschiedlichen im System laufenden Komponenten zu isolieren. Diese Isolation muss sowohl zeitlich als auch räumlich erfolgen. Für räumliche Isolation wird der von der Hardware zur Verfügung gestellte Mechanismus für virtuellen Speicher verwendet. Zeitliche Isolation

wird durch einen unterbrechenden Scheduler implementiert. Dieser sorgt dafür, dass jede Anwendung für Berechnungen einen Anteil an der Prozessorzeit erhält.

### **15.4.2 Interaktion zwischen Komponenten**

Um sinnvoll ein System bauen zu können, müssen die voneinander isolierten Komponenten miteinander kommunizieren, um zum Beispiel Daten auszutauschen, Dienste anderer Komponenten in Anspruch zu nehmen, oder zur Synchronisierung. Wir modellieren in unserem System Dienste als Objekte. Diese Objekte werden von Komponenten implementiert. Die Zugriffsrechte auf ein Objekt werden durch ein Capability-System verwaltet. Besitzt eine Komponente eine Capability, dann kann sie mit dem dazugehörigen Objekt kommunizieren. Es spielt keine Rolle, wo das Objekt implementiert ist, denn der Aufruf der Capability ist transparent für den Aufrufer. Weitere Objekte sind für einen Aufrufer nicht sichtbar, solange er keine weiteren Capabilities dafür besitzt. Damit kann man das Principle of Least Authority (POLA) umsetzen, welches besagt, dass eine Komponente nur mit den Rechten ausgestattet werden soll, die sie für die Erfüllung ihrer Aufgabe benötigt [19].

### **15.4.3 Kleiner Betriebssystemkern**

Wie im Abschn. 15.3.1 erläutert, ist es erstrebenswert, die TCB für sicherheitsrelevante Funktionen so klein wie möglich zu halten. Der Kernel ist die einzige Komponente im System, die mit den höchsten Privilegien läuft. Er sorgt für die Isolation der auf ihm laufenden Komponenten.

Bei der Entwicklung von Mikrokernen wird danach gestrebt, nur Funktionalitäten, welche für die Aufrechterhaltung der Plattform-Sicherheit benötigt wird, im Betriebssystemkern zu implementieren. Alle weiteren Funktionalitäten werden in Komponenten im Benutzermodus implementiert. Durch diesen Ansatz haben Mikrokerne eine wesentlich geringere Komplexität als monolithische Kerne. Davon profitiert auch die TCB der laufenden Anwendungen. Diese müssen nicht auf die korrekte Implementierung von Funktionen vertrauen, die für ihr Anwendungsszenario nicht benötigt werden.

#### **15.4.3.1 Sichere Ressourcenweitergabe durch Mapping**

In einem System mit vielen Komponenten spielt die Interaktion zwischen ihnen eine große Rolle. Eine wichtige Frage dabei ist, wie Zugriffsrechte auf Ressourcen zwischen Komponenten weitergegeben werden können.

Die Operation, mit der Zugriffsrechte weitergegeben werden können, wird *Map* genannt. Dieser Vorgang kann durch die *Unmap*-Funktion rückgängig gemacht wer-

den. Damit wird ein Zugriffsrecht rekursiv aus allen Komponenten entfernt, in denen es eingeblendet ist.

Mit Map und Unmap wird das Principle of Privilege Attenuation (POPA) umgesetzt. POPA sagt aus, dass ein Sender immer nur die Rechte (oder ein Subset davon) an einem Objekt weitergeben kann, die er selbst besitzt [19]. Eine Speicherseite, die beim Sender als schreibbar eingeblendet ist, kann also schreibbar oder nur lesbar an einen Empfänger weitergegeben werden. Eine nur lesbare Seite kann auch nur lesbar an einen Empfänger gegeben werden.

### ***15.4.4 Virtualisierung als sicherer Container***

Neben den Anwendungen mit erhöhten Sicherheitsanforderungen laufen auf einem Smartphone auch Applikationen, deren Anforderungen an die Sicherheit nicht so hoch sind. Zudem gilt es bei der Neuentwicklung eines Systems, den Aufwand und die Kosten für die Neuentwicklung beziehungsweise Portierung von existierender Software mit dem späteren Nutzen abzuwägen. Unter der Berücksichtigung, dass sich solche Systeme ständig weiterentwickeln, ist eine Neuimplementierung zeitlich oft nicht sinnvoll. Daher ist die Weiterverwendung nicht sicherheitskritischer Anwendungen aus aktuellen Smartphonebetriebssystemen zwingend notwendig.

In unserem Framework verwenden wir virtuelle Maschinen, um vollständige Standardbetriebssysteme in einer sicher gekapselten Umgebung auszuführen. Dafür verwenden wir das Rehosting, also die Portierung eines Betriebssystemkerns auf die Schnittstelle des Mikrokerns. Eine solche Portierung erfordert einige tiefgreifende Anpassungen am Kern des Gastsystems. Anwendungen laufen anschließend ohne Modifikationen auf dem portierten Kern. Rehosting benötigt keine besonderen Hardwareerweiterungen und ist somit auf alle aktuellen Smartphones anwendbar.

## **15.5 Implementierung des Prototypen**

In diesem Abschnitt beschreiben wir die Implementierung einiger wichtiger Komponenten unseres Systems.

### ***15.5.1 Mikrokern und Laufzeitumgebung***

Als Mikrokern verwenden wir den an der Technischen Universität Dresden entwickelten Mikrokern Fiasco.OC<sup>1</sup>. Fiasco.OC ist ein moderner Mikrokern der dritten Generation und implementiert das im Abschnitt 15.4.2 beschriebene Capability-

---

<sup>1</sup> <http://os.inf.tu-dresden.de/fiasco/>.

Modell. Der Mikrokern zeichnet sich durch eine breite Plattformunterstützung von ARM- und x86-basierten Systemen aus. Fiasco.OC ist durchgehend objektorientiert in der Programmiersprache C++ entwickelt worden.

Fiasco.OC wird auf Benutzerebene durch die Laufzeitumgebung L4Re<sup>2</sup> ergänzt. L4Re implementiert die Infrastruktur, die für die Entwicklung von Anwendungen benötigt wird. Die Schnittstellen für den Zugriff auf in L4Re implementierte Komponenten sind in Bibliotheken gekapselt. Zudem stellt L4Re Standardfunktionen wie zum Beispiel eine Teilmenge von POSIX, die pthread-Bibliothek sowie C- und C++-Bibliotheken bereit.

### 15.5.2 L4Android

Das L4Linux-Projekt<sup>3</sup> ist eine rehosted-Version von Linux, welche als Anwendung auf dem Mikrokern läuft. Im Zuge des L4Android-Projektes haben wir L4Linux um den entsprechenden Android-spezifischen Code erweitert. Dazu gehören unter anderem das IPC Framework *Binder* und der Powermanagementmechanismus *Wakelock*. Des Weiteren waren umfangreiche Konfigurations- und Testarbeiten notwendig. Der Quellcode des L4Android-Projektes ist als Open-Source-Software auf der Projektseite<sup>4</sup> verfügbar. Derzeitige Versionen von L4Android können den Android-Softwarestack in den Versionen Cupcake (2.1), Froyo (2.2) und Gingerbread (2.3) ausführen.

### 15.5.3 Prototyp

Wir haben das Framework jeweils auf einem x86-basierten Smartphone und einer ARM-Plattform als Prototyp implementiert.

Das ARM-basierte iMX.51-Board von Freescale verfügt über einen 800MHz schnellen Cortex-A8-Prozessor, der inzwischen in vielen günstigeren Smartphones verwendet wird. Der Prozessor kann auf 512 MB Arbeitsspeicher zurückgreifen. Auf dieser Plattform können wir ein voll funktionstüchtiges L4Android zeigen.

Eine weitere unterstützte Plattform ist ein Entwicklertelefon der Firma Aava. Diese setzt Intels Mobilprozessor Moorestown ein, welcher eine mit bis zu 1.6Ghz getaktete CPU enthält. Für dieses Telefon haben wir ein interaktives Demo-Setup, das ein L4Linux neben einem voll funktionstüchtigen L4Android ausführt (siehe Abbildung 15.1).

---

<sup>2</sup> <http://os.inf.tu-dresden.de/L4Re/>.

<sup>3</sup> <http://l4linux.org>.

<sup>4</sup> <http://l4android.org>.



**Abb. 15.1** L4Android zusammen mit einem L4Linux auf einem Aava-Entwicklersmartphone

## 15.6 Evaluierung

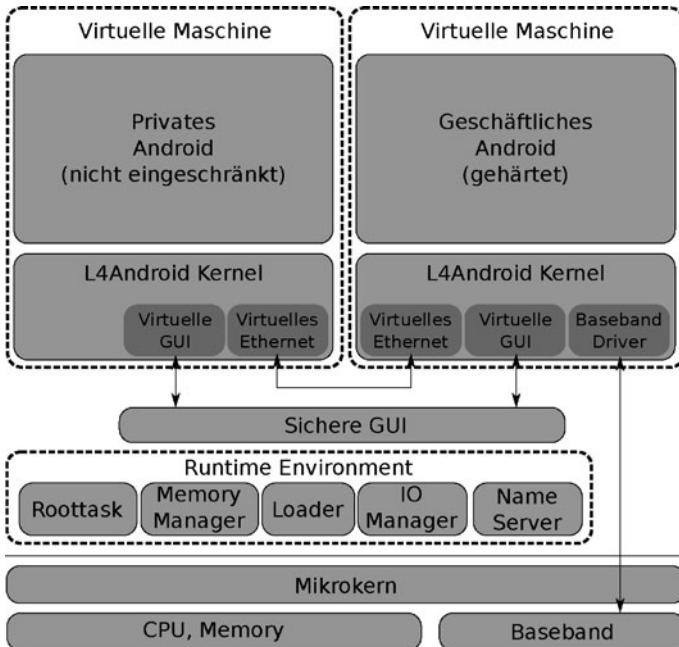
Wir haben unser Framework mit dem Ziel entwickelt, dass es für die Lösung einer breiten Palette an Sicherheitsszenarien geeignet ist. Um die Vielseitigkeit unseres Ansatzes aufzuzeigen, stellen wir nachfolgend zwei relevante Sicherheitsprobleme vor, die mit Hilfe unseres Frameworks gelöst werden können.

### 15.6.1 Vereinigung von geschäftlichem und privatem Mobiltelefon

Bei geschäftlich genutzten Smartphones wird besonderes Augenmerk auf die Sicherheit gelegt. Das System wird von der IT-Abteilung des Unternehmens provisoriert. Viele Funktionen, die die Leistungsfähigkeit eines Smartphones ausmachen, werden dabei stark eingeschränkt. So wird zum Beispiel die Installation nicht vertrauenswürdiger Software unterbunden oder der Netzzugang auf vertrauenswürdige Netze beschränkt. Diese Einschränkungen beeinträchtigen die Benutzbarkeit des Smartphones mitunter stark. Daher helfen sich viele Geschäftsleute damit, dass sie zwei Telefone mit sich führen: Das geschäftliche und das private Smartphone. Aus der Sicht des Anwenders ist es wünschenswert, nur ein Gerät zu haben, welches beide Aufgaben erfüllt.

Mit Hilfe unseres Frameworks lässt sich eine sichere Lösung entwickeln. Wie in Abb. 15.2 dargestellt, laufen zwei virtuelle Maschinen auf dem Mikrokern. In der für geschäftliche Zwecke von der IT-Abteilung bereitgestellten und provisionierten VM wird das Android durch zusätzliche Maßnahmen, zum Beispiel durch Anwendung von Mandatory Access Control (z. B. SELinux [20]), zusätzlich abgesichert.

Seite an Seite mit dem Geschäftsandroid läuft eine VM mit dem Android für private Zwecke. Dieses Android wird keinen Einschränkungen unterworfen und kann daher im vollen Umfang genutzt werden.



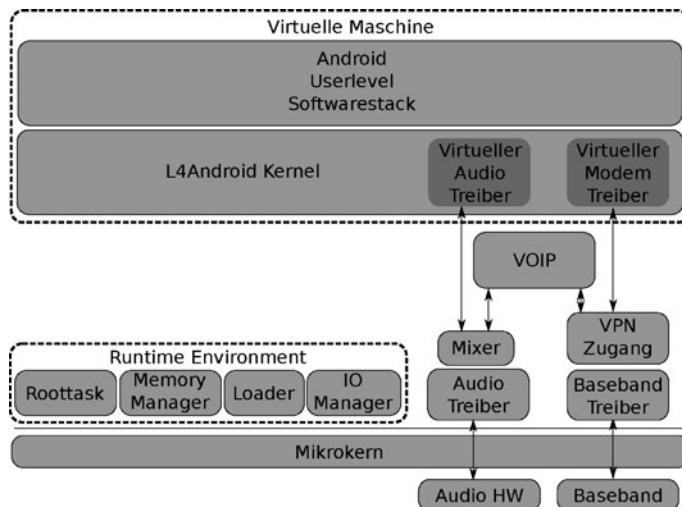
**Abb. 15.2** Architektur einer Smartphonelösung, welche ein privates und ein geschäftliches Android in virtuellen Maschinen auf einem Gerät ausführt

Der Mikrokern sorgt für die Isolation zwischen den VMs. Ein Angriff auf das private Android, welcher zum Beispiel durch Malware in einer App ausgeführt werden kann, hat keine Möglichkeit, auf die geschäftswichtigen Daten der Geschäfts-VM zuzugreifen.

### 15.6.2 Sichere mobile Kommunikation

Ein wichtiger Gesichtspunkt bei mobiler funkbasierter Kommunikation ist die Abhörsicherheit. Im Gegensatz zu kabelgebundener Kommunikation benötigt man beim mobilen Daten- und Sprachverkehr keinen physischen Zugriff auf das Übertragungsmedium. Es reicht aus, sich in Funkreichweite mit einer guten Antenne zu positionieren. Bereits vor einiger Zeit wurde die Verschlüsselung der GSM-Netze gebrochen [21]. Wenn Personen mit vertraulichen Daten umgehen, dann muss sichergestellt werden, dass diese Informationen nicht abgehört werden können.

Mit unserem Framework lässt sich eine Lösung für die genannten Probleme umsetzen. Die Architektur ist in Abb. 15.3 dargestellt. Die Datenkommunikation erfolgt über ein Virtual Private Network (VPN), welches an einem vertrauenswürdigen Endpunkt terminiert. Die Architektur sieht vor, dass sämtliche Kommunikation über das VPN abgewickelt wird. Android läuft in einer virtuellen Maschine und



**Abb. 15.3** Architektur eines abhörsicheren Smartphones. Der VPN Zugang erfolgt über eine Komponente, die von Android isoliert ausgeführt wird. Ein Angreifer hat demnach keine Möglichkeit, von einem kompromittierten Android an Schlüsselmaterial des VPN zu gelangen

hat keine Möglichkeit direkt auf die Hardware zuzugreifen. Android bekommt als einzige Kommunikationsmöglichkeit einen virtuellen Netzwerkzugang. Der virtuelle Netzwerkzugang ist als eigene Komponente implementiert. Damit ist zum einen sichergestellt, dass Android Daten ausschließlich über das VPN versenden kann, und zum anderen kann Android keine Informationen über kryptografische VPN-Schlüssel erfahren.

Wir unterbinden auch den Zugriff von Android auf das Mikrofon. Sprachkommunikation wird dabei über Voice-over-IP (VoIP) abgewickelt. Die VoIP-Komponente läuft außerhalb des Androids. VoIP-Datenpakete werden direkt über den virtuellen Netzwerkzugang verschickt, wodurch Android keine Möglichkeiten hat, Gespräche mitzuschneiden.

## 15.7 Zusammenfassung und Ausblick

In diesem Beitrag haben wir ein generisches Betriebssystemframework für mobile Endgeräte vorgestellt. Unser Framework basiert auf einem Mikrokern, der die starken Isolationseigenschaften, welche von sicherheitskritischen Anwendungen verlangt werden, implementiert. Vollständige Smartphonebetriebssysteme lassen sich in unserem Framework in virtuellen Maschinen weiterverwenden. Dabei wird die virtuelle Maschine von sicherheitskritischen Anwendungen isoliert und kann selbst im Falle eines Angriffs keinen Einfluss auf Integrität, Vertraulichkeit und Verfügbarkeit von sicheren Anwendungen nehmen. Anhand von zwei Szenarien haben

wir gezeigt, wie sich das Framework für die Lösung von Sicherheitsproblemen auf Smartphones einsetzen lässt.

Ein etwaiger Einsatz unseres Frameworks muss mit Hilfe des Geräteherstellers geschehen. So sind Treiber zu entwickeln bzw. zu portieren und es werden Eingriffe in den Bootprozess benötigt.

### 15.7.1 Ausblick

L4Android stellt den Ausgangspunkt für eine Vielzahl an möglichen Entwicklungen dar. Mögliche Entwicklungen können in zwei Gruppen eingeteilt werden. Die eine Gruppe beschreibt Anwendungen des Frameworks, und die andere Gruppe beschreibt Verbesserungen und Erweiterungen des Frameworks selbst.

Wir untersuchen derzeit, ob Rootkit-Detektoren, welche auf Virtual Machine Introspection basieren, auf unser Framework anwendbar sind. Damit könnte unser Framework auch einen Beitrag zu einer erhöhten Sicherheit des Smartphonebetriebssystems leisten. Des Weiteren erforschen wir, wie das Konzept der HoneyPots auf Smartphones angewendet werden kann. Dabei erlauben uns virtuelle Maschinen effizient Snapshots des Gastsystems anzulegen und den eingehenden und ausgehenden Datenstrom genau zu protokollieren.

Das Framework wird wie folgt erweitert werden. Wir entwickeln eine effiziente hardwarebeschleunigte Grafikvirtualisierung, welche eine signifikante Verbesserung des Ansprechverhaltens des Android-Systems erbringt. Zudem arbeiten wir an einer Optimierung der Virtualisierungsschnittstelle, die zukünftig eine Verringerung des Virtualisierungsoverheads verspricht. Dabei untersuchen wir, wie neuartige Prozessorerweiterungen wie Intel-VT und die Virtualisierungstechnik des angekündigten ARM-Cortex-A15-Prozessors effektiv eingesetzt werden kann. Ein wichtiges Forschungsfeld betrifft den Stromverbrauch unserer Lösung, sowie mögliche Optimierungen des Powermanagements, um die Batterieleistung optimal auszunutzen.

## Literaturverzeichnis

1. Canalys (2011) Android increases smart phone market leadership with 35 percent share. <http://canalys.com/pr/2011/r2011051.html> 26. Oktober 2011
2. Beresford AR, Rice A, Skehin N, Sohan R (2011) MockDroid: trading privacy for application functionality on smartphones. In: 12th Workshop on Mobile Computing Systems and Applications
3. Muthukumaran D, Sawani A, Schiffman J, Jung BM, Jaeger T (2008) Measuring integrity on mobile phone systems. In: Proceedings of the 13th ACM symposium on Access control models and technologies, ACM, New York, NY, USA, SACMAT '08, pp 155–164. DOI: <http://doi.acm.org/10.1145/1377836.1377862> 26. Oktober 2011
4. Enck W, Gilbert P, Chun B-G, Cox LP, Jung J, McDaniel P, Sheth AN (2010) TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In: Pro-

- ceedings of the 9th USENIX conference on Operating systems design and implementation, USENIX Association, Berkeley, CA, USA, OSDI'10, pp 1–6, <http://portal.acm.org/citation.cfm?id=1924943.1924971> 26. Oktober 2011
5. Portokalidis G, Homburg P, Anagnostakis K, Bos H (2010) Paranoid Android: versatile protection for smartphones. In: Proceedings of the 26th Annual Computer Security Applications Conference, ACM, New York, NY, USA, ACSAC '10, pp 347–356, DOI: <http://doi.acm.org/10.1145/1920261.1920313> 26. Oktober 2011
  6. Xie L, Zhang X, Seifert J-P, Zhu S (2010) pBMDS: a behavior-based malware detection system for cellphone devices. In: Proceedings of the third ACM conference on Wireless network security, ACM, New York, NY, USA, WiSec '10, pp 37–48, DOI: <http://doi.acm.org/10.1145/1741866.1741874> 26. Oktober 2011
  7. Zhou Y, Zhang X, Jiang X, Freeh VW (2011) Software Creates Privacy Mode To Help Secure Android Smartphones. <http://news.ncsu.edu/releases/wms-jiang-tissa/> 26. Oktober 2011
  8. Zhang X, Seifert JP, Acicmez O (2010) SEIP: Simple and Efficient Integrity Protection for Open Mobile Platforms. In: Information and Communications Security, Springer Berlin / Heidelberg, Lecture Notes in Computer Science, vol 6476, pp 107–125, [http://dx.doi.org/10.1007/978-3-642-17650-0\\_9](http://dx.doi.org/10.1007/978-3-642-17650-0_9) 26. Oktober 2011
  9. Coverity Inc (2010) Coverity Scan 2010 Open Source Integrity Report. <http://www.coverity.com/html/press/coverity-scan-2010-report-reveals-high-risk-software-flaws-in-android.html> 17. Mai 2011
  10. Becher M, Freiling FC, Hoffmann J, T Holz SU, Wolf C (2011) Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices. In: Proceedings of the IEEE Symposium on Security and Privacy
  11. Google Inc (2011) Distribution of Android Versions. <http://developer.android.com/resources/dashboard/platform-versions.html> 26. Oktober 2011
  12. Barrera D, Kayacik HG, van Oorschot PC, Somayaji A (2010) A methodology for empirical analysis of permission-based security models and its application to Android. In: Proceedings of the 17th ACM conference on Computer and communications security, ACM, New York, NY, USA, CCS '10, pp 73–84, DOI: <http://doi.acm.org/10.1145/1866307.1866317> 26. Oktober 2011
  13. Enck W, Oteau D, McDaniel P, Chaudhuri S (2011) A Study of Android Application Security. In: Proceedings of the 20th USENIX Security Symposium
  14. D Maslennikov (2011) Malware in the Android Market: here we go again. [http://www.securelist.com/en/blog/11267/Malware\\_in\\_the\\_Android\\_Market\\_here\\_we\\_go\\_again](http://www.securelist.com/en/blog/11267/Malware_in_the_Android_Market_here_we_go_again) 26. Oktober 2011
  15. Mahaffey K (2011) Security Alert: DroidDream Malware Found in Official Android Market. <http://blog.mylookout.com/2011/03/security-alert-malware-found-in-official-android-market-droiddream/> 26. Oktober 2011
  16. Droid Life: A Droid Community Blog (2011) Motorola Eases Up on Locked Bootloader Stance, Plans to Unlock Portfolio in 2011? <http://www.droid-life.com/2011/04/26/motorola-eases-up-on-locked-bootloader-stance-plans-to-unlock-portfolio-in-2011/> 26. Oktober 2011
  17. Chou P (2011) HTC to open bootloaders. <https://www.facebook.com/HTC/posts/10150307320018084> 26. Oktober 2011
  18. DroidDream (2011) <http://www.androidpolice.com/2011/03/01/the-mother-of-all-android-malware-has-arrived-stolen-apps-released-to-the-market-that-root-your-phone-steal-your-data-and-open-backdoor/> 12. April 2011
  19. Bishop M (2003) Computer Security: Art and Science. Addison-Wesley
  20. Loscocco P, Smalley S (2001) Integrating Flexible Support For Security Policies Into The Linux Operating System. In: Proceedings of the FREENIX Track of the 2001 USENIX Annual Technical Conference
  21. Elad B, Eli B, Nathan K (2007) Instant Ciphertext – Only Cryptanalysis of GSM Encrypted Communication. *J Crypt* 21(3):392–429

---

# Kapitel 16

## Zertifizierte Apps: mehr Funktionalität, Sicherheit und Bedienungsfreundlichkeit

Wolf-Rüdiger Heidemann und Ina Zumbruch

**Zusammenfassung** Die so genannten Applikationen/Applications, kurz Apps, für mobile Endgeräte fordern die Programmierer besonders heraus. Denn für optimale Funktionalität, Sicherheit und Bedienungsfreundlichkeit (Usability) bestehen komplexere Rahmenbedingungen als bei Desktop-PCs: geringere Bildschirmgrößen, schwankende Netzabdeckung und oft die Verarbeitung sensibler Daten. Weil der Stellenwert der nötigen Qualitätssicherung bei der App-Entwicklung häufig unterschätzt wird, kommt es immer wieder zu Mängeln bei der Funktionalität, Datensicherheit oder Ergonomie. Auch besteht Unsicherheit auf Seiten der Nutzer, inwiefern eine App den Qualitätsansprüchen gerecht wird. Das gilt insbesondere für das Beispiel Online-Banking, bei dem die App zur Schnittstelle zwischen dem Kunden und dem Kreditinstitut wird. Die Zertifizierung durch einen unabhängigen Dritten macht den Qualitätsstand und damit die Funktionalität, Sicherheit und die Bedienungsfreundlichkeit (Usability) einer App transparent. TÜV SÜD Product Service hat auf Basis der Dienstleistung „Geprüfte Softwarequalität“ einen eigenen Prüfkatalog entwickelt, der direkt auf mobile Applikationen zugeschnitten ist. Für mobiles Online-Banking bieten die Banken selbst Apps an, aber auch Softwareunternehmen wie die stoeger it GmbH. Das Beispiel der TÜV SÜD-Zertifizierung der sto-

---

Wolf-Rüdiger Heidemann  
TÜV SÜD Product Service GmbH  
Software-Qualität und Escrow-Services, München,  
E-mail: wolf.heidemann@tuev-sued.de

Ina Zumbruch  
TÜV SÜD Product Service GmbH  
Software-Qualität und Escrow-Services, München,  
E-mail: ina.zumbruch@tuev-sued.de

eger it-App iOutBank zeigt, wie die erfolgreiche Bewältigung von unabhängigen Prüfkriterien die Funktionalität und Sicherheit einer Online-Banking-Anwendung weiter verbessert. Der Prüfprozess hat hier neben der Funktionalität und Datensicherheit die zugehörigen Prozesse der Softwareentwicklung und Qualitätssicherung betrachtet. Der kombinierte Prüfansatz auf Produkt- und Prozessebene verhindert, dass lediglich eine Momentaufnahme zum Stand der Software entsteht. Dafür sind neben mehreren umfassenden Einzelprüfungen im Labor auch wiederkehrende Vor-Ort-Audits erforderlich.

## 16.1 Einleitung

Der Erfolg von Smartphones und Tablet-PCs inspiriert Softwareentwickler weltweit, Information, Unterhaltung und Business-Anwendungen mobil nutzbar zu machen. Täglich wächst die Zahl der online erhältlichen Anwenderprogramme – der so genannten Applikationen/Applications, kurz Apps. Mobile Endgeräte wie iPhone, Blackberry aber auch Android-, Symbian- oder Windows-basierte Handys oder die zahlreichen Touch-Pad-Varianten fordern die Programmierer im Vergleich zum Desktop-PC jedoch besonders heraus. Denn in puncto Funktionalität, Sicherheit und Bedienbarkeit (Usability) bestehen zusätzliche Anforderungen. Grund sind die wesentlich komplexeren Rahmenbedingungen bei den mobil eingesetzten Apps: geringere Bildschirmgrößen und Prozessorleistungen, schwankende Netzabdeckung, unterschiedliche Datenübertragungswege wie W-LAN oder Bluetooth sowie umgebungsbedingte Störquellen und weitaus mehr sensible und schützenswerte personenbezogene Daten (z. B. ortsabhängige Informationen).

Bei der Entwicklung von Apps werden diese neuen Anforderungen nicht immer ausreichend berücksichtigt. Der Stellenwert der nötigen Qualitätssicherung wird häufig unterschätzt. Die Folgen: Mängel beim Datenschutz und der Datensicherheit, instabile Programme oder wenig ergonomische Menüführungen. Das dürfte auch der Grund für den Misserfolg vieler Neuentwicklungen sein. Schon früh hat eine Studie gezeigt, dass nur ein Bruchteil der heruntergeladenen Apps tatsächlich dauerhaft genutzt wird.<sup>1</sup> Gerade einmal drei Prozent sind 30 Tage nach dem Download noch in Verwendung. Und mit negativen Bewertungen durch den Nutzer im Web oder Online-Store sinken auch die Downloadzahlen einer App.

Häufig besteht Unsicherheit auf Seiten von privaten und kommerziellen Nutzern, inwieweit eine App den hohen Qualitätsansprüchen tatsächlich gerecht wird. Das gilt insbesondere dann, wenn das Programm sensible Daten verarbeitet, austauscht oder kostenpflichtig ist. Die Zertifizierung durch einen unabhängigen Dritten macht den Qualitätsstand und damit die Funktionalität, Sicherheit und die Bedienungsfreundlichkeit (Usability) einer App transparent. Dies ist neben Business-

---

<sup>1</sup> Peter Müller 2009, URL: [http://www.macwelt.de/artikel/\\_News/364237/studie\\_iphone\\_apps\\_nur\\_selten\\_genutzt/1](http://www.macwelt.de/artikel/_News/364237/studie_iphone_apps_nur_selten_genutzt/1) [08.06.11]

Anwendungen auch für nicht kommerzielle Apps – mit Ausnahme von reinen Spiele-Applikationen – sinnvoll.

TÜV SÜD Product Service hat auf Basis der Dienstleistung „Geprüfte Softwarequalität“ einen eigenen Prüfkatalog entwickelt, der direkt auf mobile Applikationen zugeschnitten ist. Im Rahmen des Verfahrens wird eine App zunächst einem umfangreichen Prüfverfahren unterzogen, dessen Kriterienkataloge die relevanten gesetzlichen Vorgaben sowie internationale Normen und Standards umfasst. Geprüft werden die Aspekte Funktionalität, Datenschutz und Datensicherheit sowie die Bedienungsfreundlichkeit (Usability). Durch die Zertifizierung von Apps werden nicht nur die Qualitätsstandards transparent. Auf Basis des vorausgehenden Prüfprozesses können schon in der Entwicklungs- und Produktionsphase entscheidende Verbesserungen erzielt werden. Das senkt die Entwicklungskosten und minimiert mögliche Haftungsrisiken, die bspw. bei Sicherheitsmängeln auf den Unternehmer zukommen können.

Welche Haftungsrisiken im Einzelnen bestehen und wie hoch die Qualitätsanforderungen an eine App sind, hängt von ihrem Verwendungskontext ab. Datenschutz und Datensicherheit sind wichtige Aspekte. Dabei spielt auch die Art der verwendeten Daten eine entscheidende Rolle. Das gilt insbesondere für das Beispiel Online-Banking. Die App wird dort zu einer Schnittstelle zwischen dem Kunden und dem Kreditinstitut. Nicht autorisierte Zugriffe bspw. auf PIN-Nummern müssen sowohl bei der Datenübertragung, als auch beim Verlust des Endgeräts und bei dessen unbefugter Nutzung durch Dritte unterbunden werden.

Zugleich steigt die Nachfrage nach mobilem Online-Banking, und damit verändern sich auch die Anforderungen an den Zahlungsverkehr. Zeitersparnis, die hohe Flexibilität und geringe Kosten durch Datenflatrates sind die Treiber dieser Entwicklung. Die zunehmende Verbreitung des Onlinebankings ist zudem als Vorstufe eines generellen Wandels im Zahlungsverkehr zu sehen. Schon heute werden bspw. Bahntickets online gekauft, auf dem Smartphone zur Kontrolle bereithalten und online bezahlt. Mit der Transformation der Zahlungssysteme in Richtung mobiler Anwendungen steigt indes die Relevanz von aussagekräftigen Zertifikaten. Die Güte einer Zertifizierung steht und fällt mit den Prüfanforderungen und ihren Kriterienkatalogen. Wie diese sinnvoll ausgestaltet sind, zeigt das Beispiel der erfolgreich zertifizierten Online-Banking-App iOutBank, die sich auf den Geräten iPhone, iPad und iPod touch installieren lässt. Die stoeger it GmbH hat dafür vergangenes Jahr nicht nur das erste TÜV SÜD-Zertifikat für eine mobile Applikation erhalten; Apple hat iOutBank zudem als „App des Jahres 2010“ ausgezeichnet.

## 16.2 Umfangreicher Prüfprozess als Basis für die Zertifizierung

Der Zertifizierung von Apps geht ein umfangreicher Prüfprozess voraus. Die Kriterienkataloge von TÜV SÜD Product Service umfassen neben der Funktionalität die Datensicherheit und den Datenschutz sowie optional die Bedienungsfreundlichkeit (Usability). Des Weiteren werden die Prozesse in Entwicklung und Qualitätssiche-

rung betrachtet. Die Softwareexperten zertifizieren Apps für iPhone, iPod und iPad, aber auch für PDA und BlackBerry sowie Windows, Linux, Android oder Symbian Phone. Dabei orientieren sich die Prüfgrundlagen am Stand der Technik, an den gesetzlichen Vorschriften und an Vorgaben aus den relevanten anerkannten Standards, gehen aber inhaltlich noch über diese hinaus. Die Hauptprüfriterien sind abgeleitet aus Grundsatznormen zur Prüfung von Softwareprodukten und dem Qualitätsmanagement. Im Rahmen des Prüfprozesses ist überdies der Hersteller verpflichtet, seine App frühzeitig umfangreichen Tests zu unterziehen, die wiederum von TÜV SÜD evaluiert und verifiziert werden.

Der Prüfprozess gliedert sich in umfangreiche Audits, Vorort-Prüfungen der produktbezogenen Prozesse und deren Dokumentation. Zum einen wird dabei der „Ist-Stand“ des Softwareprodukts im TÜV SÜD-eigenen Prüflabor in München analysiert. Zum anderen werden vor Ort einzelne Prozess-Schritte der Softwareentwicklung und -pflege betrachtet. Diese betreffen die Softwareentwicklung und -einführung, genauso wie Wartung, Support und die Qualitätssicherung. Das schließt sowohl technische als auch organisatorische Maßnahmen beim Softwarehersteller ein. Wurden die grundlegenden Anforderungen zur Softwarequalität berücksichtigt, umgesetzt und dokumentiert? Wurden die Voraussetzungen geschaffen, um eine hohe funktionale Produktqualität auch langfristig sicherzustellen? Hier kommen Dokumente aus dem Entwicklungsprozess der Software in den Blick, wie Spezifikationen, Testpläne und Testprotokolle sowie weitere Unterlagen – einschließlich der Handbücher, Produktbeschreibungen und Benutzerdokumentation oder der Online-Hilfe. Zwar bestehen dabei für jedes Unternehmen grundsätzlich dieselben Prüfanforderungen. Doch werden die Gegebenheiten vor Ort dahingehend berücksichtigt, dass bspw. für Kleinstunternehmen angemessene Dokumentationspflichten festgelegt werden.

### **16.2.1 Funktionalität**

Funktionalitätsprüfungen basieren auf den Anforderungen der internationalen Norm ISO 25051 „Software-Engineering – Softwareproduktbewertung – Qualitätsanforderungen an seriennmäßig produzierte Softwareprodukte (COTS) und Prüfanweisungen“. Für das TÜV SÜD-Prüfzeichen „Softwarequalität – Funktionalität“ ist sicherzustellen, dass die Software grundlegende Anforderungen erfüllt und der Produktbeschreibung/Bedienungsanleitung entspricht.

Folgende Qualitätsmerkmale werden betrachtet:

- Sind die Funktionen in der App vollständig, korrekt und konsistent?
- Ist auch die Dokumentation vollständig, korrekt und konsistent?
- Ist die App zuverlässig und robust?
- Existieren innerhalb des Unternehmens geeignete Strukturen und verantwortliche Mitarbeiter, um die Funktionalität der App zu sichern und die TÜV SÜD-Prüfkriterien zu erfüllen?

- Liegen zum Prüftermin alle relevanten Dokumente wie Prozessbeschreibungen, Fehlerreports, Pflichten- und Lastenhefte vor?
- Kann das erreichte Qualitätsniveau bei der Funktionalität dauerhaft gesichert werden?

Diese allgemeinen Beispielkriterien zur Funktionalitätsprüfung werden auf einer zweiten Detaillierungsstufe produktspezifisch jeweils auf die zu zertifizierende App angepasst.

### **16.2.2 Datenschutz und Datensicherheit**

Datenschutz und Datensicherheit sind Teilthemen mit jeweils unterschiedlichen Anforderungen und Prüfansätzen. Datenschutz beinhaltet den Schutz personenbezogener Daten. Dessen rechtliche Anforderungen fixiert u. a. das Bundesdatenschutzgesetz (BDSG), das Telemediengesetz (TMG) oder das Telekommunikationsgesetz (TKG). Hingegen adressiert die Datensicherheit technische und organisatorische Maßnahmen zur IT-Security. Als grundlegend gelten dort die Empfehlungen des Bundesamts für Sicherheit in der Informationstechnik (BSI) zum IT-Grundschutz. Weitere Qualitätskriterien für Internet-Angebote stammen von D21, einer Initiative von Politik und Wirtschaft.

Für die Vergabe des TÜV SÜD-Prüfzeichens „Softwarequalität – Funktionalität – Datensicherheit“ muss eine App den Anforderungskatalog der Funktionalitätsprüfung erfüllen. Im Rahmen des Audits wird ermittelt, welche Anforderungen und Gesetze im Einzelnen für das Unternehmen und die App gelten. Sowohl vor Ort als auch im Zuge von Tests wird überprüft, ob die Anforderungen an Datenschutz und Datensicherheit eingehalten werden. Das Unternehmen selbst muss ggf. – abhängig von der Unternehmensgröße und der Art der verarbeiteten Daten – einen eigenen Datenschutzbeauftragten bestellen, der mit den sich wandelnden gesetzlichen Vorschriften und Praxisrisiken vertraut ist und sich kontinuierlich fortbildet. TÜV SÜD Product Service verfügt mit dem Prüfprogramm „Datenschutz/Datensicherheit in IT-Systemen“ über einen eigenen Standard mit besonderen Qualitätskriterien.

Die Tabelle 16.1 zeigt für die IT-Sicherheit relevante Beispielkriterien, die im Rahmen des Audits geprüft, dokumentiert und protokolliert werden.

Die Praxis zeigt immer wieder, dass der Faktor Mensch bei der Datensicherheit besondere Aufmerksamkeit verdient. Selbst ein sonst sicheres System kann durch fehlerhafte, unachtsame oder nicht bestimmungsgemäße Nutzung Risiken für die Datensicherheit bergen. Beispielsweise sind Passwortrichtlinien und -analysen obligatorisch, welche triviale oder unsichere Passwörter erkennen, die vom Programm auch nicht akzeptiert werden sollten. Zudem ist bei einem Verlust des Endgeräts sicherzustellen, dass ein Timeout eine laufende Anwendung beendet. Wenn sensible Daten verarbeitet werden, dürfen diese erst nach einer erneuten Passworteingabe freigegeben werden. Bei einer wiederholten fehlerhaften Passworteingabe, die auf einen nicht autorisierten Nutzer schließen lässt, kann es sinnvoll sein, dass ein automatischer Löschvorgang initiiert wird, um sensible Daten unzugänglich zu machen.

**Tabelle 16.1** Beispielkriterien

Stichwort	Inhalt
Personenangaben	Welche Personenangaben (Name/Telefon/E-Mail) z. B. von Administratoren sind von außen zu ermitteln?
Sichtbare IT-Komponenten des Auftraggebers	Welche IT-Komponenten des Auftraggebers sind aus dem öffentlichen Netz erreichbar?
Schnittstellen zum internen Netz	Welche Schnittstellen zwischen dem Internet und dem firmeninternen Intranet des Auftraggebers sind von außen erkennbar?
Externe Dienste	Sind aktive externe Dienste von außen zugänglich?
Systemschwachstellen	Welche bekannten Schwachstellen bei Hard- und Software sind auf den erkannten Komponenten sichtbar?
Exploits	Ausführbarkeit von Exploits (Schadprogramme oder Befehlsfolgen, die Sicherheitslücken und Fehlfunktionen in Anwendungsprogrammen ausnutzen).
Mail-Attacken	<i>Für Mailserver (bspw. im Rahmen des Supports für eine App):</i> Missbrauchstest zu Spam- und Massenmails, Relaying (missbräuchliches Weiterleiten von E-Mails), Spoofing (Täuschungsversuche in der Informationstechnik).
Sniffing	<i>Bei Datenübertragung als Teil des Systems:</i> Test, ob unbefugtes Mitlesen von Daten möglich ist (einschließlich einer Decodierung codierter Daten).

### 16.2.3 Bedienungsfreundlichkeit bzw. Usability/Ergonomie

Neben der grundlegenden Zertifizierung von Funktionalität und Datenschutz/Datensicherheit kann optional die Bedienungsfreundlichkeit geprüft werden. Die Prüfung für das Zertifikat „Softwarequalität – Funktionalität – Ergonomie“ erfolgt auf Grundlage folgender Normen und Kriterien:

- DIN EN ISO 9241 „Ergonomie der Mensch-System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung“. Bei mobilen Apps mit Touch Screen ist zusätzlich der Teil 16 (Dialogführung mittels direkter Manipulation) relevant.
- Des Weiteren gelten Anforderungen/Empfehlungen hinsichtlich der mobilen Usability, da mobile Geräte im Vergleich zu PCs oder Notebooks häufiger in einem relativ störungsreichen und daher ablenkenden Umfeld genutzt werden. Zudem suchen Nutzer im mobilen Web häufig noch gezielter nach bestimmten bspw. ortsbezogenen Informationen.
- Usability-relevante Bereiche der Norm ISO 25051 „Software-Engineering – Softwareproduktbewertung – Qualitätsanforderungen an kommerzielle serienmäßig produzierte Softwareprodukte (COTS) und Prüfanweisungen“.

Die App wird im jeweiligen Nutzungskontext betrachtet. Usability-Prüfungen erfolgen dann als Expertenreview, teilnehmende Beobachtung oder Benutzerbefragung.

Bspw. werden folgende Qualitätsaspekte betrachtet:

- Aufgabenangemessenheit: Werden tatsächlich nur relevante Informationen angezeigt und zur Eingabe zugelassen? Ist die Ausfüllreihenfolge sinnvoll?
- Selbstbeschreibungsfähigkeit: Gibt das Programm verständliche Rückmeldungen in einer dem Kontext angemessenen Sprache? Wird über die Folgen einer Eingabe sinnvoll informiert?
- Lernförderlichkeit: Existiert ein Lernmodus im Programm? Bestehen ausreichende Programmhilfen zur Menüführung und den verwendeten Symbolen?
- Steuerbarkeit: Kann die Programmnutzung unterbrochen und ohne Verlust der Arbeitsergebnisse wieder aufgenommen werden? Können die Arbeitsschritte rückgängig gemacht werden?
- Erwartungskonformität: Werden die verwendeten Programm-Elemente, Anzeigen und Bezeichnungen im Verlauf konsistent verwendet? Sind die Größenverhältnisse sinnvoll und werden Informationen richtig priorisiert?

Weitere Punkte betreffen die Fehlertoleranz, Individualisierbarkeit, Effektivität, Effizienz und die Zufriedenstellung der Benutzer sowie Besonderheiten der mobilen Nutzung.

#### ***16.2.4 Qualitätssicherung bei den Prozessen***

Kernthema der Audits vor Ort sind die Qualitätssicherung und das Freigabeverfahren in Bezug auf die Softwareversionen, die vom Hersteller an den Kunden ausgeliefert werden sowie die damit verbundenen produktbezogenen Prozesse wie Wartung, Support und das Produktmanagement. Entscheidend dabei sind Feinheiten, die wesentlich zur dauerhaften Sicherung einer hohen Prozessqualität beitragen und sich nur vor Ort qualifiziert analysieren lassen: Wie sind interne Abläufe und Strukturen gestaltet? Gibt es automatisierte Tests und Regressionstests und wer überprüft diese? Existiert ein Qualitätsverantwortlicher, der eine Gesamtperspektive auf den erzielten Qualitätsstand und mögliche Veränderungen hat? Der Themenbereich der Qualitätssicherung ist Bestandteil der Norm ISO 25051 und lehnt sich zusätzlich an grundlegende Anforderungen zum Qualitätsmanagement wie die Norm ISO 9001 an, schließt aber auch die anerkannte Praxis aus dem jeweiligen Anwendungsbereich ein.

#### ***16.2.5 Anpassungsfähigkeit an die technische Entwicklung***

Prüfpläne und -kriterien lassen sich in der Regel schneller an neue technische Neuerungen und deren Implikationen anpassen als Normen, Gesetze und Regelwerke – ein wesentlicher Vorteil, den es für mehr Sicherheit, Funktionalität und Bedienkomfort von Apps zu nutzen gilt. Aufgrund schneller Innovationszyklen bei der mobilen

Netzinfrastruktur (von GSM über UMTS hin zu LTE), insbesondere aber bei der Hardware und vor allem bei der Software im IT-Sektor, beinhalten aussagekräftige Zertifizierungen daher wiederkehrende Prüfungen. Das TÜV SÜD-Zertifikat gilt für ein Jahr, sofern unterjährig keine wesentlichen Änderungen an der App oder dem Betriebssystem erfolgen. Tritt dieser Fall ein, wird eine Revisionsprüfung vor Ablauf der Jahresfrist obligatorisch. Dies gilt insofern als bestimmender Faktor, weil durch Updates sowohl der Software als auch des Betriebssystems bspw. Sicherheitslücken entstehen können, die bei einer für das Vorgängersystem zertifizierten App-Version nicht aufgetreten sind. Gerade in einem neuen Wachstumsmarkt, in dem noch wenige Erfahrungswerte bestehen, ist es entscheidend, dass bestehende Zertifizierungen über einen gewissen Grad an Flexibilität verfügen. Neu identifizierte Problemstellungen und daraus abgeleitete Erkenntnisse müssen schnellstmöglich in einer Neuprüfung berücksichtigt werden können.

## 16.3 Fallbeispiel Online-Banking: die App „iOutBank“

Aufgrund der 24-stündigen Verfügbarkeit von Online-Banking wächst die Zahl der Teilnehmer stetig. Über 30 Millionen Online-Konten bei deutschen Banken werden genutzt für Überweisungen, Daueraufträge, Kontoinformationen oder für Wertpapiertransaktionen. Zunehmend nutzen die Kontoinhaber nicht nur Desktop-PCs für das Online-Banking, sondern auch Smartphones und Tablets. Die dazu nötigen Apps bieten die Banken selbst an und Softwareunternehmen wie die stoeger it GmbH aus Dachau bei München. Das Beispiel der TÜV SÜD-Zertifizierung der Online-Banking-Anwendung iOutBank von stoeger it zeigt, wie die erfolgreiche Bewältigung von unabhängigen Prüfkriterien die Sicherheit und Funktionalität einer App weiter verbessert.

Mit iOutBank können Kunden mittels einer App mehrere Konten sicher und schnell verwalten, selbst wenn sie diese bei unterschiedlichen Banken führen. Umsatzdaten, Überweisungen und Lastschriften lassen sich vom iPhone, iPod oder iPad mobil versenden. Die App iOutBank ist das Kernprodukt von stoeger it und wurde von Apple als „App des Jahres 2010“ ausgezeichnet. Seit 2003 entwickelt das Unternehmen mobile Banking-Software für Symbian-Smartphones und Apple-Produkte. Der Gründer und Geschäftsführer des Unternehmens verfügt über jahrelange Expertise in der Finanzwirtschaft und Unternehmensberatung und leitet ein Entwicklungsteam aus hoch qualifizierten IT-Spezialisten.

Bei der Entwicklung von iOutBank war die Sicherheit der Anwendung oberstes Ziel. Das Unternehmen stoeger it wollte sich die erzielte hohe Qualität von einer unabhängigen und renommierten Institution bestätigen lassen und damit den sichtbaren Nachweis führen, dass auch ein Mittelständler sichere und zuverlässige Software anbieten kann. Vor diesem Hintergrund hat das Unternehmen TÜV SÜD Product Service beauftragt, seine App auf Funktionalität und Datensicherheit zu prüfen. Mit dem TÜV SÜD-Zertifikat will das Unternehmen zudem seine langfristige und nachhaltige Arbeit im Bereich Online-Banking dokumentieren.

### **16.3.1 Kombinierter Produkt- und Prozessansatz bei der Prüfung**

Für den Prüfprozess bei iOutBank standen mit der Funktionalität und Datensicherheit des Softwareprodukts die zugehörigen Prozesse im Fokus. Diese umfassen die Softwareentwicklung, die Qualitätssicherung, die Freigabeverfahren sowie Wartung und Support. Da diese Prozesse ineinander greifen, müssen die bestehenden Wechselwirkungen mit analysiert werden. Der kombinierte Prüfansatz auf Produkt- und Prozessebene verhindert, dass eine Momentaufnahme entsteht, die lediglich den aktuellen Stand der Software abbildet. Dafür sind neben mehreren umfassenden Einzelprüfungen im Labor Vor-Ort-Audits erforderlich. Zentrale Elemente waren unterschiedliche Softwaretests hinsichtlich Funktionalität und Datensicherheit sowie Audits zu den Aspekten Prozessqualität und Datenschutz/Datensicherheit. In allen Bereichen wurden zudem detaillierte Dokumentenprüfungen vorgenommen.

### **16.3.2 Beispiel Datensicherheit**

Das Unternehmen stoeger it verarbeitet keine personenbezogenen Daten, welche über die App iOutBank eingegeben werden. Daher gilt das Unternehmen nur als verantwortliche Stelle im Sinne des BDSG für Daten, die bei Anfragen über den Kundenservice erfasst werden. Die Anforderungen zum Datenschutz beziehen sich somit auf die technischen und organisatorischen Maßnahmen zur Erfüllung der gesetzlichen Anforderungen §9 BDSG im Rahmen der Kundenbetreuung. Für diese gesetzlichen Anforderungen sowie für die weitergehenden Anforderungen zur Datensicherheit wurde ein detailliertes Prüfkonzept mit zugehörigem Prüfplan erarbeitet und der gesamte Prüfprozess dokumentiert.

Zum Aspekt der Datensicherheit bspw. wurden Angriffsmöglichkeiten und Risiken analysiert und beschrieben. Dabei kam insbesondere der Verlust der Vertraulichkeit in den Blick, wie er durch Lesen von Datenübertragungen durch Dritte, den unbefugten Zugang zur App oder unbefugte Zugriffe auf Speicherinhalte entstehen kann sowie bei Verlust der Datenintegrität durch unbefugte Veränderungen übermittelter Daten. Diese können im vorliegenden Fall auf zwei Wegen übertragen werden, die beide potenzielle Angriffsziele sind. Das ist einerseits der Datenaustausch mit der Bank (per HBCI,<sup>2</sup> über Mobilfunknetz oder WLAN) und andererseits die Datensicherung auf dem Computer (über WLAN).

Die Anforderungen sind spezifisch und aktuell nach dem Stand der Technik zu beschreiben. Entscheidend hierfür ist u. a. die Beantwortung folgender Fragen: Sind nach BDSG die technischen Voraussetzungen vorhanden für die Zugangskontrolle

---

<sup>2</sup> „HBCI“ steht für „Homebanking Computer Interface“ und beschreibt einen Standard, der eine verschlüsselte Datenübertragung erlaubt. Dieser wird von allen Bankengruppen, die der Zentrale Kreditausschuss (ZKA) repräsentiert, unterstützt. HBCI ist 2002 im „Financial Transaction Service“ (FinTS) aufgegangen, mit zusätzlichem Sicherheitsgewinn und Möglichkeiten beim PIN/TAN-Verfahren.

(App), Zugriffskontrolle (Daten im Speicher des iPhones bzw. iPads) und die Weitergabekontrolle (WLAN oder Mobilfunknetz)? Wie steht es um die Empfehlungen zum IT-Grundschutz? Werden die gespeicherten und übertragenen Daten verschlüsselt und zuverlässig authentisiert bzw. autorisiert? Führen mehrere Fehl-Logins dazu, dass die Daten gesperrt oder gelöscht werden? Lassen sich Sicherheitslücken durch unerwartet geschlossene Programm- oder Systemprozesse bzw. durch undefinierte Systemzustände verhindern?

Alle im Rahmen des Prüfprozesses vorgeschriebenen Dokumente haben die TÜV SÜD-Experten auf definierte Sicherheitsanforderungen hin eingehend analysiert und zudem die Entwickler vor Ort interviewt. Bezogen auf die Funktionsweise der App wurde überprüft: der sichere Login, unerwartet geschlossene Programm- oder Systemprozesse und undefinierte Systemzustände sowie Fehlfunktionen, die unbefugte Zugriffe erlauben. Bei den gespeicherten Daten wurde der Normalfall verglichen mit dem Zustand nach Fehl-Logins oder unerwartet geschlossenen Systemprozessen. Festgestellte Mängel wurden im Prüfungszeitraum behoben und waren Gegenstand erneuter Kontrollprüfungen.

### ***16.3.3 Lösungen und Ergebnisse***

Die Prüfung der TÜV SÜD-Experten hat gezeigt, dass der Zugang zu iOutBank ausreichend gesichert ist. Der Nutzer erhält Hinweise für ein sicheres Passwort, mehrfache Fehl-Logins führen zum Löschen aller Daten. Auch wurden keine Fehlfunktionen des Programms festgestellt, über die ein unbefugter Zugang möglich wäre. Alle Funktionsaufrufe oder Eingaben von Nutzern, die ein erhöhtes Risiko nach sich ziehen, werden mit Warnungen und deren Begründung quittiert. Alle relevanten Daten werden verschlüsselt gespeichert und sind durch Unbefugte mit einem realistischen Aufwand nicht aus dem Speicher auszulesen. Dies ist selbst nach einer möglichen unerwartet geschlossenen Anwendung gewährleistet. Die Datenübertragung zwischen App und Bank erfolgt verschlüsselt und ist auch dort durch Unbefugte nicht mit einem realistischen Aufwand lesbar.

Weil die Datenübertragung zwischen der App und dem Computer (z. B. zu Backup-Zwecken) via WLAN nicht verschlüsselt erfolgt, ist diese allerdings lesbar. Beim Aufruf dieser Funktion wird der Nutzer explizit gewarnt und aufgefordert, eine solche Datenübertragung nur über ein eigenes und sicheres WLAN vorzunehmen.

Bei der abschließenden Prüfung wurde die Softwarequalität auditiert – sowohl bezüglich der Softwareanwendung iOutBank als auch des damit verbunden Systems der Qualitätssicherung und Prozesse. Alle aufgefunden Verbesserungsmöglichkeiten wurden im Prüfverlauf umgesetzt und durch Folgetests bestätigt. Die App iOutBank hat die anerkannten Anforderungen zur Softwarequalität bezüglich Funktionalität, Datenschutz/Datensicherheit und Qualitätsmanagement erfüllt. TÜV SÜD Produkt Service konnte stoeger it daher erfolgreich das Prüfzeichen „Softwarequalität – Funktionalität und Datensicherheit“ für deren App iOutBank vergeben. Die

Qualitätsmaßstäbe gemäß Prüfspezifikation werden überdies für die Weiterentwicklung der Software und die damit verbundenen Prozesse zu Grunde gelegt. Das prüft TÜV SÜD Product Service im Zuge von jährlichen Revisionsprüfungen. Zudem sind bei wesentlichen Änderungen am App oder dem Betriebssystem vorgezogene Nachprüfungen erforderlich.

## 16.4 Fazit

Die Zertifizierung von Apps durch unabhängige Prüforganisationen fördert die Entwicklung einheitlicher Standards und ist damit ein wesentlicher Schritt hin zu mehr Qualität und Sicherheit bei mobilen Anwendungsprogrammen im Segment Smartphones und Tablets. Zertifikate, die auf soliden, aber sich kontinuierlich an die technischen Entwicklungen anpassenden Prüfgrundlagen basieren, geben Softwarekunden eine hilfreiche Orientierung in einem schnell wachsenden Markt. Softwareentwickler profitieren von verbesserten Prozessen und Strukturen. Das Prüfzeichen setzt zudem einen Akzent in Sachen Verlässlichkeit und dokumentiert die erzielten hohen Qualitätsstandards nach außen.

## Literaturverzeichnis

1. Apple Inc. iOutBank – App des Jahres 2010, elektronisch veröffentlicht: URL: <http://itunes.apple.com/de/app/ioutbank-app-des-jahres-2010/id299609615?mt=8> [08.06.11]
2. Bundesamt für Sicherheit in der Informationstechnik (BSI), IT-Grundschutz, URL: [https://www.bsi.bund.de/DE/Themen/ITGrundschutz/itgrundschutz\\_node.html](https://www.bsi.bund.de/DE/Themen/ITGrundschutz/itgrundschutz_node.html) [Stand Februar 2009]
3. Bundesministerium für Wirtschaft und Technologie (BMWi), 20.12.1990, Bundesdatenschutzgesetz (BDSG), URL: [http://www.gesetze-im-internet.de/bundesrecht/bdsg\\_1990/gesamt.pdf](http://www.gesetze-im-internet.de/bundesrecht/bdsg_1990/gesamt.pdf) [Stand August 2009]
4. Bundesministerium für Wirtschaft und Technologie (BMWi), 22.06.2004, Telekommunikationsgesetz (TKG), URL: [http://www.gesetze-im-internet.de/bundesrecht/tkg\\_2004/gesamt.pdf](http://www.gesetze-im-internet.de/bundesrecht/tkg_2004/gesamt.pdf) [Stand März 2011]
5. Bundesministerium für Wirtschaft und Technologie (BMWi), 26.02.2007, Telemediengesetz (TMG), URL: <http://www.gesetze-im-internet.de/bundesrecht/tmg/gesamt.pdf> [Stand Mai 2010]
6. Initiative D21, D21-Qualitätskriterien für Internet-Angebote, URL: [http://www.foehlisch.de/guetesiegel/docs/D21\\_Qualitaetskriterien\\_2011.pdf](http://www.foehlisch.de/guetesiegel/docs/D21_Qualitaetskriterien_2011.pdf) [Stand Oktober 2010]
7. International Organization of Standardization, DIN EN ISO 9001, Stand Dezember 2008, „Qualitätsmanagementsysteme – Anforderungen“
8. International Organization of Standardization, DIN EN ISO 9241, Stand Januar 2011, „Ergonomie der Mensch-System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung“
9. International Organization of Standardization, DIN EN ISO 9241, Stand Januar 2011, „Ergonomie der Mensch-System-Interaktion – Teil 16: Dialogführung mittels direkter Manipulation“
10. International Organization of Standardization, ISO/IEC 25051, Stand April 2006, „Software-Engineering – Softwareproduktbewertung – Qualitätsanforderungen an kommerzielle serienmäßig produzierte Softwareprodukte (COTS) und Prüfanweisungen“

11. Müller, P. (2009): Studie: iPhone Apps nur selten genutzt, elektronisch veröffentlicht: URL: [http://www.macwelt.de/artikel/\\_News/364237/studie\\_iphone\\_apps\\_nur\\_selten\\_genutzt/1](http://www.macwelt.de/artikel/_News/364237/studie_iphone_apps_nur_selten_genutzt/1) [08.06.11]
12. Zentraler Kreditausschuss: FinTS, elektronisch veröffentlicht: URL: <http://www.zka-online.de/zka/zahlungsverkehr/electronic-banking.html>, [www.FinTS.org](http://www.FinTS.org) [08.06.11]

---

# Kapitel 17

## Mobile Apps in Enterprise-Anwendungen unter Berücksichtigung von Sicherheitsaspekten

Jens Bertram und Carsten Kleiner

**Zusammenfassung** In diesem Kapitel wird die Eignung aktueller Betriebssystem-Plattformen mobiler Endgeräte für den Einsatz im Bereich der Unternehmensanwendungen untersucht. Im Unterschied zu Verbraucheranwendungen zeichnen sich die Apps im Enterprise-Bereich zumeist durch die Nutzung wesentlicher komplexerer und semantisch anspruchsvoller Schnittstellen aus. Außerdem spielen im Unternehmensumfeld nicht-funktionale Anforderungen wie z.B. Sicherheitsfunktionen eine deutlich größere Rolle. Gemeinsam mit der erforderlichen Integration in bestehende Geschäftsprozesse führt dies dazu, dass Dienste in diesem Bereich oft mithilfe von SOAP Web Services implementiert sind. Dazu kommen zumeist auch einschlägige weitere Web-Spezifikationen für zusätzliche Anforderungen zum Einsatz.

Um Apps im Unternehmensbereich einsetzen zu können, ist es erforderlich, dass alle diese Spezifikationen von der Client-Plattform unterstützt werden, um einerseits den Entwicklungsaufwand für die App begrenzen zu können und andererseits von den Vorteilen der Verwendung von SOAP Web Services wie Flexibilität, Plattform-unabhängigkeit sowie Änderungsfreundlichkeit profitieren zu können. In diesem Kapitel werden wir aufzeigen, dass die aktuellen Plattformen eher an Verbraucheranforderungen orientiert sind und die Unterstützung für SOAP Web Services sowie Sicherheitsfunktionen noch nicht sehr ausgereift ist. Wir werden ferner zeigen, wie die Plattformen erweitert werden können, um die Integration in Geschäftsprozesse erreichen zu können. Allerdings bleibt durch die Nutzung von Erweiterungen das Risiko, dass diese Erweiterungen auch auf Neu- oder Weiterentwicklungen der Plattformen zeitnah zur Verfügung stehen müssen.

---

Jens Bertram  
Fachhochschule Hannover, Fakultät IV, Abt. Informatik,  
E-mail: jens.bertram1@stud.fh-hannover.de

Carsten Kleiner  
Fachhochschule Hannover, Fakultät IV, Abt. Informatik,  
E-mail: ckleiner@acm.org

## 17.1 Einleitung

Zurzeit werden der Absatz und damit auch die Funktionalität mobiler Endgeräte vornehmlich durch den Markt für Endbenutzer geprägt. Dies hat zur Entwicklung und Verbreitung neuartiger Bedienungsmodelle für diese Geräte geführt. Auch das Installieren von Anwendungen auf den Geräten (Apps) hat sich, u. a. durch die Bereitstellung entsprechender Online-Shops der Gerätehersteller, in der Masse durchgesetzt.

Allerdings haben die meisten dieser Anwendungen ebenfalls eher den Endkunden als Nutzer im Blick. Ein breiter Einsatz mobiler Geräte basierend auf diesen Plattformen im Enterprise-Bereich ist bisher noch nicht zu erkennen. Gerade der Einsatz mobiler Geräte im Unternehmensumfeld sollte jedoch zu wesentlichen Prozessverbesserungen führen können. Diese gehen wiederum mit Effizienzsteigerung und damit insgesamt verbesserter Unternehmensperformanz einher. Außerdem ermöglicht die Nutzung mobiler Endgeräte völlig neue Geschäftsmodelle oder bewirkt eine Verbesserung bestehender. Allerdings müssen mobile Geräte dazu in die bereits bestehenden Geschäftsprozesse der Unternehmen integriert werden. Die Schaffung komplett neuer unternehmensweiter IT-Systeme und zugehöriger IT-Infrastruktur unter Beteiligung von Mobilgeräten ist meist nicht zielführend.

In diesem Kapitel werden wichtige aktuelle Plattformen für mobile Endgeräte (wie z. B. Android, SymbianOS, iOS, Windows Mobile/Phone) auf ihre Unterstützung der spezifischen Anforderungen von Enterprise-Anwendungen hin untersucht. Dabei werden verschiedene funktionale, organisatorische und wirtschaftliche Aspekte für die Bewertung verwendet. Die Informationen zu den einzelnen Plattformen beziehen sich auf die jeweils aktuelle Version zum Zeitpunkt des Schreibens und unterliegen damit naturgemäß einem schnellen Wandel. Trotzdem werden die Ergebnisse zeigen, welche Aspekte bei den einzelnen Plattformen zur zufriedenstellenden Unterstützung von Enterprise Apps noch fehlen; diese Erkenntnisse werden noch einige Zeit gültig bleiben, da sie teilweise auf fundamentalen Problemen beruhen.

## 17.2 Spezifische Anforderungen von Enterprise Apps

Die IT-Anwendungslandschaft wird in vielen Unternehmen heutzutage in Anlehnung an das Paradigma der Service-orientierten Architektur (SOA, vgl. [1, 2]) organisiert. Ein wesentlicher Vorteil einer SOA ist dabei einerseits, dass die über eine wohl definierte Schnittstelle deklarierten und angebotenen Dienste flexibel und plattformunabhängig genutzt werden können. Komplexe Geschäftsprozesse werden durch Kombination zahlreicher, einfacherer Dienste definiert. Dienste werden häufig sogar in Landschaften, in denen keine Web Services genutzt werden, durch die einschlägigen Spezifikationen aus der Web-Service-Welt beschrieben, z. B. durch die Web Service Description Language (WSDL, [3]).

**Tabelle 17.1** Vergleich der Kommunikationsprotokolle SOAP und REST im Kontext von Apps (Skala: 5 – sehr gut, 1 – sehr schlecht)

Aspekt	SOAP	REST
Größe der Nachrichten	3	5
Client-Anforderungen	2	5
Aufwand Client-Implementierung	4 (mit Generierung)	4
Flexibilität der Dienste-Nutzung	5	3
Vielfalt der Sicherheitsfunktionalitäten	5	2
Weitere nicht-funktionale Anforderungen (QoS, etc.)	5	1

Sollen nun in bereits bestehenden Geschäftsprozessen auf mobilen Endgeräten einige der Aktionen ausgeführt werden, so ist es wichtig, dass sich diese Anwendungen nahtlos in die existierende IT-Landschaft einfügen. Dies bedeutet in SOAs, dass eine Unterstützung von Web Services benötigt wird. Beispiele für den Einsatz von Web Services in Unternehmen finden sich z. B. in [4, 5].

In der Praxis werden Web Services zumeist mit einem der beiden Kommunikationsprotokolle SOAP oder REST angesprochen. Eine kurze Gegenüberstellung dieser Protokolle findet sich in Tabelle 17.1, eine ausführliche Diskussion z. B. in [6]. Im Unternehmensumfeld zeichnen sich die Dienste zumeist durch eher komplexe, semantisch reichhaltige Schnittstellen aus. Der wesentliche Nachteil von SOAP, größere Nachrichten müssen ausgetauscht werden, spielt in der heutigen Zeit kaum noch eine Rolle, da inzwischen alle aktuellen Endgeräte sogar die Bandbreitenanforderungen des Video-Streaming erfüllen. Oft sind im Unternehmensbereich die Interaktionsmuster komplexer als sie mit REST im Endanwenderbereich zum Einsatz kommen (Senden einer Key-Value-Liste, Empfangen einer Antwort). Daher wird hier häufig SOAP als Kommunikationsprotokoll bevorzugt. Um also leicht in bestehende Unternehmensanwendungen integriert werden zu können, müssen die mobilen Plattformen die SOAP-Technologie nativ oder mit einfachen Erweiterungen unterstützen können.

Ein weiterer wesentlicher Unterschied der Enterprise-Anwendungen im Vergleich zu Endanwender-Apps sind die erhöhten Anforderungen im nicht-funktionalen Bereich. So ist es häufig erforderlich, eine bestimmte Dienstgüte garantieren zu können, Kommunikation nachweisbar aufzeichnen zu können oder auch Anwendungen mit möglichst geringem Aufwand an geänderte Dienste anpassen zu können. Auch für die erhöhten und spezifischeren Sicherheitsanforderungen im Unternehmensumfeld ist SOAP den REST-basierten Web Services überlegen, dies wird in einem späteren Abschnitt noch detaillierter diskutiert. Für die meisten dieser nicht-funktionalen Anforderungen gibt es für SOAP Web Services zahlreiche Spezifikationen des W3C, die standardisierte Umsetzungen definieren. Durch Nutzung dieser Spezifikationen wird so eine breite Einsetzbarkeit erreicht. Mobile Geräte für den Enterprise-Einsatz sollten also auch hier nativ oder mit wenig komplexen Erweiterungen möglichst viele der Spezifikationen erfüllen.

Wenn wir in diesem und dem folgenden Abschnitt von einer SOAP-Unterstützung sprechen, ist im Gegensatz zu vielen Werbebroschüren von Anbietern eine um-

fassende Unterstützung gemeint, die auch eine Nutzung der Vorteile von SOAP (Plattformunabhängigkeit, Flexibilität, Potenzial einer automatisierten Integration) ermöglicht. Aus unserer Sicht sind daher die folgenden Mindestanforderungen zu stellen:

1. XML-Verarbeitung
2. Automatische Generierung von Datentypen für SOAP-Nachrichten in der Entwicklungssprache der Client-Plattform
3. Generierung von Client-Stubs zu beliebigen WSDL-Beschreibungen, die die Web-Service-Aufrufe kapseln.

## 17.3 SOAP auf aktuellen Plattformen

Um die Vorteile von SOAP-Web Services nutzen zu können, sind zahlreiche Anforderungen an die Client-seitige Unterstützung zu stellen. So muss eine Plattform zum einen die Verarbeitung von XML-Dokumenten mit einem XML-Parser unterstützen. Oft wird (vgl. [7]) schon von einer SOAP-Unterstützung gesprochen, wenn XML-Dokumente auf einer Plattform in eine Zeichenkettendarstellung umgewandelt werden können und diese mit den entsprechenden String-Operationen bearbeitet werden kann. Auf diese Weise wird jedoch der Vorteil einer semantischen Beschreibung der ausgetauschten Nachrichten in einem allgemeinen Format nicht genutzt, sondern es wird nur der Nachteil des Overheads für diese Art der Beschreibung in Kauf genommen. Für die semantisch komplexen Schnittstellen in Enterprise-Anwendungen ist diese Art der XML-Verarbeitung ungeeignet. Des Weiteren wird die Flexibilität und automatisierte Integration mit solch einer Realisierung nie genutzt werden können, da bei jeder (und sei es nur syntaktischen aus Betriebsgründen) Änderung der Schnittstelle auf Server-Seite eine manuelle Änderung des Client-Code erforderlich ist.

Die Plattform sollte ferner eine Möglichkeit der Generierung von Datentypen und Client-Stubs zu einer gegebenen WSDL-Beschreibung eines Diensts ermöglichen. Dies erleichtert zum einen signifikant den Entwicklungsaufwand auf Client-Seite, da nur der eigentliche Code der mobilen Anwendung selbst zu implementieren ist. Zum anderen ermöglicht dies eine weitgehend automatische Anpassung der Client-Anwendung bei kleineren semantischen Änderungen am Service. Änderungen des Web Servers oder syntaktische Änderungen am Service können ebenfalls automatisch berücksichtigt werden. Schließlich ermöglicht diese Generierung später eine zumindest in der SOA-Vision angedachte automatisierte Integration von Diensten in Prozesse.

Bei der Generierung ist ferner zwischen einer Unterstützung beliebiger in XML-Schema definierbarer Datentypen und einer fest definierten Teilmenge davon zu unterscheiden. Erstere ist natürlich wünschenswert, jedoch auch wenig verbreitet auf mobilen Geräten. Nur so ist eine Nutzung zusätzlicher in SOAP integrierbarer Spezifikationen (z. B. für Sicherheitsfunktionen wie unten, semantischer Beschreibungen, vgl. [8], oder domänenspezifischer Standards, vgl. [9]) möglich.

**Tabelle 17.2** Überblick über SOAP-Unterstützung auf aktuellen mobilen Plattformen

Plattform	API	SOAP nativ	SOAP mit Add-On
Android (2.2, API Level 8)	Java SE/Java ME ähnlich	Kaum: XML ja, Rest nur manuell	Stub-Generierung möglich, Datentypen manuell
Windows Mobile 6	.NET	XML und Generierung Datentypen und Stubs, aber nur WS-I Basic Profile 1.1 nutzbar	Für WS-I Basic Profile 1.1 nicht erforderlich; sonst gSOAP möglich
Windows Phone 7	.NET	XML und Generierung Datentypen und Stubs, aber nur WS-I Basic Profile 1.1 nutzbar	Nicht möglich
iOS	Objective C	Kaum: XML ja, WSDL stark eingeschränkt, sonst manuell	gSOAP siehe letzte Zeile
Blackberry OS	Java ME ähnlich	XML ja, JSR-172 [10] ja, sonst manuell	Stub-Generierung möglich, Datentypen manuell
Andere C-basierte Plattformen	C/C++	–	sehr gut (mit gSOAP Aufsatz)

Versteht man die SOAP-Unterstützung in diesem Sinne- und nur dies macht im Kontext von Unternehmensanwendungen Sinn - so ergeben sich auf aktuellen mobilen Plattformen die eher schwachen nativen Nutzungsmöglichkeiten der Tabelle 17.2, die wir im Folgenden etwas genauer analysieren werden.

### 17.3.1 Android

Android ist zurzeit die Plattform für Smartphones mit den am stärksten steigenden Marktanteilen und hat auch bereits eine große Verbreitung. Die Entwicklung von Anwendungen findet gegen eine Java-ähnliche API statt, die vom Umfang der Bibliotheken zwischen Java SE und Java ME angesiedelt ist. Die Verarbeitung von allgemeinen XML-Dokumenten (SAX- und DOM-Parser) ist seit API Level 8 (Android 2.2) vorgesehen. Allerdings ist keine native Unterstützung für SOAP Web Services vorhanden. Die SOAP-Kommunikation muss also manuell oder über eine Zusatzbibliothek realisiert werden. Dies hat die bereits zuvor genannten Nachteile wie komplexeres Deployment und erfordert möglicherweise manuelle Anpassungen für jede neue Version der Android-Plattform. Eine bereits vor Jahren für SOAP auf Java ME entwickelte Bibliothek in diesem Kontext ist kSOAP2; ein Einsatz ist prinzipiell möglich (vgl. [11]).

Die Bibliothek kapselt die Transportschicht, führt die Verarbeitung der SOAP-Nachrichten aus und ermöglicht die Behandlung von Fehlern auf Ebene der Service-Schicht. Die (De-)Serialisierung der SOAP-Nachrichten wird durch Interfaces ermöglicht. Die konkreten Implementierungen der Interfaces bilden die Datentypen der Service-Beschreibung (WSDL) in Klassen ab. Dadurch wird die Semantik des Services abgebildet und die dynamische Formulierung einer Anfrage zur Laufzeit ermöglicht. Allerdings erfolgt die Abbildung der Datentypen in Klassen durch Im-

plementierung der Interfaces nicht automatisiert und ist Aufgabe des Entwicklers. Somit steigt der Aufwand der Implementierung mit Komplexität und Umfang des Web Services. Für die Nutzung einfacher strukturierter Services mit wenigen definierten Datentypen und Funktionen stellt kSOAP2 eine geeignete Lösung dar, für die Integration in komplexe Geschäftsprozesse ist die Bibliothek jedoch eher nicht geeignet.

Insgesamt kann man also feststellen, dass SOAP Web Services auf der Android-Plattform kaum unterstützt werden (dies gilt auch noch für die zum Zeitpunkt des Schreibens aktuelle API Version 12, Android 3.1) oder zumindest einen sehr hohen manuellen Entwicklungsaufwand erfordern.

### **17.3.2 Windows Mobile/Phone**

Microsoft bietet seit Herbst 2010 mit Windows Phone 7 (WP7) eine weitere Plattform für Smartphones an und löst damit den Vorgänger Windows Mobile 6 (WM6) ab. Aufgrund des jungen Alters (der frühen Version) von WP7 bezieht sich dieser Abschnitt auf beide Plattformen und die Darstellung wesentlicher Unterschiede der Unterstützung von SOAP.

WM6 basiert auf Windows CE und unterstützt das .NET Compact Framework 3.5; es beinhaltet damit eine eingeschränkte Version der Windows Communication Foundation (WCF). Die WCF stellt eine einheitliche Plattform und API für die Kommunikation in Service-orientierten Architekturen für .NET-Anwendungen bereit. Die Unterstützung für die Verarbeitung von XML ist hier in vollem Umfang gegeben. Für die Unterstützung von SOAP Web Services werden zwei Bindings unterschieden: BasicHttpBinding und CustomBinding. Das BasicHttpBinding definiert Services nach dem WS-I Basic Profile 1.1. Das CustomBinding ermöglicht durch die Konfiguration weiterer Standards, wie z. B. WS-Security, eine feinere Anpassung der Service-Nutzung an einen Service.

Für WP7 besteht mit Silverlight eine neue Plattform für die Anwendungsentwicklung. Silverlight vereinheitlicht die Entwicklung und ermöglicht so die Wiederverwendung bestehender .NET-Anwendungen und Bibliotheken für WP7. Grundlage sind auch hier das .NETCF und die WCF, allerdings wird der Zugriff durch Silverlight gekapselt. Auch hier ist eine volle Unterstützung der XML-Verarbeitung vorhanden.

Vereinfacht wird die Implementierung für beide Plattformen durch die Generierung eines Client-Stub durch die Entwicklungsumgebung Visual Studio. Bei dieser Generierung werden auch die erforderlichen Datentypen für den Client automatisch erzeugt. Allerdings werden auch hier nicht alle spezifizierten Funktionalitäten der WSDL unterstützt, sondern nur die im BasicHttpBinding definierten Services nach dem WS-I Basic Profile 1.1. Für allgemeine Dienste ist dies oft ausreichend; im Bereich der Sicherheitsfunktionalitäten wird später in diesem Kapitel noch auf wesentliche Einschränkungen eingegangen.

Zusammenfassend bieten insgesamt beide Plattformen eine gute Unterstützung für SOAP Web Services auf Basis des WS-I Basic Profile 1.1 mit geringem Entwicklungsaufwand.

### 17.3.3 iOS

Die Plattform iOS für Apples iPhone und iPad ermöglicht die Entwicklung von Anwendungen in Objective-C mit Einsatz der Cocoa API, welche von dem Desktop-System Mac OS X portiert und für die Anforderungen mobiler Geräte angepasst wurde. Ähnlich wie Android bietet iOS keine native Unterstützung für SOAP. So stehen von Seiten der Plattform lediglich XML-Parser (SAX und DOM) zur Verfügung. Für einen effektiveren Einsatz von SOAP muss auch hier auf externe Lösungen zurückgegriffen werden.

Der bereits beschriebene Ansatz der Vorformulierung der SOAP-Nachrichtenstruktur in einem String und dem Ersetzen einzelner Parameter über String-Operationen stellt lediglich eine Notlösung für einfache Services dar und ist für die Abbildung von Geschäftsprozessen nicht praktikabel.

Eine bessere Lösung verspricht das Projekt wsdl2objc [12], welches einen Objective-C Client-Stub aus einer Service-Beschreibung (WSDL) erzeugt. Allerdings wird der WSDL-Standard nicht in vollem Umfang unterstützt, wodurch die Generierung nicht für jeden Service erfolgreich ist. Diese Lösung scheidet daher für Unternehmensanwendungen ebenfalls aus.

Zusammenfassend existiert für iOS keine Lösung für eine (teil)automatisierte Nutzung von SOAP Web Services. Es bleibt lediglich die manuelle Implementierung, welche sowohl einen hohen initialen Entwicklungsaufwand wie auch großen potentiellen Pflegeaufwand bedeutet.

### 17.3.4 Blackberry OS

Das Betriebssystem Blackberry OS wird für Smartphones des Herstellers Research In Motion (RIM) eingesetzt. Die Blackberry-Plattform ist insbesondere im Unternehmensumfeld weit verbreitet, allerdings wird ein sinkender Marktanteil prognostiziert (vgl. [13]). Für die Ausführung von Anwendungen stellt die Plattform eine Java-ME-Umgebung (MIDP 2.0, CLDC 1.1) bereit, die neben der in MIDP standardisierten APIs weitere Spezifikationen und APIs umsetzt und beinhaltet [14]. Mit Unterstützung des JSR 172 [10] wird die Plattform unter anderem um einen XML-Parser (SAX) und eine Teilmenge der Spezifikation JAX-RPC 1.1 erweitert. Zusätzlich kann mit dem Java ME SDK aus einer WSDL-Beschreibung ein JSR 172-konformer Client-Stub generiert werden. Allerdings umfasst [10] nur eine sehr eingeschränkte Menge von Datentypen, so dass Dienste mit komplexeren Schnittstellen sowie WS-\* Standards nicht verwendbar sind.

Eine alternative Lösung bietet hier die Bibliothek kSOAP2. Wie bereits zu Android beschrieben, wurde kSOAP2 ursprünglich für Java ME entwickelt und eingesetzt. Diese Möglichkeit besteht auch für Blackberry und es gelten die gleichen Vorteile und Nachteile wie für den Einsatz auf Android.

Insgesamt bietet die Blackberry-Plattform eine Möglichkeit zur Stub-Generierung und Nutzung von SOAP Web Services. Allerdings ist diese Lösung nur für sehr eingeschränkte Services nutzbar und dürfte damit für einen Großteil der Unternehmensanwendungen ungeeignet sein. Es bleibt auch hier nur die - mit hohem Entwicklungsaufwand verbundene – manuelle Implementierung.

### **17.3.5 Weitere Plattformen**

In dieser Kategorie wäre einerseits das aktuell noch sehr verbreitete, aber auf dem Rückzug befindliche SymbianOS sowie auch die native Entwicklung auf bereits zuvor betrachteten Plattformen wie z. B. Android Native Development [15] zu nennen. Letzteres hat dann allerdings den massiven Nachteil der potenziell erforderlichen Anpassungen für jede neue, alternative Plattformversion, möglicherweise sogar jedes Gerät. Der Aufwand hierfür ist um ein Vielfaches höher als die Anpassung der in Tabelle 17.2 erwähnten Add-Ons.

Aus dieser Klasse von Plattformen ist uns ebenfalls keine mit nativer SOAP-Unterstützung bekannt. Allerdings gibt es eine sehr ausgereifte, auf alle C-basierten Plattformen portable Bibliothek, die SOAP-Web Services unterstützt. Es handelt sich um ein kommerzielles Produkt (gSOAP, [16]), das einerseits alle eingangs genannten Anforderungen an eine SOAP-Unterstützung erfüllt, andererseits wegen seiner breiten Portabilität interessant ist. Es werden sowohl die automatisierte Erzeugung von Datentypen wie auch Aufruf-Stubs für die Client-Seite aus einer WSDL-Beschreibung unterstützt. Außerdem bietet es die (teilweise) Unterstützung weiterer Standards wie WS-Policy, WS-Addressing, WS-ReliableMessaging und WS-Security. Neben den technischen Möglichkeiten stellt insbesondere für Unternehmens-Anwendungen die flexible Lizenzierung der Bibliothek einen Vorteil dar: gSOAP bietet sowohl eine freie Open Source (GPL) wie auch eine Lizenz für kommerziellen Einsatz, zusätzlich eine Enterprise Edition mit Support.

Als einziger Nachteil bleibt zu nennen, dass das Tool zunächst auf jede neue Version einer Gerät-Plattform portiert werden muss, bevor SOAP Web Services dort zur Verfügung stehen. Da es auf ANSI-C/C++ basiert, sollte der Aufwand dafür jedoch sehr gering sein. Das Tool wurde von uns erfolgreich auf Symbian OS [17] sowie Android im nativen Modus eingesetzt.

## **17.4 Bedeutung von Sicherheitsfunktionen in Enterprise Apps**

Neben den bereits im vorhergehenden Abschnitt diskutierten Vorteilen der SOAP Web Services im Vergleich mit REST-basierten Diensten aus Schnittstellensicht

spielt SOAP seine Vorteile insbesondere dann aus, wenn man die Dienstschnittstellen auch inhaltlich sowie organisatorisch betrachtet. Für SOAP existieren bereits zahlreiche Spezifikationen zur automatisierten Integration spezifischer nicht-funktionaler Anforderungen in den Diensten. Beispiele hierfür sind verlässliche Dienstgüte, Effizienz, Semantik und insbesondere auch Sicherheitsaspekte. Exemplarisch sollen in diesem Kapitel die Sicherheitsaspekte genauer betrachtet werden.

Während REST im Bereich der Sicherheitsfunktionen i. W. die Ende-zu-Ende-Verschlüsselung der gesamten Kommunikation zwischen zwei Kommunikationsendpunkten anbietet, gibt es für SOAP Web Services zahlreiche Spezifikationen des W3C, die eine passgenaue Wahl der gewünschten Sicherheitsfunktionen ermöglichen (z. B. [18, 19]). Dies kann im Bereich von Unternehmensanwendungen in verschiedenen Einsatzszenarien von Bedeutung sein, wie z. B.:

- Individuelle Verschlüsselung bestimmter Nachrichtenelemente für verschiedene Empfänger: Diese Anforderung zeigt sich häufig in organisationsübergreifenden Prozessen, bei denen aus einer Anfrage eines Clients Daten im Laufe eines Prozesses an verschiedene Beteiligte gesandt werden. Nicht zuletzt aus Gründen des Datenschutzes kann es dabei wichtig sein, dass der erste Empfänger bzw. weitere Intermediäre Teile der Nachricht nicht im Klartext lesen können.
- Signierung bestimmter Nachrichtenteile durch Client oder Server: Für die Verbindlichkeit der Kommunikation, die im Unternehmensbereich oft schon aus juristischer Sicht erforderlich ist, sowie der kommunizierten Inhalte ist eine Signierung von Nachrichten oder Nachrichtenteilen erforderlich. Dabei kann dies von relativ einfacher Signierung der gesamten Nachricht durch den Sender bis zu einer individuellen Signierung verschiedener Elemente durch einzelne Prozessbeteiligte reichen.

Anhand dieser exemplarischen Anforderungen im Bereich der Sicherheit wird bereits deutlich, dass die Nutzung von REST Web Services diese nicht alle erfüllen kann. Weitere Anforderungen sowie beliebig komplexe Kombinationen der genannten sind denkbar und in einigen Fällen erforderlich. Dabei ist auch wieder zu berücksichtigen, dass eine manuelle Implementierung dieser Funktionen auf Anwendungsebene natürlich möglich wäre, aber aufgrund der bereits existierenden Spezifikationen für SOAP unnötig ist. Außerdem wird spätestens bei der automatisierten Generierung von Client-Stubs, die sowohl bei sich ändernden Versionen von Diensten als auch bei einer automatisierten Integration von Diensten in Prozesse erforderlich ist, diese Methode nicht mehr möglich sein.

In Enterprise-Umgebungen sind die Anforderungen an die Sicherheitsfunktionen einer Web-Service-Schnittstelle also deutlich komplexer als im Endkundenbereich. Außerdem spielen die Vorteile der einfacheren Anpassbarkeit an neuere Versionen durch die größere Lebensdauer der Dienste eine größere Rolle. Schließlich ist die einfachere Integrierbarkeit in komplexe Prozesse im Unternehmensumfeld von großer Bedeutung. Insgesamt können die spezifischen Sicherheitsanforderungen also im Enterprise-Umfeld nur mit SOAP Web Services erfüllt werden. Nach den eher durchwachsenen Ergebnissen der SOAP-Unterstützung auf aktuellen Plattformen mobiler Geräte stellt sich also nun die Frage, welche der Sicherheitsfunktionen

unterstützt werden bzw. mit welchem Aufwand eine solche Unterstützung ergänzt werden kann.

## 17.5 SOAP Security auf aktuellen Plattformen

Da die meisten aktuellen Plattformen SOAP zusammen mit den erforderlichen Sicherheitsspezifikationen gar nicht nativ unterstützen, wird dieses Kapitel aufgeteilt in plattformspezifische Implementierungen, die nur bei vorhandener nativer Unterstützung Sinn machen, sowie plattformübergreifende Implementierungen, die die Nutzung eines plattformübergreifenden SOAP-Add-On voraussetzen. Ein Überblick der Ergebnisse ist Tabelle 17.3 zu entnehmen.

### 17.5.1 Plattformspezifische Implementierung

Wie bereits aus Tabelle 17.2 ersichtlich, ist es im Prinzip nur die Windows-Plattform, die nativen Support für SOAP Web Services im vollen Umfang bietet. Daher muss in diesem Abschnitt auch nur diese auf die Unterstützung von Sicherheitsfunktionalität untersucht werden.

WM6 bietet für die Sicherung auf Anwendungsebene lediglich eine einzige Konfiguration in Form eines asymmetrischen Bindings. Durch Verwendung des CustomBinding und die Unterstützung einer Teilmenge des WS-Security Standards in Version 1.0 erfolgt die Authentisierung des Service Consumers und des Service Providers durch X.509-Zertifikate. Zusätzlich ermöglichen die Zertifikate die Erzeugung bzw. Überprüfung von Signaturen sowie die Ver- bzw. Entschlüsselung der Nachrichten nach den Standards XML-Signature und XML-Encryption.

Durch die Verwendung von Silverlight in WP7 ergibt sich für die Auswahl der Bindings ein Unterschied zu WM6. Das BasicHttpBinding steht weiterhin zur Verfügung; die Möglichkeit, ein Binding an weitere Standards anzupassen (CustomBinding), besteht jedoch nicht mehr. Die Konfiguration weiterer Web Service Standards

**Tabelle 17.3** Überblick über SOAP-Security-Unterstützung auf mobilen Plattformen

SOAP-Plattform	Geräteplattformen	SOAP-Security-Unterstützung	Entwicklungsaufwand
.NET CF	Windows Mobile 6/Phone 7	4 (eine Konfiguration), 1 (sonst)	5
Java ME und kSOAP2	Android, Blackberry	2	1
C/C++ und gSOAP	iOS, SymbianOS, Android	4	3
.NET und Mono	iOS, Android	4 (eine Konfiguration), 1 (sonst)	3

wird in Silverlight durch das WsHttpBinding ermöglicht, allerdings ist dieses nicht Bestandteil von WP7. Insgesamt kann man also sagen, dass die Unterstützung für Enterprise Apps in WP7 schlechter als in WM6 geworden ist, sofern man nicht mit dem recht allgemeinen BasicHttpBinding auskommt.

Mit dem .NET CF erstellte Anwendungen sind auch auf anderen Plattformen lauffähig, wenn diese mit entsprechender Zusatzsoftware versehen wurden (z. B. Mono für Android). Insofern gilt dieser Abschnitt auch für die plattformübergreifende Entwicklung im folgenden Abschnitt, allerdings mit der Einschränkung, dass eine zusätzliche Abhängigkeit von der Verfügbarkeit der Zusatzsoftware entsteht. Diese kann sich bei langsamem oder ausbleibenden späteren Updates als großer Nachteil erweisen. Native Unterstützung ist daher stets zu bevorzugen.

### **17.5.2 Plattformübergreifende Implementierung**

Ein plattformübergreifendes SOAP-Add-On für Java-ME-Anwendungen und damit für Android und Blackberry relevant ist kSOAP2. Die Bibliothek erleichtert die Verwendung von SOAP durch Kapselung der Transportschicht und Serialisierung der Nachrichten. Allerdings ermöglicht kSOAP2 keine weiteren Sicherheitsfunktionen wie Signaturen und Verschlüsselung. In [11] wurde für Android eine prototypische Erweiterung von kSOAP2 für die Signierung einer Nachricht und die prinzipielle Möglichkeit zur Verschlüsselung gezeigt. Allerdings ist fraglich, ob die Plattform Java ME selbst in Zukunft noch von Bedeutung sein wird.

Das SOAP-Add-On gSOAP ermöglicht, wie bereits zuvor beschrieben, die Nutzung von SOAP Web Services in C und C++. Das wsse-Plugin für gSOAP ermöglicht die Sicherung von SOAP-Nachrichten auf der Anwendungsebene nach dem WS-Security Standard 1.0. Für die kryptografischen Funktionen werden Bibliotheken von OpenSSL verwendet, wodurch sich eine weitere Abhängigkeit für jede Zielplattform ergibt. Für SymbianOS und iOS besteht die Möglichkeit, OpenSSL und damit prinzipiell auch gSOAP mit Security-Plugin zu verwenden; für Android konnte diese Kombination prototypisch erfolgreich getestet werden. Im Folgenden werden einzelne Sicherheitsaspekte genauer betrachtet:

- Sicherheits-Token: Das Plugin ermöglicht das Einfügen verschiedener Tokens für die Authentifizierung der Kommunikationspartner. Zur Verfügung stehen UsernameToken und BinarySecurityToken. Erstere umfasst Benutzername und Passwort (Klartext oder digest), letzteres stellt ein X.509-Zertifikat dar und ermöglicht die Verifikation von Signaturen.
- Signaturen: Die Signierung einer Nachricht bzw. die Verifikation einer Signatur kann sowohl mit einem symmetrischen (sharedsecret, gemeinsamer Schlüssel) als auch mit einem asymmetrischen (X.509) Verfahren erfolgen. Für den Umfang der Signatur bestehen hier keine Einschränkungen, so können der gesamte Body als auch einzelne Elemente einer Nachricht signiert werden.
- Verschlüsselung: Das wsse-Plugin bietet die Möglichkeit der Ver- und Entschlüsselung von Nachrichten mit einem symmetrischen und einem asymmetrischen

Verfahren. Allerdings ist die Verschlüsselung nur für den gesamten Body und nicht für einzelne Elemente/verschiedene Empfänger möglich.

- Zeitstempel: Durch das Einfügen von Zeitstempeln in den Header einer Nachricht wird die Gültigkeit des Sicherheitskontexts der Nachricht eingeschränkt.

Insgesamt zeigt sich für gSOAP eine gute Unterstützung der im WS-Security Standard definierten Sicherungsmechanismen. Allerdings gelten auch hier Einschränkungen, wie z. B. die fehlende Verschlüsselung einzelner Nachrichtenelemente für verschiedene Empfänger. Dennoch stellt gSOAP eine umfangreiche und komfortabel zu nutzende Lösung für SOAP Web Services auf Plattformen dar, die nativ keine Unterstützung für SOAP bieten. Dies gilt insbesondere auch, da neben Sicherheitsfunktionen auch andere übergreifende Standards wie WS-Addressing [20] oder WS-ReliableMessaging [21] unterstützt werden, die im Bereich der Enterprise Services und Apps von großer Bedeutung sind.

Schließlich gibt es noch Implementierungen von Mono (eine Plattform-übergreifende Implementierung des .NET-FW) für mobile Endgeräte. Hier sind bspw. *MonoTouch* für iOS sowie *Mono forAndroid* [22] zu nennen. Diese Frameworks erweitern die entsprechenden Plattformen ebenfalls um SOAP-Unterstützung inklusive Generierung von Datentypen und Client-Stubs. Außerdem wird hierbei wie auf den Windows-Plattformen das WS-I Basic Profile 1.1 unterstützt. Als Nachteile bleiben hier nur die Zusatzinstallation einer Laufzeitumgebung für Mono sowie die prinzipielle Abhängigkeit von einem Drittanbieter-Produkt zu nennen. Diese gelten für gSOAP und Mono jedoch gleichermaßen.

## 17.6 Fazit

Lediglich eine der heute in breitem Einsatz befindlichen Plattformen für mobile Endgeräte stellt eine native Unterstützung für SOAP Web Services in der in Unternehmen vorkommenden Komplexität bereit. Selbst diese Plattform verschenkt aber viele der durch SOAP Web Services und die zugehörigen Sicherheitsspezifikationen ermöglichten Einsatzmöglichkeiten. Implementierungsunabhängige, flexible und bedarfsgerechte Nutzung verschiedener Sicherheitsfunktionen für verschiedene Aspekte von Diensten wird durch die Unterstützung genau einer Sicherheitskonfiguration unmöglich gemacht. So ist es nativ bspw. nicht möglich, einzelne Elemente einer Nachricht für unterschiedliche Empfänger zu verschlüsseln. Immerhin gibt es hier eine gute Grundlage, die auf eine Verbesserung der Nutzungsmöglichkeiten in der näheren Zukunft hoffen lässt.

Alle anderen Plattformen stellen gar keine akzeptable native Unterstützung bereit. Hier ist man also auf die Nutzung einer Zusatzbibliothek angewiesen. Dies hat insbesondere für Unternehmensanwendungen gravierende Nachteile:

- Bei Aktualisierungen des zugrunde liegenden Betriebssystems ist zunächst auf die Anpassung der entsprechenden Bibliothek zu warten, bevor die App für die neue Plattform angepasst und eingesetzt werden kann.

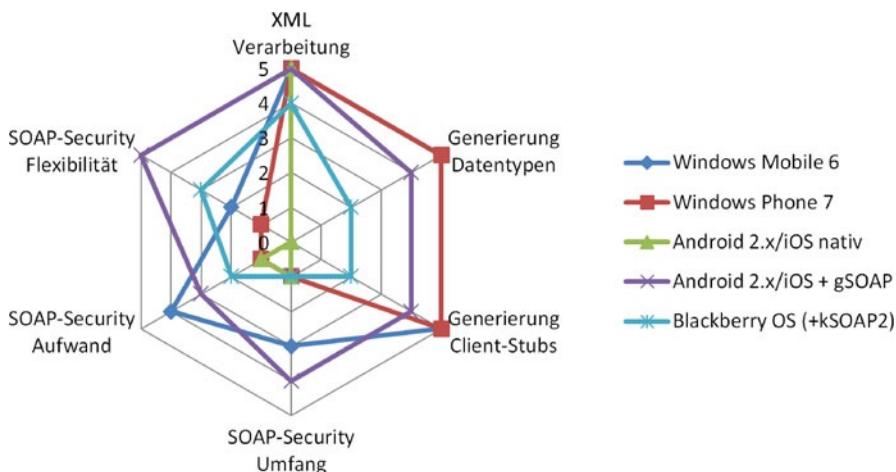


Abb. 17.1 Vergleich aktueller Plattformen (ggf. mit Add-Ons) für Enterprise Apps

- Der Entwicklungszyklus der Betriebssysteme mobiler Endgeräte ist zurzeit so kurz, dass man eine Vielzahl solcher Systeme mit einer App unterstützen muss; dies wird durch die Nutzung nicht nativer Schnittstellen stark erschwert.
- Das technologische Risiko wird durch ein komplexeres Szenario (Abhängigkeit von Gerät-, Betriebssystem- und Bibliothekshersteller) ebenfalls größer.

Immerhin kann durch den Einsatz einer auf mehreren Plattformen verfügbaren Bibliothek (wie gSOAP) der nicht zu unterschätzende Aufwand für die Pflege einer App für mehrere Geräteplattformen etwas reduziert werden. Dies dürfte auch bei Unternehmensanwendungen eine große Rolle spielen, da die Nutzer der Apps weiterhin ihre ohnehin schon vorhandenen Geräte nutzen möchten und nicht ein Gerät speziell für eine Firmen-App verwenden wollen. So ist die Unterstützung mehrerer Plattformen für eine Unternehmensanwendung von großer Bedeutung. Hält man sich an das WS-I Basic Profile 1.1 (und kann mit den damit verbundenen o. g. Einschränkungen leben, was im Unternehmensumfeld eher unwahrscheinlich ist), so erreicht man auch bei Verwendung von .NET/Mono immerhin eine Unterstützung der Plattformen Windows Mobile/Phone, iOS und Android.

Betrachtet man die Sicherheitsaspekte in Web Services, so kann man festhalten, dass die erforderlichen Spezifikationen bereits seit einiger Zeit in guter Qualität zur Verfügung stehen. Umfassende Funktionalität, Interoperabilität und Flexibilität sind berücksichtigt, so dass einem Einsatz aus konzeptioneller Sicht nichts entgegensteht. Eine Verwendung im Unternehmensumfeld findet sich bisher aber selbst im Backend in deutlich geringerem Maße. Somit ist es verständlich, dass einerseits auf den wesentlich neueren mobilen Geräten bisher kaum eine umfassende Nutzung der Sicherheitsfunktionalitäten stattfindet. Andererseits wird so auch noch kein größerer Druck auf die Gerätehersteller ausgeübt, Funktionen in diesem Bereich in ihren Produkten vorzuhalten.

## 17.7 Ausblick

Der breite Einsatz von mobilen Apps im Unternehmensumfeld lässt bisher noch auf sich warten. In diesem Kapitel haben wir aufgezeigt, warum dies aus technologischer Sicht verständlich ist. Um also Apps in die zumeist bereits in Software gegossenen Geschäftsprozesse integrieren zu können, fehlen bisher einfach die technischen Möglichkeiten oder sie sind mit sehr großem Aufwand verbunden.

Insgesamt muss man außerdem feststellen, dass der Einsatz von mobilen Apps in komplexen Geschäftsprozessen heute noch ein recht großes Risiko beinhaltet. Dies gilt sowohl für den möglichen initialen Entwicklungsaufwand dieser Apps selbst wie auch für die erforderliche Wartung und Pflege im Laufe der Zeit. Da die Gerätehersteller sich aktuell eher an den Bedürfnissen der Endverbraucher denn der Unternehmensanwender orientieren, dürfte diese Aussage auch noch für die nähere Zukunft gelten. Im Prinzip handelt es sich um ein klassisches Henne-Ei-Problem: Apps werden im Unternehmensumfeld kaum eingesetzt, da die technischen Möglichkeiten nur gering sind; andererseits wird die technische Unterstützung nicht verbessert, da es nur wenig Interessierte gibt, die diese nachfragen.

Ein Ausweg aus dieser Situation könnte darin bestehen, dass einerseits die Gerätehersteller zumindest rudimentäre Funktionen bereitstellen, die es den Unternehmen ermöglichen, erste Apps zum Einsatz zu bringen. Dies könnte die Unternehmen vom Potenzial überzeugen und zu einer größeren Nachfrage nach fortgeschrittenen Funktionalitäten führen.

Aus unserer Sicht stellt der Bereich der mobilen Apps für Unternehmensanwendungen einen Markt mit extrem großem Potenzial dar. Sollte dies auch einer der Gerätehersteller erkennen und durch entsprechende Integration der erforderlichen Funktionalitäten (SOAP Web Services und weitere WS-\* -Spezifikationen) in sein Betriebssystem unterstützen, so könnte ihm dies einen entscheidenden Vorteil in den nächsten Jahren verschaffen. Vielleicht sorgt dies sogar noch einmal für einen größeren Wechsel bei den Marktanteilen der Plattformen. Die Offenheit der eigenen Plattform und freie Entscheidung über die einzusetzende Software wäre an dieser Stelle auf jeden Fall hilfreich.

## Literaturverzeichnis

1. Erl T (2009) SOA Design Patterns. Boston, MA, Prentice Hall
2. Krafzig D, Banke K, Slama D (2004), Enterprise SOA:Service-Oriented Architecture Best Practices. Prentice-Hall, Upper Saddle River, USA
3. Booth D, Liu CK (2007) Web Services Description Language (WSDL) Version 2.0, W3C Recommendation. <http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/> 26. Oktober 2011
4. Sahai A, Graupner S (2005) Web Services in the Enterprise: Concepts, Standards, Solutions, and Management, Springer
5. Chatterjee S, Webber J (2005) Developing Enterprise Web Services: An Architect's Guide, Prentice-Hall

6. Heffner R SOAP Versus REST: A Comparison, Forrester Research, [http://www.forrester.com/rb/Research/soap\\_vs\\_rest\\_comparison/q/id/35361/t/2](http://www.forrester.com/rb/Research/soap_vs_rest_comparison/q/id/35361/t/2) 26. Oktober 2011
7. <http://www.osxentwicklerforum.de/index.php?page=Thread&threadID=12853> 26. Oktober 2011
8. W3C: OWL-S Semantic Markup Web Services, <http://www.w3.org/Submission/OWL-S/> 26. Oktober 2011
9. MarwaMabrouk (2008) OpenGIS Location Services: Core Services. OGC Inc.
10. Calder B et al JSR 172: Java ME Web Services Specification, <http://jcp.org/aboutJava/communityprocess/final/jsr172/index.html> 26. Oktober 2011
11. Kleiner C, Schneider (T2011) Securing SOAP Web Services for Mobile Devices on Different Platforms. In: Höpfner, Specht, Ritz, Bunse (eds.): MMS 2011: GI LNI 185/2011, pp 25–38.
12. <http://code.google.com/p/wsdl2objc/> 26. Oktober 2011
13. <http://www.gartner.com/it/page.jsp?id=1622614> 26. Oktober 2011
14. [http://docs.blackberry.com/en/developers/deliverables/5580/Support\\_for\\_standard\\_Java\\_APIS\\_446981\\_11.jsp](http://docs.blackberry.com/en/developers/deliverables/5580/Support_for_standard_Java_APIS_446981_11.jsp) 26. Oktober 2011
15. Android Native Development Kit. <http://developer.android.com/sdk/ndk/index.html> 26. Oktober 2011
16. Genivia Inc. gSOAP toolkit for SOAP/XML web services, <http://www.genivia.com/Products/gsoap/index.html> 26. Oktober 2011
17. Bertram J, Kleiner C, Zhang D (2010) Implementation of Generic OpenLS-compliant Web Service Clients for Mobile Devices, 6.GI/ITG KuVS-FG, Bonn, pp 91–100.
18. Nadalin A, Kaler C, Monzillo R, Hallam-Baker P (2004) WebServices Security: SOAP Message Security 1.1. OASIS. <http://docs.oasis-open.org/wss/v1.1/>. 26. Oktober 2011
19. Nadalin A, Goodner M et al WS-SecureConversation 1.3. OASIS, <http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation1.3-os.html> 26. Oktober 2011
20. W3C: Web Services Addressing (WS-Addressing), W3C Member Submission 10 August 2004, <http://www.w3.org/Submission/ws-addressing/> 26. Oktober 2011
21. OASIS: Web Services Reliable Messaging TC WS-Reliability 1.1, [http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws\\_reliability-1.1-spec-os.pdf](http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf) 26. Oktober 2011
22. <http://monotouch.net/> 26. Oktober 2011
23. Ekelhart A, Fenz S, Goluch G, Steinkellner M, Weippl E (2008) XML security – A comparative literature review. *J Syst Softw* 81:1715–1724
24. Gudgin M, Hadley M et al (2007) SOAP Version 1.2 Part 1: Messaging, W3C Recommendation. <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/> 26. Oktober 2011
25. Ferris C et al (ed) (2006) Web Services Interoperability Organization Basic Profile Version 1.1, <http://www.ws-i.org/Profiles/BasicProfile-1.1.html> 26. Oktober 2011

---

# Kapitel 18

## Ein interoperables Bezahlverfahren für In-App- und Online-Käufe

Toni Goeller und Tet Hin Yeap

**Zusammenfassung** Nach 15 Jahren praktischer Nutzung von Online-Bezahlverfahren gibt es etliche voneinander unabhängige Lösungen sowie davon abgeleitete oder spezialisierte In-App-Bezahlverfahren. Die mangelnde Interoperabilität der existierenden Lösungen hat Nachteile für Kunden und Anbieter und führt so zu höheren Transaktionskosten und suboptimaler Akzeptanz. In diesem Kapitel werden nach einem geschichtlichen Überblick Kriterien für eine Bezahllösung entwickelt, die die bestehenden Nachteile aufhebt. Eine beispielhafte Lösung, die global skalierbar, wettbewerbsfreundlich, auf die einfache Nutzung vorhandener Infrastruktur und auf geringe Transaktionsgebühren ausgelegt ist, wird vorgestellt. Abschließend werden mögliche Einführungsszenarien und die Vorteile für alle beteiligten Parteien diskutiert.

### 18.1 Geschichte und Marktsituation

Bis in die zweite Hälfte der neunziger Jahre war die Auswahl an Bezahlverfahren mit nennenswerter Verbreitung überschaubar. Bezahlung über nationale Grenzen ließ sich im Wesentlichen über Kreditkarten, internationale Überweisungen oder über Bargeldanweisungen realisieren.

---

Toni Goeller  
Toposis Corp., Ottawa, Ontario, Canada,  
E-mail: toni.goeller@toposis.com

Tet Hin Yeap  
University of Ottawa, School of Electrical Engineering and Computer Science, Ottawa, Ontario, Canada,  
E-mail: yeap@site.uottawa.ca

Mit der zunehmenden Nutzung des Internets und den damit verbundenen Geschäftsmodellen („Internetblase“) rückten auch Online-Bezahlverfahren in den Fokus von innovativen Unternehmen.

In der Folge entstand eine große Zahl an Lösungen für die Online-Bezahlung auch kleinster Beträge. Eine selektive Übersicht findet sich z. B. in [1]. Durch die Ausrichtung an neuen Geschäftsmodellen im Rahmen der ersten „Internetblase“ waren die meisten dieser Lösungsangebote durch folgende Eigenschaften gekennzeichnet:

- Monopolansatz ohne Interoperabilität mit anderen Lösungen: Investoren wurden umso mehr angezogen, wenn eine Beherrschung eines Marktsegments Teil des Geschäftsansatzes eines jungen Unternehmens war.
- Innovationsansatz ohne Maximierung der Nutzung vorhandener Infrastruktur: Kostenminimierung bei Investment und Betrieb war für Investoren damals bedeutungslos; nennenswerte Kosten standen für einen ernsthaften Markteintritt.

Mit zunehmender Nutzung des Internets wurde der Verkauf physischer Güter über das Internet zum Standard. Gleichzeitig nahmen die Geschäftsmodelle für den Verkauf virtueller Güter und Dienstleistungen zu. Zunächst wurden Waren angeboten, die gleichzeitig auf physischen Medien erhältlich waren, wie Software, Musik, Fotos und Videos. Der Online-Verkauf für diese Waren erlaubt neben der höheren Liefergeschwindigkeit auch eine flexiblere Aufteilung des Angebots wie z. B. den Verkauf einzelner Lieder oder Fotos, der bei der physischen Auslieferung zu unattraktiven Preisen führen würde. Diese Entwicklung erhöht das Interesse an Micropayment, das niedrige Preise für Einzelmedien ermöglicht.

Für den zunehmenden Erfolg originär virtueller Güter stehen Klingeltöne und – in weitaus größerem Umfang – der Verkauf virtueller Güter im Rahmen von Online-Spielen. Wenn einzelne Ausrüstungsgegenstände spontan im Spiel erwerbbar sein sollen, sind auch hier Micropayment und eine Vorab-Authentifizierung gefragt.

Da Online-Spiele in zunehmendem Maß auch von mobilen Terminals gespielt werden, wandelt sich der In-Game-Verkauf virtueller Ausrüstungsgegenstände zunehmend zum In-App-Verkauf. Falls es gelingt, über den Browser einen standardisierten Zugriff auf die Gerätehardware – z. B. Grafikhardware und GPS/Galileo – zu ermöglichen, kann der Schwerpunkt dieses Handels sich wieder vom In-App-Verkauf zum klassischen Online-Verkauf zurück wandeln. Deshalb sollten zukunftsichere Bezahlverfahren gleichermaßen für In-App- wie für Internet-Käufe einsetzbar sein.

Durch diese Marktverschiebungen haben sich auch Veränderungen in der Nutzung von Bezahlsystemen ergeben. Derzeit sind folgende Systeme überregional verbreitet:

- Kreditkarten: Neben der klassischen Card-Not-Present-Transaktion unter Angabe von Kartennummer, Karteninhaber und Ablaufdatum bieten die großen Kreditkartenunternehmen weitere Sicherheitsmaßnahmen an, durch deren Einsatz Händler ihr Zahlungsausfallrisiko reduzieren können (z. B. Sicherheitscode, Mastercard SecureCode, Verified by Visa).

- Global verfügbare Online-Bezahlsysteme: Derzeit sind hauptsächlich iTunes (Medien, Apps, In-App-Bezahlung), PayPal (Medien und physische Güter), Google (Apps), sowie amazon (Medien) verbreitet. PayPal und Google bieten seit kurzem auch In-App-Payment-Schnittstellen an. Es ist zu erwarten, dass auch Facebook Credits, ein Prepaid-Konto von Facebook, eine weit verbreitete Währung für Angebote und Dienste wird, die über das führende soziale Netzwerk kommuniziert werden.
- In-Game-Bezahlsysteme: Hier sind aufgrund des Erfolgs der Online-Spiele auf Facebook vor allem Zynga, Facebook und Google – nach der Übernahme von Jambool – zu nennen.

Mobile Bezahlverfahren sind fast in jedem Land verfügbar und gut erforscht – siehe z. B. [2] – sind jedoch bestensfalls regional interoperabel und durch hohe Transaktionskosten limitiert [3]. Besonders bedauerlich an der gegenwärtigen Marktsituation ist es, dass es praktisch keine micropaymentfähigen Prepaid- und Postpaidkonten gibt, die sparten- und anbieterunabhängige Bezahlungen ermöglichen.

## 18.2 Geldakquisition und Transaktionsgebühren

Wesentlich für die Gebühren eines Bezahlsystems sind die Geldeinzugsmethoden der Bezahlverfahren, da diese häufig über andere Verfahren kaskadiert werden. Über die daraus folgende Gebührenkaskadierung definiert sich auch eine untere Schranke für die Gebühren eines Bezahlsystems. Für die Begleichung kleiner Forderungen (Micropayment) sind vor allem die absoluten Gebühren, d. h. die Sockelbeträge der Gebühren, von Bedeutung.

### Direkte Verfahren:

- Bankeinzug oder Überweisung: Niedrigste absolute Gebühren im Centbereich für große Anbieter, keine prozentualen Gebühren. Verfahren länder- bzw. regionsspezifisch.
- Bareinzahlung: Für Prepaid-Konten oder Konten, die Guthaben führen können. Die Einzahlungs-Infrastruktur bzw. die Händler, die die Aufladung bestimmter Prepaid-Konten anbieten, sind länderspezifisch. Hohe prozentuale Gebühren im Bereich von 10 % der eingezahlten Summe.

### Beispiele für aggregierte Verfahren:

- Kreditkarte: Aggregation verschiedener Bankeinzugsmethoden. Mittlere prozentuale Gebühren im Bereich von 2 bis 5 % der Transaktionssumme. Hohe absolute Gebühren im Bereich von 30 Cent.

- PayPal: Aggregation verschiedener Bankeinzugsmethoden und Kreditkarten. Mittlere Gebühren im Bereich von 3 % der Transaktionssumme. Mittlere absolute Gebühren zwischen 10 und 35 Cent.
- Facebook: Aggregation verschiedener Bezahlverfahren wie PayPal, Bankeinzugsmethoden und Kreditkarten. Hohe prozentuale Händlergebühren von 30 % der Transaktionssumme.
- Rechnung über anderen Anbieter: Aggregation verschiedener Bankeinzugsmethoden. Anbieter mit Kundenkontakt länder- bzw. regionsspezifisch. Gebühren abhängig von der Zahl der eigenen und fremden Forderungen, die über die Rechnung beglichen werden. Bei einer hohen Forderungszahl sind niedrige absolute und prozentuale Transaktionsgebühren möglich.

### 18.3 Kriterien für ein globales In-App- und Online-Bezahlsystem

Die im Abschn. 18.1 skizzierte Geschichte erklärt einige der im Folgenden aufgelisteten Kriterien sowie die Tatsache, dass etliche Kriterien durch existierende Bezahlverfahren nur eingeschränkt erfüllt werden.

Folgende Kriterien sollen betrachtet werden, s. a. [4]:

1. Kundenbasis: Aus Händlersicht sollen mit einem Bezahlverfahren möglichst viele potenzielle Kunden erreicht werden, idealerweise die gesamte Weltbevölkerung. Kunden sollten eine Organisation ihres Vertrauens als ihren Vertragspartner für das Bezahlverfahren auswählen können – s. [5] zur Bedeutung des Vertrauens in den Vertragspartner.
2. Händlerbasis: Aus Kundensicht sollen idealerweise alle innerhalb einer App oder online angebotenen Güter über ein einziges Bezahlverfahren zu erwerben sein. Dies reduziert Registrierungsaufwand und die Anzahl von Kopien der Zahlungsdaten.
3. Benutzerfreundlichkeit: Wesentlich für eine Vereinfachung des Kaufprozesses ist die automatische Authentifizierung des Kunden über eine sichere Identifikationseinheit, z. B. eine eingebuchte SIM-Karte. Ein einfacher Kaufprozess ist umso wichtiger, je kleiner der zu bezahlende Betrag ist. So wird kaum ein Kunde vor einem Telefonanruf einen Benutzernamen, ein Passwort oder eine SMS-Bestätigung eingeben wollen.
4. Transaktionskosten: Aus Kunden- und Händlersicht sollen die Transaktionskosten möglichst niedrig sein. Erreichbar wird eine deutliche Reduzierung der Transaktionskosten durch ein Bezahlsystem, das Wettbewerb fördert. Für das Micropayment sind niedrige Mindestgebühren entscheidend. Wenn Preise unter 5 Cent realisiert werden sollen, müssen die Minimalgebühren bei einzelnen Cent liegen.
5. Risikozuweisung: Ein In-App- und Online-Bezahlverfahren sollte hinsichtlich der Risikozuweisung flexibel sein, so dass Händler die Transaktionsgebühren durch Risikoübernahme minimieren können.

## 18.4 Existierende globale Online-Bezahlssysteme

Um Investitionen und Kundenakquisekosten zu vermeiden, sollen bei der Einführung eines neuen Bezahlssystems möglichst existierende Infrastruktur und existierende Kundenbeziehungen genutzt werden.

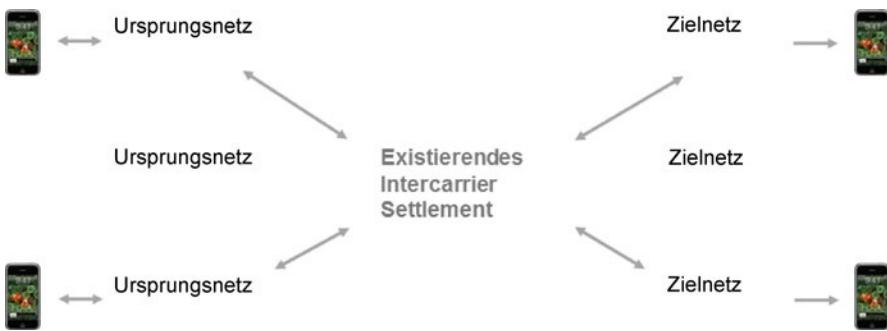
Im Wesentlichen lassen sich zwei Bezahlssysteme als global und interoperabel bezeichnen:

- Kreditkarten
- Intercarrier-Verbindungen in der Telekommunikation.

Die Kriterien aus Abschn. 18.3 werden durch das Intercarrier-Regime in der Telekommunikation mit einer Ausnahme in höherem Maße erfüllt als durch Kreditkartentransaktionen:

1. Kundenbasis: Gesamtzahl der Telekommunikationsverträge liegt weltweit über 6 Milliarden – Mobil- und Festnetzverträge nach [6] –, was bei konservativer Abschätzung der Mehrfachnutzung – eine Person kann mehrere Verträge halten, andererseits können mehrere Personen einen Vertrag nutzen – auf mehr als 3 Milliarden Telekommunikationskunden schließen lässt. Die Verbreitung von Kreditkarten ist insbesondere in Asien, Afrika und Europa deutlich geringer.
2. Händlerbasis: Hier haben Kreditkarten einen Vorteil, da die meisten Online-Händler Kreditkartenzahlungen anbieten.
3. Benutzerfreundlichkeit: Telekommunikationsunternehmen führen für Sprach- und Datenverbindungen eine automatische Authentifizierung ohne Beteiligung des Nutzers durch. Bei Prepaid-Kunden findet sogar eine Echtzeitprüfung der Zahlungsfähigkeit und ggf. eine Reservierung des Rechnungsbetrags statt.
4. Transaktionskosten: Durch die höhere Anzahl an Transaktionen – Telefonengespräche, Datensessions – liegen die Kosten pro Transaktion in der Telekommunikationsbranche niedriger als bei der Kreditkartenabrechnung.
5. Sowohl bei Kreditkarten als auch in der Telekommunikation existieren verschiedene Risikozuweisungen. Die Kreditkartenunternehmen unterscheiden zwischen Card Present – das Kreditkarten-Label trägt das Risiko – und Card Not Present – der Händler trägt das Risiko. Im Intercarriergeschäft trägt üblicherweise das Ursprungsnetz das Risiko, es gibt jedoch auch Fälle, in denen der Anbieter das Risiko trägt, wie zum Beispiel in der Abrechnung von offline gebillten Mehrwertdiensten in Deutschland.

Nimmt ein Telekommunikationskunde die Leistung eines anderen als seines eigenen Anbieters in Anspruch, im klassischen Fall durch einen Anruf zu einem Anschluss in einem fremden Netz, so geschieht der Verbindungsaufbau über ein Intercarrier-Gateway. An diesem Gateway werden vereinbarte Bedingungen überprüft, im Positivfall wird das Gespräch vermittelt und die Terminierung wird dem Ursprungsnetzbetreiber in Rechnung gestellt. Mit Ausnahme von Sonderregeln für spezielle Rufnummern muss der Ursprungsnetzbetreiber die Terminierungsgebühr bezahlen, unabhängig davon, ob der Anrufer das Gespräch bezahlt.



**Abb. 18.1** Grundprinzipien der Intercarrier-Abrechnung

Es kann zwischen Ursprungs- und Zielnetz ein oder mehrere Transitnetze geben, um die Verbindungen zwischen Netzbetreibern mit geringem bilateralem Verkehrs-aufkommen zu aggregieren und die Anzahl der Intercarrierverträge zu reduzieren. In diesem Fall rechnen jeweils die direkten Schnittstellenpartner miteinander ab. Praktisch jeder Kunde kann mit jedem anderen Kunden in der Welt kommunizieren, da durch die Transitnetze die Kosten eines direkten Vertragsabschlusses und einer direkten Abrechnungsbeziehung zwischen den meisten Netzbetreibern vermieden werden.

Sowohl die Kreditkartenabrechnung als auch die Intercarrier-Abrechnung werden seit Jahrzehnten erfolgreich genutzt und umfassen eine Vielzahl involvierter Parteien. Das Komplexitätsargument ist somit kein Grund, der gegen ein Bezahlverfahren spricht, das an einer der beiden Lösungen orientiert ist.

## 18.5 Systemlösung für globale In-App- und Micropayments

Will man die Intercarrier-Abrechnung auf In-App- und Online-Bezahlungen ausweiten, muss die Identität des Ursprungsnetzes gegenüber dem Zielnetz gesichert ausgewiesen werden.

Zudem ändern sich die Rollen: Der Angerufene ist ein Anbieter, der digitale oder online vermittelte Dienste erbringt bzw. Produkte verkauft. Dies entspricht der Vermittlung von Mehrwertdiensten in der Sprachtelefonie. Das Zielnetz bietet dem Anbieter Zugang zur Bezahlmethode „Intercarrier Micropayment“ bzw. aggregiert Inhalte und Angebote aus Sicht der Bezahlmethode.

Die Preise für die angebotenen Dienste und Güter legt der Anbieter fest, abgesehen von ursprungsseitigen Gebühren und Steuern. Es handelt sich also in der Terminologie der deutschen Intercarrier-Abrechnung um einen offline-gebillten Dienst [7].

Die technische Lösung für dieses Bezahlverfahren läuft entsprechend der Sequenz in Abb. 18.3ab. Varianten sind möglich, solange die Grundprinzipien der Identifikation, der Abrechnung und der Preisfestlegung erhalten bleiben.

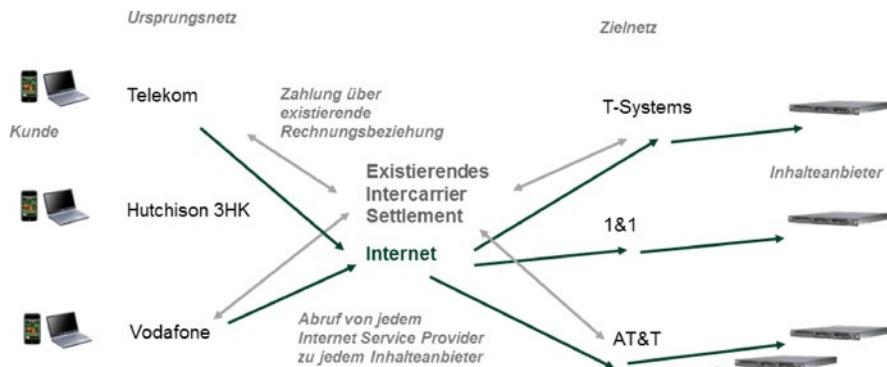


Abb. 18.2 Erweiterung des Intercarrier Micropayment auf Online-Bezahlungen

1. Der Kunde benutzt eine schon erworbene App oder ruft eine öffentliche Seite eines Anbieters auf.
2. Der Anbieter generiert Links zu Kaufangeboten in der Seite bzw. generiert einen kundenspezifischen Link zur Bezahlung vorab gewählter Elemente (Checkout). Die Links enthalten als Prefix eine Domäne, die für alle Teilnehmer am Bezahlverfahren gleich ist, z. B. <http://pay.yapay.biz/www.spiegel.de/sport/dortmund-bayern-okt-2011.html?price=1000&currency=EUR>. Sowohl bei der In-App-Bezahlung als auch in der Online-Bezahlung werden HTTP-Links mit den beschriebenen Informationen verwendet. Im Falle der In-App-Bezahlung hat die App erweiterte Möglichkeiten zur visuellen Integration der unten beschriebenen Dialoge. Dies erfordert allerdings auch ein Review der App und ihres Anbieters, um sicherzustellen, dass die gestalterischen Freiheiten nicht zur Unterdrückung von Bestätigungsanfragen und damit zur Täuschung des Benutzers eingesetzt werden.
3. Ist der Kunde in seinem Ursprungsnetz eingebucht und nimmt der Ursprungsnetzbetreiber am Bezahlverfahren teil, wird diese Domain über netzinterne DNS-Einstellungen auf ein internes Netzelement, den Originating Payment Server (OPS), geleitet. Selbstverständlich lässt sich dieses Netzelement als Cluster oder in einer georedundanten Konfiguration aufbauen.  
Ist der Kunde nicht in sein Ursprungsnetz eingebucht, wird er über eine Ursprungsnetzauswahl zum Originating Payment Server seines Anbieters geleitet.
4. Der OPS authentifiziert den Benutzer automatisch, wenn er im eigenen Netz eingebucht ist, bzw. manuell – über Username und Passwort, Zertifikate, SecurID oder ein anderes Verfahren – wenn der Benutzer über ein Fremdnetz eingebucht ist, z. B. ein Hotel-WLAN.
5. Der OPS rechnet, falls nötig, den Preis in die Währung des Ursprungsnetzes um und fügt ggf. lokale Steuern und eigene Gebühren zum Preis hinzu. Diese Gebühren können vom Vertragstyp des Benutzers abhängig sein, da z. B. das Prepaid-Recharging über Prepaid-Voucher höhere Payment Acquisition Cost aufweist als die Payment Collection für eine Postpaid-Rechnung.

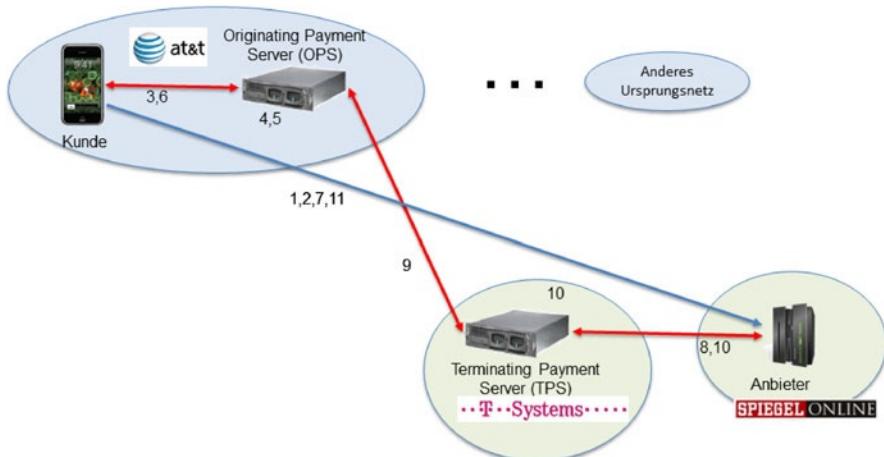


Abb. 18.3 Ablauf des Kaufvorgangs

6. Dann erfragt der OPS das Einverständnis des Benutzers zum Kaufvorgang und zum berechneten Preis. Diese Abfrage enthält die Signatur des Ursprungsnetzes für den Preis, den gewünschten Inhalt und die Identität des Ursprungsnetzes.
7. Im Gutfall stimmt der Benutzer der Preisbestätigung zu. Diese Zustimmung wird zunächst zum Anbieter weitergeleitet.
8. Der Anbieter lässt die signierte Bestätigung vom Zielnetzbetreiber im Netzelement Terminating Payment Server (TPS) verifizieren.
9. Bei positivem Verifikationsergebnis fordert der Zielnetzbetreiber den Ursprungsnetzbetreiber auf, den Kunden mit dem Endkundenpreis zu belasten.
10. Der Zielnetzbetreiber schreibt einen Datensatz für die Intercarrier-Abrechnung und quittiert die Validierung an den Anbieter.
11. Der Anbieter liefert die gewünschte Ware oder den gewünschten Dienst aus.

## 18.6 Abrechnung und Gebühren

An den oben beschriebenen Kaufvorgang schließen sich folgende Abrechnungsvorgänge an:

- Abrechnung zwischen Ursprungsnetz und Kunde – soweit diese nicht schon während des Kaufvorgangs über ein Prepaidkonto vollzogen wurde.
- Abrechnung zwischen Ursprungsnetz und Zielnetz. Diese kann nach dem Vorbild und mit den Werkzeugen der klassischen Intercarrier-Abrechnung erfolgen, hat aber den Vorteil einer sicheren Identifikation des Ursprungsnetzes in jeder einzelnen Transaktion.

- Abrechnung zwischen Zielnetz und Inhalteanbietern. Diese kann nach dem Vorbild und mit den Werkzeugen der Service-Provider-Auszahlung in der sprachbasierten Telekommunikation erfolgen.

Wesentlich ist, dass die Abrechnungsverfahren so gestaltet sind, dass sie bestehende Abrechnungswerkzeuge nutzen können, die schon in ihrer ursprünglichen Anwendung einen hohen Durchsatz bei niedrigen Transaktionskosten bieten.

Da die vorgeschlagene Lösung Kunden- und Anbieterbeziehung mit dem Ziel einer globalen Skalierbarkeit trennt, muss sie auch zwei unabhängige Gebührenkomponenten vorsehen:

- anbieterseitige Gebühr, die das Zielnetz dem Anbieter verrechnet. Diese Gebühr ist im vom Ursprungsnetz geforderten Preis enthalten und wird durch den Wettbewerb zwischen unterschiedlichen Zielnetzen begrenzt.
- kundenseitige Gebühr, die das Ursprungsnetz zusätzlich zu lokalen Steuern auf den Kundenpreis aufschlägt. Diese Gebühr wird bei der Einholung der Bestätigung des Benutzers als Preiskomponente ausgewiesen. Sie wird durch den Wettbewerb zwischen unterschiedlichen Ursprungsnetzen begrenzt.

Sofern Ursprungsnetz und Zielnetz keine direkte Abrechnungsbeziehung haben, kann ein Teil der Gebühren in das Clearing durch andere Zahlungsdienstleister oder dedizierte Clearinghäuser investiert werden.

## 18.7 Mögliche Einführungsszenarien

Wie in vielen aktuellen Innovationsfeldern werden sich neue Bezahlverfahren zunächst nicht in Westeuropa und Nordamerika durchsetzen. Das liegt am hohen Anteil von Konten- und Kreditkarteninhabern, aber auch an einem weniger preissensitiven Umfeld, in dem Micropayment eine geringere Dringlichkeit gegenüber Paketangeboten und Abonnements hat.

Da Netzbetreiber und Banken in den letzten 10 Jahren keine Vision für globale Online-Bezahlverfahren umgesetzt haben, ist es weiterhin unwahrscheinlich, dass Banken oder Netzbetreiber den ersten Schritt bei der Einführung eines solchen Verfahrens gehen.

Betrachtet man das Verfahren und seine Voraussetzungen neu, ist es keineswegs auf Telekommunikations-Netzbetreiber beschränkt. Lediglich bei der automatischen Netzauthentifizierung von Benutzern haben Netzbetreiber noch ein Alleinstellungsmerkmal. Dieses Alleinstellungsmerkmal reduziert sich mit zunehmender Verbreitung von RFID-Chips außerhalb der SIM-Karte in Mobiltelefonen.

Die Rolle des „Ursprungsnetzes“ kann jede Organisation mit Abrechnungsbeziehungen zu Kunden einnehmen, die Rolle des „Zielnetzes“ jede Organisation, die Inhalte und Dienste anbietet oder Rechnungsstellung und Inkasso für Inhalte- und Diensteanbieter übernimmt.

Beispielhaft wird dies in Abb. 18.4 gezeigt. Hierbei ist zu beachten, dass die Quasimonopolisten unter den Online-Bezahlverfahren, die sowohl Kunden- als auch

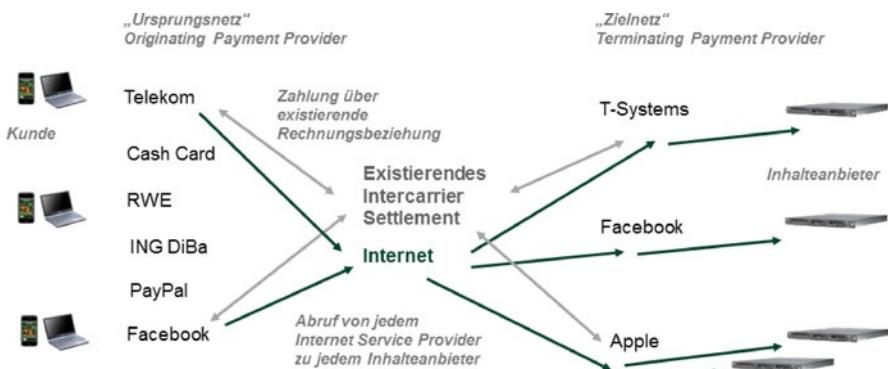


Abb. 18.4 Erweiterung des Intercarrier Micropayment auf Online-Bezahlungen

Händlerverträge haben, beide Rollen ausfüllen können, allerdings nur für ihre eigenen Kunden und Händler.

Mögliche Ansätze für die Markteinführung sind:

### Cash Cards

Sobald Cash Cards über eine Aggregation verschiedener Anbieter auf ein großes Angebot Zugriff haben, werden sie folgende Vorteile ausspielen:

- Unkomplizierter Erwerb und unkomplizierte Nutzung, ohne ein Bankkonto vorzusetzen;
- Prepaid-Konto, eine Prüfung auf Kreditwürdigkeit und das zugehörige Risiko entfällt.

### Peer-to-Peer Selling

Auf Social Networks sind unzählige Medien verfügbar, von denen ein Teil durchaus für Kleinstbeträge Käufer finden würde. Peer-to-Peer Selling auf Social Networks könnte somit zu einer neuen Killerapplikation für globales Micropayment werden.

## 18.8 Zusammenfassung

Das vorgestellte globale Bezahlverfahren zeichnet sich durch signifikante Vorteile für alle Beteiligten aus:

## Kunden

Die Kunden können eine bestehende Vertragsbeziehung nutzen und müssen sich nicht bei vielen Quasi-Monopolisten gesondert registrieren, um auf alle gewünschten Inhalte zugreifen zu können.

Der existierende Vertragspartner, zu dem ohnehin schon ein Vertrauensverhältnis besteht, kann die Zahlung vermitteln, ohne Daten an Dritte weiterzugeben. Mit dem Schutz der Daten erhält der Vertragspartner auch das Vertrauensverhältnis zum Kunden und die Fortführung seines Geschäftes. Inhalteanbieter können durch Sonderangebote eine direkte Vertragsbeziehung – zum Beispiel ein Abonnement bei Medien – anbieten. Diese Vertragsbeziehung setzt jedoch ein bewusstes Einverständnis und Handeln des Kunden voraus.

Ein weiterer Vertrauensfaktor ist die Herkunft des Kunden-Vertragspartners. In der vorgestellten Lösung kann der Kunde einen lokalen Anbieter wählen und ist nicht gezwungen, einen Vertragspartner aus einem bestimmten Land zu wählen, wie es bei Quasi-Monopolen der Fall ist.

Telekommunikationsunternehmen und Unternehmen mit Zugriff auf die Endgeräte-Identifikation können ihren Kunden eine automatische Authentifizierung zur Vereinfachung des Kaufvorganges anbieten, so wie es heute schon bei Telefonanrufen praktiziert wird. Dies ist umso bedeutender für die Nutzungshäufigkeit, je geringer der Transaktionsbetrag ist.

Die vorgestellte Lösung fördert den Wettbewerb. Der Kunde kann jederzeit seinen Vertragspartner wechseln und behält den Zugriff auf dieselben weltweiten Bezahlangebote. Damit entsteht zum Wohle der Kunden auch bleibender Druck auf die kundenseitigen Transaktionsgebühren.

## Partizipierende Netzbetreiber und Zahlungsdienstleister („Ursprungsnetz“ und „Zielnetz“)

Partizipierende Netzbetreiber oder andere Zahlungsdienstleister können Umsatz und Gewinn gegenüber einer Quasi-Monopol-Lösung signifikant erhöhen:

- Das Ursprungsnetz verdient bei allen Einkäufen seiner Kunden mit und nicht nur bei Käufen von Anbietern, die einen Vertrag mit demselben Zahlungsdienstleister haben.
- Das Zielnetz verdient bei Inhaltsabrufen durch alle Kunden weltweit mit und nicht nur bei Abrufen durch Kunden, die sich bei einem bestimmten Zahlungsdienstleister registriert haben.

Vergleicht man die Nutzerzahlen einzelner Zahlungsdienstleister mit den aggregierten Nutzerzahlen von Telekommunikationsnetzen, lässt sich in der vorgestellten Lösung selbst für die größten existierenden Zahlungsdienstleister eine Steigerung der Transaktionszahlen um mehr als eine Zehnerpotenz erwarten.

Gleichzeitig lassen sich durch weitestgehende Nutzung existierender Abrechnungssysteme die Investitionen zur Bereitstellung der Bezahlösung minimieren und somit von Anfang an geringe Transaktionsgebühren realisieren.

## Anbieter

Die Anbieter müssen keine mühsame und kostspielige Auswahl von Zahlungsanbietern durchführen, mit denen sie die Anzahl der abrechenbaren Kunden kostenbegrenzt maximieren. Stattdessen können Anbieter durch einen Vertrag mit einem Mitglied des Zahlungsnetzwerks alle ans Netzwerk angeschlossenen Kunden – mittelfristig mehrere Milliarden – erreichen.

Die vorgestellte Lösung fördert den Wettbewerb. Der Anbieter kann jederzeit seinen Vertragspartner wechseln und behält den Zugang zu allen abrechenbaren Kunden weltweit. Damit entsteht zum Wohle der Anbieter auch bleibender Wettbewerbsdruck auf die anbieterseitigen Transaktionsgebühren.

Zudem kann der Anbieter einen lokalen Vertragspartner wählen und ist hinsichtlich der Prüfung seines Angebots nicht auf einen Vertragspartner aus einem anderen Kulturräum angewiesen.

Eine Bezahlösung, die von Grund auf global skalierbar, wettbewerbsfreundlich, auf die einfache Nutzung vorhandener Infrastruktur und auf geringe Transaktionsgebühren ausgelegt ist, bringt allen Marktteilnehmern so gewichtige Vorteile, dass sie sich auf Dauer gegenüber Quasi-Monopolen und regionalen Insellösungen durchsetzen wird.

## Literaturverzeichnis

1. O'Mahoney D, Peirce M, Tewari H (2001) Electronic Payment Systems for E-Commerce 2<sup>nd</sup> ed. Artech, Boston
2. Pousttchi K, Schiessler M, Wiedemann DG (2007) Analyzing the Elements of the Business Model for Mobile Payment Service Provision. International Conference on the Management of Mobile Business (ICMB) Toronto
3. Yang D, Wang H, Ren Y, Wang J (2010) Mobile Payment Pattern Based on Multiple Trusted Platforms – China Case. Ninth International Conference on Mobile Business and Ninth Global Mobility Roundtable (ICMB-GMR) Athens
4. Goeller T (2010) What can Online Payment Do for Online Content Sales? Online Payment Overview and Market Needs. IEEE TMC German Chapter Webcast 15 April 2010. [http://ewh.ieee.org/r8/germany/ems/joomla/index.php?option=com\\_docman&task=cat\\_view&gid=28&Itemid=26](http://ewh.ieee.org/r8/germany/ems/joomla/index.php?option=com_docman&task=cat_view&gid=28&Itemid=26). Accessed 19 June 2011
5. Goeke L, Pousttchi K (2010) A scenario-based analysis of mobile payment acceptance. Ninth International Conference on Mobile Business and Ninth Global Mobility Roundtable (ICMB-GMR) Athens
6. International Telecommunication Union (2010) Key Global Telecom Indicators for the World Telecommunication Service Sector. Updated on 21 October 2010. [http://www.itu.int/ITU-D/ict/statistics/at\\_glance/KeyTelecom.html](http://www.itu.int/ITU-D/ict/statistics/at_glance/KeyTelecom.html). Accessed 17 July 2011

7. Müller A, Weyerstraß L (2003) Abrechnungsverfahren zwischen Netzbetreibern sowie zwischen Netzbetreibern und Endkunden. Arbeitskreis Nummernportierung und Netzzusammenschaltung (AKNN). Version 1.0.0 vom 14 Oct 2003. [http://www.aknn.de/fileadmin/uploads/oeffentlich/Dokument\\_Abrechnungsverfahren\\_UAK\\_B\\_1.0.0.pdf](http://www.aknn.de/fileadmin/uploads/oeffentlich/Dokument_Abrechnungsverfahren_UAK_B_1.0.0.pdf) . Accessed 19 June 2011

**Teil VI**

## **Technologische Sicht: Kontext, Navigation & Personalisierung**

---

# Kapitel 19

## Technologische Herausforderungen für kontextsensitive Geschäftsanwendungen

Martin Werner, Moritz Kessel, Florian Gschwandtner, Michael Dürr,  
Kevin Wiesner und Thomas Mair

**Zusammenfassung** Der Einfluss der Informationstechnologie auf die Wirtschaft nimmt ständig zu. Der geschickte Einsatz von Information und Kommunikation kann wesentlich zum Erfolg eines Unternehmens beitragen. Die ständig wachsende Leistungsfähigkeit von mobilen Endgeräten ermöglicht durch den Einsatz von Kontextinformationen wie dem Aufenthaltsort des Endgerätes revolutionäre Anwendungen. In diesem Artikel erläutern wir, wie Geschäftsanwendungen mit Kontextinformationen umgehen können und weshalb sich dennoch der Einsatz von Kontext in Geschäftsanwendungen noch nicht durchgesetzt hat. Darüber hinaus zeigen wir aktuelle Entwicklungen und Trends im Umgang mit mobilen Endgeräten und Kommunikation auf, die sich schließlich zu einem Software-Ökosystem kombinieren lassen, welches die derzeitigen Probleme überwinden kann.

---

Martin Werner  
Ludwig-Maximilians-Universität München,  
E-mail: martin.werner@ifi.lmu.de

Moritz Kessel  
Ludwig-Maximilians-Universität München,  
E-mail: moritz.kessel@ifi.lmu.de

Florian Gschwandtner  
Ludwig-Maximilians-Universität München,  
E-mail: florian.gschwandtner@ifi.lmu.de

Michael Dürr  
Ludwig-Maximilians-Universität München,  
E-mail: michael.duerr@ifi.lmu.de

Kevin Wiesner  
Ludwig-Maximilians-Universität München,  
E-mail: kevin.wiesner@ifi.lmu.de

Thomas Mair  
Ludwig-Maximilians-Universität München,  
E-mail: thomas.mair@ifi.lmu.de

## 19.1 Einführung

Die Informationstechnologie durchläuft in dieser Zeit eine rasante Entwicklung, die das Verhältnis von Menschen und Maschinen revolutioniert. Zu Beginn der Informatik gab es im Wesentlichen einzelne Großrechenanlagen, die von vielen Personen genutzt wurden. Diese Phase wird häufig als Mainframe-Ära bezeichnet. Durch die Einführung des Personal Computers (PCs) veränderte sich die Beziehung zwischen Person und Gerät. Nun verwendete durch die Einführung des PCs nur eine einzige Person einen Computer. Dieses Modell war so erfolgreich, dass mittlerweile fast jeder Haushalt über (mindestens) einen PC verfügt. Derzeit stehen wir vor einem neuen Umbruch im Verhältnis von Person und Computer. Wir sind umgeben von vielen mobilen Computern, die miteinander kommunizieren. Jede Person wird in Zukunft eine ganze Menge an Computern besitzen. Darüber hinaus werden diese auch zunehmend miteinander kommunizieren.

Aber nicht nur das Verhältnis von Computern zu Personen ist in einem dramatischen Umbruch, auch die Ausstattung und Bedienung unterliegt derzeit einer enorm rasanten Entwicklung. War es bis vor wenigen Jahren noch üblich, dass ein Computer über eine Maus und eine Volltastatur gesteuert wird und ein Handy über eine 12-Tasten-Tastatur verfügt, so kann man heute alle Kombinationen in der Realität antreffen: Manche Handys werden nur über einen Touch-Screen bedient, andere besitzen eine bei Bedarf ausfahrbare Voll-Tastatur, einige moderne PCs und insbesondere Software-Entwickler setzen auch auf den Einsatz von Touch-Screen-Technologie zur Steuerung und erste Expertensysteme lassen eine Bedienung per Sprache oder per Gesten zu.

In dieser Zeit ist es auch für die Geschäftswelt von wesentlichem Interesse, wie sie auf diese Entwicklungen reagieren kann beziehungsweise wie sie diese Entwicklung nutzen kann, um sich einen Wettbewerbsvorteil durch die intelligente Nutzung von Informationstechnologie zu erarbeiten oder auszubauen. Alle Anwendungen, welche direkt oder indirekt dem Zweck dienen, einen Wettbewerbsvorteil zu erarbeiten, werden in diesem Text als Geschäftsanwendungen aufgefasst. In diesem Sinne können auch Anwendungen für Kunden Geschäftsanwendungen sein, insbesondere dann, wenn durch diese Anwendungen ein Alleinstellungsmerkmal erarbeitet wird. Eine klare Unterscheidung von Geschäfts- und Privatanwendungen gestaltet sich allerdings schwierig. Der wichtigste Unterschied zwischen Geschäfts- und Privatanwendungen liegt in den Anforderungen, die an Zuverlässigkeit und Korrektheit gestellt werden müssen.

Der Einsatz von Kontextinformation für die Steuerung von Unternehmensabläufen kommt in den letzten Jahren so langsam in Fahrt. Durch die stets sinkenden Kosten für Rechenleistung und Speicherplatz und die Durchdringung des täglichen Lebens durch digitale und vernetzte Geräte besteht die Möglichkeit, aus einer immensen Masse an Informationen Wertschöpfung zu erlangen. So könnte zum Beispiel ohne viel Aufwand der Aufenthaltsort und mit einiger Ungenauigkeit auch die derzeitige Tätigkeit von Mitarbeitern bestimmt werden. Für Informationen dieser Art gibt es selbstverständlich unendlich viele Möglichkeiten des Einsatzes. So kann die telefonische Verfügbarkeit einer Person automatisch anhand solcher Informa-

tionen bestimmt und die Anzahl störender Unterbrechungen verringert werden. Bei Arbeiten, bei denen der Wert einer aufwändigen Dokumentation der Arbeitsschritte die berechtigen Datenschutzinteressen übersteigt, kann und wird ein solches System zur Reduktion von Produktionsfehlern verwendet werden [1].

Natürlich gibt es in dieser noch sehr jungen Forschungsrichtung noch viele Herausforderungen, denen sich die Wissenschaftler aller Disziplinen in den nächsten Jahrzehnten stellen müssen. Aus gesellschaftlicher und juristischer Sicht gibt es schwerwiegende Probleme, die die Verwendung von Kontextinformation in der Geschäftswelt behindern. So ist auf der einen Seite die massive Verwendung von Informationen über das Verhalten von Mitarbeitern sehr wertvoll für eine Firma, und diese Informationen sollten deshalb nach Möglichkeit erfasst werden, um Prozesse zu optimieren. Auf der anderen Seite gefährdet eine umfassende Datensammlung die individuellen Freiheiten der Mitarbeiter, insbesondere das Recht auf informativelle Selbstbestimmung. Darüber hinaus ist ein Missbrauch durch die Erfasser oder Dritte in der Regel nicht auszuschließen. Der jüngste Skandal über Datendiebstahl bei Sony [2] belegt dies eindrücklich.

Ein wesentliches Problem im Hinblick auf die Sammlung von Daten liegt in der Tatsache, dass der Wert der Erfassung solcher Daten wegen der statistischen Unsicherheit erst aus der Retrospektive korrekt bewertet werden kann. Ob also signifikante Verbesserungen und sinnvolle Entscheidungen mit Verfahren des maschinellen Lernens abgeleitet werden können, ist leider nicht im Vorhinein bewertbar und kann daher nicht verwendet werden, um für die Akzeptanz der Sammlung von Daten zu werben.

In diesem Artikel wollen wir uns hauptsächlich auf diesen Bereich konzentrieren und zunächst die Anforderungen an kontextabhängige Prozesse in der Geschäftswelt diskutieren. Danach geben wir einen Überblick über mögliche Quellen für Kontextinformation und über die Modellierung von Kontext. In Abschn. 19.4 diskutieren wir, welche besondere Situation sich daraus ergibt, wenn Kontextinformation mit Hilfe von mobilen Endgeräten gesammelt wird. Im Abschn. 19.5 beschreiben wir grundlegende Methoden, um in einer Menge von Daten einen bestimmten kausalen Zusammenhang zu finden und auf Grund von unstrukturierten Daten Voraussagen und Entscheidungen zu treffen. Abschließend diskutieren wir im Abschn. 19.6 über den Einfluss von neuen Kommunikationsparadigmen auf die Entwicklung mobiler, kontextabhängiger Geschäftsanwendungen.

## 19.2 Kontexterkennung: Spezielle Anforderungen der Geschäftswelt

Betrachtet man die derzeitige Situation, in der Privatkunden mit ihren privaten Mobiltelefonen sehr freigiebig mit persönlichen Daten umgehen, so stellt sich die Frage nach den Gründen, weshalb diese wertvolle Informationsquelle in der Geschäftswelt bisher weitgehend ungenutzt bleibt. Wie bereits in der Einleitung erwähnt gibt es viele Gefahren und Risiken, die zwar im privaten Raum eingegangen werden

können, aber keinesfalls durch eine Firma von den eigenen Mitarbeitern gefordert werden können. Wir wollen in diesem Kapitel die wesentlichen Anforderungen an eine kontextabhängige Unternehmensanwendung zusammenstellen, die – unserer Meinung nach – den größten Einfluss in der Abgrenzung zu Anwendungen für die private Nutzung haben.

### **19.2.1 Kosten**

Die Kosten eines kontextabhängigen Systems sind in der Regel kaum seriös zu beziffern. Der Betrieb der notwendigen Hard- und Software ist hier in der Regel nur ein kleiner Faktor. Die Betreuung der Daten durch Experten im Hinblick auf mögliche Auswertung und Ergebnisse, also die Weiterverarbeitung der gewonnenen Daten zu belastbaren Grundlagen für Entscheidungen, kann beliebig komplex ausfallen. Aber auch die Betreuung der Mitarbeiter im Hinblick auf den Datenschutz, die Kosten für die Erfüllung von Auskunftspflichten über den Bestand der gespeicherten Daten und deren Verwendung, ist selten kalkulierbar. Dieses neue betriebliche Risiko wird sicherlich oft falsch eingeschätzt. Zusammenfassend für die Abgrenzung zwischen privater und geschäftlicher Nutzung ist vielleicht zu sagen, dass die Verwendung von Geschäftsanwendungen in der Regel in einem Unternehmen vorgeschrieben wird und damit im Allgemeinen höhere Kosten für die datenschutzrechtliche Behandlung entstehen, als für ein freiwilliges System, welches wesentlich liberalere Datenschutzklauseln in den Allgemeinen Geschäftsbedingungen durchsetzen kann.

### **19.2.2 Zuverlässigkeit**

Für einen Einsatz in Geschäftsanwendungen ist die zuverlässige Funktionsweise unerlässlich. Systeme müssen die ihnen auferlegte Aufgabe in der geplanten Art und Weise erledigen. Für kontextabhängige Systeme bedeutet dies, dass, falls ein bestimmter Kontext vorliegt, dieser auch erkannt werden soll. Zuverlässigkeit bedeutet hierbei auch, dass solche Systeme zu jeder Zeit und auf jeden bestimmten Kontext in geeigneter Weise reagieren können.

### **19.2.3 Dokumentierbarkeit und Erklärbarkeit**

Falls nun ein kontextabhängiges System eine Empfehlung ausspricht, wie verhält es sich dann mit der Nachvollziehbarkeit und Dokumentierbarkeit von Entscheidungen? Ist immer noch der Entscheidungsträger allein verantwortlich oder ist vielmehr ein Experte, der die Voraussagen mit Daten zu belegen glaubte, in der Verantwor-

tung? Kann und wird ein solches System vielleicht durch Einspielen gefälschter Daten attackiert?

In der Praxis werden gerade wegen dieser Probleme häufig nur sehr einfache Wissensabstraktionen (einfache und prägnante Regeln) verwendet. Diese lassen sich dann dokumentieren, mit Daten belegen und sind trotzdem häufig auch intuitiv verständlich. Allerdings kann dadurch nur selten die volle Leistungsfähigkeit von künstlicher Intelligenz ausgenutzt werden. Hier ist die Berechnung eines finanziellen Risikos einer Entscheidung wünschenswert um eine Abstufung in der Verantwortlichkeit zu ermöglichen.

#### **19.2.4 Akzeptanz**

Vielleicht die wichtigste Anforderung an eine kontextabhängige Geschäftsanwendung liegt in ihrer Akzeptanz. Um diese zu erreichen, muss sehr genau zwischen den Interessen der Mitarbeiter und denen der Firma abgewogen werden.

Aus Sicht des Mitarbeiters hängt die Akzeptanz einer Geschäftsanwendung zum einen von der Art der verwendeten Kontextinformationen, zum anderen von der Einfachheit der Kommunikation, also von der Transparenz der zugrunde liegenden Kommunikationsmedien ab. Aus Sicht der Firma selbst hängt die Akzeptanz hingegen im Wesentlichen von der Integrierbarkeit, der Zuverlässigkeit sowie der Sicherheit der Kontextinformationen und der verwendeten Kommunikationsmedien ab. Gerade im Hinblick auf die Integration neuartiger sozialer Kommunikationsmedien muss daher streng darauf geachtet werden, dass sowohl die Interessen der Mitarbeiter, als auch die Interessen der Firma bei der Umsetzung einer kontextabhängigen Geschäftsanwendung gewahrt bleiben.

### **19.3 Modellierung von Kontext**

Wenn über kontext- und situationsabhängige Software geschrieben wird, besteht meist eine enorme Unsicherheit in der Frage, was Kontext eigentlich ist. Die wohl anerkannteste Definition, die wir für unseren Beitrag verwenden wollen, stammt von Dey und definiert Kontext (in freier Übersetzung) wie folgt [3]:

„Kontext ist jegliche Information, die geeignet ist, die Situation einer Entität zu charakterisieren. Eine Entität ist eine Person, ein Ort oder ein Objekt, das als relevant für die Interaktion zwischen Nutzer und Applikation – inklusive Nutzer und Applikation selbst – betrachtet wird.“

Üblicherweise besteht der tatsächlich vorliegende Kontext aus Sensordaten über Ort, Zeit und direkte Umgebung sowie aus Daten des Terminkalenders oder aus Daten von Geschäftsanwendungen wie Customer-Relationship-Managementsystemen.

Eine zentrale Eigenschaft von Kontextinformationen ist die örtlich sowie zeitlich eingeschränkte Bedeutung eines einzelnen Kontextwertes. Mit zeitlicher sowie

räumlicher Entfernung von einer tatsächlichen Kontextmessung nimmt die Relevanz des Wertes natürlich ab.

Darüber hinaus spielt die Qualität der Sensoren gerade bei der Unterstützung von Geschäftsprozessen eine große Rolle. Um Aussagen über die Verlässlichkeit eines kontext-sensitiven Systems zu treffen, muss die Verlässlichkeit jeder Kontextinformation bekannt und besser sein als eine globale Mindestanforderung.

Naheliegende Beispiele für Kontextquellen sind Uhren, GPS-Empfänger, Systeme zur Indoor-Positionierung, Beschleunigungssensoren, Mikrofone, Thermometer, Barometer und ähnliche Sensoren, mit denen der Zustand der individuellen Umgebung beobachtet werden kann.

Darüber hinaus bieten noch Werkzeuge zur Zeitplanung und Kommunikation sinnvolle Kontextquellen. So kann beispielsweise der Kalender Aufschluss über Tätigkeiten, Termine und Aufgaben geben oder das E-Mail-System Kommunikationswege und Bekanntschaften offenlegen.

Gerade das Ableiten bestimmter Kontext-Informationen aus Sensorik, das Erzeugen sogenannter höherwertiger Kontext-Informationen, spielt eine wichtige Rolle zur Entscheidungsfindung. Da die Verarbeitung von primärem Kontext genauso wie die Verwaltung von abgeleitetem sekundären Kontext für den Erfolg einer Anwendung entscheidenden Einfluss haben kann, hat die Forschung bereits seit einiger Zeit versucht, Kontext-Informationen so zu modellieren, dass eine gewisse Allgemeingültigkeit und Flexibilität entsteht und so Algorithmen und Systeme wieder verwendbar werden.

Im Folgenden werden zunächst einige Aspekte der Modellierung formuliert und anschließend die gängigsten Modelle erläutert [4]:

Einer der wichtigsten Aspekte ist die Modellierung von Datenheterogenität. So unterliegen z. B. Sensoren bei Messungen immer auch Ungenauigkeiten, die einen erheblichen Einfluss bei der Interpretation der Werte haben können. Im Vergleich zu Sensoren liefern andere Kontext-Quellen wie die oben erwähnte Zeitplanung präzise Informationen, die auch nur geringer oder gar keiner Interpretation bedürfen. Aus diesem Grund muss ein Kontext-Modell in der Lage sein, den verschiedensten Arten von Kontext-Quellen zu genügen und auf deren Stärken und Schwächen einzugehen. Zusätzlich zur Heterogenität ist es durch die Mobilität von Sensoren und des Nutzers notwendig, dass das Modell in der Lage ist, auch mit wechselnden Kontext-Quellen und fehlenden Werten beliebiger Kontextquellen umzugehen.

Ein weiterer wichtiger Aspekt ist die Modellierung von Abhängigkeiten zwischen unterschiedlichen Arten von Kontext-Informationen. So kann die Gesamtheit der Messwerte unterschiedlicher Quellen eine höherwertige Information liefern als jede Quelle einzeln. Gerade bei Änderungen an bestimmten Messwerten muss die Gültigkeit der höherwertigen Information überprüft werden. Diese höherwertige Information muss als Entscheidungsgrundlage genutzt werden können und damit neben ihrem tatsächlichen Wert auch einen Indikator für Verlässlichkeit haben.

Um eine sinnvolle und richtige Nutzung des Modells zu gewährleisten ist es zudem wichtig, dass eine einfache Abbildung der Realität in das Kontext-Modell möglich ist. Dadurch wird die Bereitstellung kontextabhängiger Anwendungen erleicht-

tert und die Interpretation der Kontext-Informationen durch Anwendungen verbessert.

In der Forschung auf dem Gebiet haben sich im Laufe der Zeit unterschiedliche Ansätze etabliert, wie Kontext modelliert werden kann. Neben ursprünglichen Modellierungsverfahren, die auf Schlüssel-Wert-Paaren basieren oder Markup-Sprachen verwenden [5], wird immer stärker auch auf strukturierte Verfahren wie Objekt-Rollen-basierte Modelle, räumliche Modelle und Ontologie-basierte Modelle umgeschwenkt [4, 5]. Diese eignen sich auch insbesondere, um spezielles Domänenwissen in Verfahren des maschinellen Lernens (vgl. Abschn. 19.5) zu integrieren.

Object Role Modeling (ORM) beschreibt ein Verfahren, um das Design von Datenbanken und die Abfragen darauf bereits auf konzeptioneller Ebene zu planen und durchzuführen. Aufbauend auf diesen Verfahren kann auch ein Modell für Kontext-Informationen angelegt werden. Analog zu den oben genannten Anforderungen muss dabei jedoch das Modell erweitert werden, um zum Beispiel Werte mit einer gewissen Unsicherheit beschreiben zu können oder um die Spezifikation von komplexen Bedingungen zu ermöglichen. Ist dies erreicht, so besticht ORM besonders mit der Möglichkeit, durch eine grafische Notation das Design und das Verständnis stark zu vereinfachen.

Ein Nachteil von ORM ist jedoch die flache Hierarchie der Kontext-Arten. Diese wirkt sich besonders dann aus, wenn eine Kontext-Art besonders bei der Verarbeitung hervorsticht. Handelt es sich hierbei zum Beispiel um den Kontext Ort, welcher heutzutage sehr häufig verwendet wird, so bietet sich der Gebrauch von räumlichen Modellen an. Diese verwenden den physikalischen Ort zur Organisation der Kontext-Informationen. Der Ort kann dabei die Position eines Objektes darstellen, die durch den Kontext beschrieben wird, oder aber die Position eines Sensors darstellen, der zur Kontext-Generierung verwendet wird. Darüber hinaus können virtuelle Objekte mit einer Position versehen werden und so einen örtlichen Kontext erhalten. Der Ort kann dabei vordefiniert sein oder aus irgendwelchen anderen Informationen abgeleitet werden. In den in diesem Artikel beschriebenen mobilen Verfahren ist es jedoch wahrscheinlicher, dass der Ort über ein beliebiges Positionierungssystem bestimmt wird und dass die sonstigen Messwerte und Kontextdaten sich direkt auf diesen Ort beziehen. Räumliche Modelle erlauben es, Anfragen über den Ort und über räumliche Beziehungen effizient auszuführen. Besonders verbreitet sind hier Anfragen nach den nächstgelegenen Nachbarn. Doch leider tendieren räumliche Modelle dazu, den Ort in Bezug auf nicht ortsabhängige, höherwertige Kontextinformation überzubewerten. So können Zusammenhänge zwischen Orten und Zuständen hergestellt werden, die nur zufällig entstehen.

Ontologie-basierte Modelle beruhen auf der Ausdruckskraft logischer Sprachen und der Unterstützung von automatisierten Schlussfolgerungen in Beschreibungslogiken. Dabei können durch die Logik auch komplexe Kontextdaten einfach modelliert werden und verschiedenartige Kontext-Informationen aus unterschiedlichen Quellen anhand einer formalen Semantik einbezogen werden. Ein weiterer Vorteil von Ontologie-basierten Modellen ist die einfache Modellierung von Abhängigkeiten.

ten. Die Abhängigkeiten zwischen verschiedenen Messwerten können anhand von Verfahren der Knowledge-Discovery (Abschn. 19.5) aufbauend auf der Beschreibungslogik automatisch in höherwertige Kontext-Informationen überführt werden. Ebenso lässt sich die Konsistenz von Kontext-Informationen automatisch überprüfen. Ontologie-basierte Modelle eignen sich also vor allem bei einer großen Heterogenität von Kontext-Quellen und hohen Anforderungen an Interoperabilität zwischen verschiedenen Endgeräten, wie sie gerade im geschäftlichen Umfeld bestehen. Allerdings müssen noch Mechanismen erforscht werden, welche die zeitliche und örtliche Relevanz von Kontextdaten in Ontologie-basierten Modellen flexibel hervorheben können.

## 19.4 Mobiles, nutzerzentrisches Sensing

Da mobile, kontextabhängige Geschäftsanwendungen in der Regel vorhandene Smartphone-Infrastrukturen verwenden sollen, wollen wir in diesem Kapitel ein wenig auf die speziellen Eigenschaften von Kontextdaten eingehen, wenn diese verteilt mit den Mobiltelefonen der Nutzer gesammelt werden. Dabei kann man die Menge an Mobiltelefonen, die für eine kontextabhängige Anwendung kooperieren, als eine spezielle Art Sensornetz begreifen und Verfahren aus diesem Bereich anwenden.

Eine Erfassung von Sensordaten mit Hilfe mobiler Endgeräte wird oft als mobiles, nutzerzentrisches Sensing („People-Centric Sensing“ [6], „Mobile Phone Sensing“ [7]) bezeichnet. Dieser Forschungsbereich ist dem der kabellosen Sensornetze, sog. „Wireless Sensor Networks“ (WSN) sehr nahe, bei den (meist stationären) Sensorknoten über funkbasierte Kommunikation gemessene Sensordaten austauschen und verbreiten. Jedoch gibt es einige wichtige Unterschiede von mobilen Sensornetzwerken (MSN) im Vergleich zu WSNs [8]:

- Energie: Im Gegensatz zu stationären Sensorknoten kann bei mobilen Endgeräten, die ebenfalls sehr begrenzte Energiekapazitäten besitzen, davon ausgegangen werden, dass diese regelmäßig vom Nutzer wieder aufgeladen werden.
- Verbreitung: Die großflächige Verbreitung von Mobiltelefonen bietet auf der einen Seite eine große Menge von Sensorknoten, diese messen allerdings auch nur in den von Menschen besuchten Gegenden.
- Sensordiversität: Durch die große Anzahl an verschiedenen Endgeräten gibt es typischerweise auch eine Vielzahl verschiedener Sensoren, wohingegen WSNs meist aus einer eher homogenen Sensormenge bestehen.
- Privatsphäre: Da Mobiltelefone von ihren Nutzern mitgeführt werden, können diese auch viel über die Aktivitäten des Nutzers verraten, was einen effektiven Schutz der Privatsphäre nötig macht.

MSNs und darauf aufbauende Anwendungen kann man auf der einen Seite nach Nutzungsfokus kategorisieren, also ob die Messungen und Daten für einzelne Per-

sonen, Gruppen oder eine Gemeinschaft gedacht sind [7], auf der anderen Seite auch nach der Art und Weise, wie die Nutzer die Daten beitragen, hierbei wird zwischen aktiv und passiv (opportunistisch) teilnehmenden Nutzern unterschieden [9].

In der Forschung gibt es bereits einige Arbeiten und Projekte [10–13], die sich mit dieser Thematik beschäftigen. Als typische Einsatzszenarien werden neben der Messung von Umweltdaten wie beispielsweise Luftverschmutzung oder Temperatur auch die Bestimmung der Umgebungslautstärke oder Straßengegebenheiten genannt. Gerade im Geschäftsumfeld bietet diese neue Art, Daten und Informationen zu erhalten, viele neue Möglichkeiten, da kostengünstig großflächige und aktuelle Daten geliefert werden können. In industriellen Anlagen und Fabriken könnten diese auf Mobiltelefonen basierenden Sensorknoten zur Überwachung der Sicherheit und Gesundheit der Arbeiter eingesetzt werden. So könnte beispielsweise auf den Austritt gefährlicher Gase oder auf eine dauerhafte Schadstoffbelastung umgehend reagiert werden und falls nötig automatisch in die Steuerung von Prozessen und Maschinen eingegriffen werden. Weitere Einsatzszenarien sind die Nutzung in Krankenhäusern oder als Echtzeit-Informationsplattform in Städten zur Optimierung des öffentlichen Transports und des Energiemanagements.

Die größten Herausforderungen um mobiles, nutzerzentrisches Sensing im geschäftlichen oder industriellen Umfeld zu nutzen, liegen in der Sicherstellung und Bewertung von Zuverlässigkeit (vgl. Abschn. 19.2.2) und Qualität. Die Qualität der Daten wird zum einen durch die in der Regel relativ geringe Qualität der eingebauten Hardware, zum andern auch durch die Heterogenität der Endgeräte negativ beeinflusst. Eine wichtige Herausforderung besteht daher in der automatischen und verteilten Kalibrierung mobiler Sensoren, um auf diese Weise die Qualität der Daten zu erhöhen und gleichzeitig eine Möglichkeit für die Qualitätsbewertung des Gesamtdienstes zu erhalten. Es gibt bereits erste Ansätze zur Selbstkalibrierung [14], allerdings in der Regel ohne eine Bewertungsmöglichkeit der Qualität von gemessenen Werten. Dies ist jedoch essentiell notwendig, um die Vertrauenswürdigkeit der Information einzuschätzen. Auch die Lage des mobilen Endgeräts ist entscheidend bei der Erfassung von Daten. Mobiltelefone werden beispielsweise die meiste Zeit in Taschen getragen und können dort nur bedingt sinnvolle Messungen durchführen. Erste Ansätze versuchen durch Bestimmung der Lage des Geräts [15] oder durch Mehrfachmessungen [8] damit umzugehen. Ein weiteres Problem für die Zuverlässigkeit besteht darin, dass durch die dynamische Verteilung eine gleichmäßige und vollständige Abdeckung eines Einsatzgebietes nicht erreicht wird. Zudem können mobile Knoten durch Ausschalten, ob manuell oder aufgrund eines leeren Akkus, auch temporär ausfallen.

Ein weiterer, entscheidender Aspekt liegt in der Tatsache, dass die nutzerzentrisch erhobenen Daten ganz klar als persönliche Daten einzuschätzen sind und daher Mechanismen zur Anwendung kommen sollten, die eine Abwägung zwischen dem berechtigten Schutz der Privatsphäre [8, 11] und der Nachweisbarkeit und Lokalisierbarkeit von Fehlfunktionen und Angriffen ermöglichen.

## 19.5 Knowledge Discovery im geschäftlichen Einsatz

Wie kann jetzt aber eine kontextabhängige Anwendung die Masse an Rohdaten – seien es Rohdaten, die von vielen Nutzern opportunistisch beigetragen werden oder seien es längere Messreihen in fest definierten Umgebungen – verwenden, um aus neu ankommenden Messungen sinnvolle Schlüsse zu ziehen? Das Erkennen von Mustern in größeren Datenbanken ist ein großes und altes Forschungsfeld. Im Wesentlichen wird hier eine algorithmische Abstraktion der Begriffe „Information“, „Wissen“, „Lernen“ und „Begründen“ definiert und mit Leben gefüllt.

Im Folgenden wollen wir die grundlegenden Strukturen erläutern, mit denen aus größeren Datenmengen ein Modell abgeleitet wird, welches zur Vorhersage zukünftiger Ereignisse geeignet ist. Hierzu werden die vorhandenen Daten in der Regel durch Vorbereitungsalgorithmen, häufig unter Verwendung von speziellem Domänenwissen, in eine Menge von Attributen organisiert. Attribute können binär sein, zum Beispiel ob ein bestimmter Kunde eine bestimmte Leistung in Anspruch genommen hat, oder numerisch, so wie der Stand des Zeitkontos eines Mitarbeiters. Darüber hinaus gibt es noch die Möglichkeit, diskrete mehrwertige Attribute einzuführen. So kann zum Beispiel das Wetter in die Klassen „sonnig“, „bewölkt“ und „regnerisch“ unterteilt werden. Derzeitige industrielle Anwendungen umfassen häufig mehrere tausend verschiedene Attribute. So könnte zum Beispiel ein System zum Börsenhandel die Kurse aller Aktien an allen deutschen Märkten und deren aktuelle Handelsvolumina umfassen.

Sobald alle Kontextinformationen als Attribute in einer einheitlichen Form dargestellt werden können, benötigen wir eine Möglichkeit, unterschiedliche Datensätze zu speichern. Hierbei ist eine Instanz eine Zusammenstellung der Werte aller Attribute zu einem bestimmten Zeitpunkt. Instanzen müssen dabei nicht unbedingt vollständig sein, einzelne Werte können auch unspezifiziert bleiben. Darüber hinaus müssen nicht immer alle verfügbaren Instanzen verwendet werden. Man kann auch nur die Instanzen in einem bestimmten Zeitraum verwenden oder insbesondere um Trends zu erkennen, eine Vorhersage auf Grund aller Instanzen bis zum aktuellen Zeitpunkt mit der Vorhersage auf Grund aller Instanzen bis zu einem (vergangenen) Referenzzeitpunkt vergleichen.

Die grundlegende Vorgehensweise zum Erkennen von Mustern in großen Datensätzen fängt damit an, dass man einen Satz von Trainingsdaten erstellt. Trainingsdaten haben die Eigenschaft, dass für die einzelnen Datensätze das Ergebnis welches man später für neue Datensätze voraussagen will, schon bekannt ist. Aus den Trainingsdaten wird mit sogenannten Lernalgorithmen ein Modell trainiert, welches die Muster in den Trainingsdaten analysiert und die Parameter des Modells an die Vorgaben der Trainingsdaten anpasst. Das Training des Modells findet normalerweise in einer Offline-Phase statt, da es sich meist um zeitaufwändige Algorithmen handelt. Das durch einen Lernalgorithmus erstellte Modell wird anschließend für die Voraussage von Datensätzen mit unbekannten Ergebnissen verwendet. In manchen Situationen ist es auch sinnvoll, das Modell nachträglich mit neuen Daten anzureichern. Dies ermöglichen sogenannte Online-Algorithmen. Sie integrieren neue Trainingsdatensätze in das Modell ohne auf die alten Trainingsdaten zurückzugreifen.

In der Anwendungsphase werden für Datensätze, für welche kein Ergebnis bekannt ist, mit Hilfe des trainierten Modells Voraussagen getroffen.

Die Erstellung eines Modells zur Vorhersage von Ergebnissen mithilfe der Trainingsdaten und die Auswahl eines geeigneten Lernalgorithmus ist stark von der Struktur der Trainingsdaten und der vorauszusagenden Ergebnissen abhängig. So gibt es beispielsweise Modelle, welche nur diskrete Klassen vorhersagen können, oder spezielle Modelle für numerische Voraussagen. Um ein bestimmtes Modell zu trainieren, benötigt man auch einen speziell auf das Modell angepassten Lernalgorithmus. Je nach Art der vorhandenen Trainingsdaten gibt es drei generelle Arten von Lernalgorithmen. Existiert für jeden Trainingsdatensatz ein Ergebnis, so verwendet man Methoden des Supervised-Learning. Dabei optimiert man das Modell darauf, für so viele Datensätze wie möglich die in den Trainingsdaten vorhandenen Ergebnisse richtig vorauszusagen. Manchmal besteht auch die Notwendigkeit, aus Daten, zu denen keine Ergebnisse vorhanden sind, Korrelationen zu extrahieren. Dies nennt man Unsupervised-Learning. Oft sind aber auch nur für manche Trainingsdatensätze Ergebnisse bekannt und für andere wiederum nicht. Für diese Situation gibt es die Verfahren des Semisupervised-Learning, welche die Daten mit Ergebnissen geschickt mit denen ohne Ergebnisse kombinieren und dadurch bessere Ergebnisse erzielen, als wenn nur die Daten mit bekanntem Ergebnis verwendet worden wären [16].

Wichtig bei allen Methoden zur Erstellung eines Modells ist es, darauf zu achten, dass das Modell nicht zu sehr an die Daten angepasst wird (sogenanntes Overfitting). Dabei trifft das Modell für die vorhandenen Trainingsdaten optimale Voraussagen. Allerdings beruhen diese zu einem großen Teil nur auf speziellen, oft zufälligen Eigenschaften der Trainingsdaten und nicht auf einer inhaltlichen Grundlage. Voraussagen für neue, nicht im Trainingsdatensatz enthaltene Daten sind dann oft falsch. Zur Vermeidung dieses Phänomens werden die Modelle am Ende der Lernphase analysiert und vereinfacht. Meist besteht diese Phase darin, jene Teile des Modells zu identifizieren, welche wenig zur Ergebnisfindung beitragen, und sie aus dem Modell zu entfernen. Insbesondere die Analyse des tatsächlichen Informationsgewinns, also im Wesentlichen die Frage, wie viel Information mit einer Regel subsumiert wurde, ist an dieser Stelle eine geeignete und viel verwendete Technik.

Im Folgenden wollen wir uns nun den verschiedenen Modellen, welche die Lernalgorithmen aus den Trainingsdaten erzeugen, zuwenden. Jedes Modell erfordert einen speziell auf das Modell angepassten Lernalgorithmus. Ein einfaches, aber sehr verständliches Modell sind Entscheidungsbäume. Entscheidungsbäume sind hierarchische Strukturen, welche von oben nach unten durchlaufen werden. Dabei wird an jedem Knoten eine Entscheidung basierend auf einem Attribut eines Datensatzes getroffen. Der wesentliche Vorteil von Entscheidungsbäumen liegt darin, dass die Entscheidungsfindung verständlich und leicht nachvollziehbar ist. Entscheidungsbäume werden nicht nur von Lernalgorithmen berechnet, sondern zum Beispiel auch von Experten zur Diagnose von Krankheiten von Hand erstellt. Ein weiteres Modell sind neuronale Netze. In neuronalen Netzen werden Zusammenhänge zwischen Eingabe- und Ausgabeattributen in einer gewichteten Graph-Struktur modelliert. Die Gewichte werden dabei durch Erfahrung (d. h. Trainingsdaten) angepasst. Ge-

nau wie im menschlichen Gehirn werden Verbindungen zwischen Knoten gestärkt, wenn passende Trainingsdaten vorliegen und geschwächt, wenn dies nicht der Fall ist. Durch geschicktes Training mit Beispielen entsteht so ein System, welches korrekte Entscheidungen treffen kann. Leider sind die dabei entstehenden Strukturen im Gegensatz zu Entscheidungsbäumen kaum verständlich, da sie nicht von einzelnen Attributen abgeleitet wurden, sondern sich aus der Gesamtheit der Daten herab Bilden. Ein weiteres Modell aus dem Bereich der Statistik sind Bayes'sche Netze. Sie beschreiben kausale Zusammenhänge zwischen bedingten Wahrscheinlichkeiten von Zufallsvariablen als gerichtete Graphen. Bayes'sche Netze eignen sich sehr gut um Unsicherheit zu modellieren. Häufig sind aber auch fast triviale Modelle viel erfolgreicher. So kann gelegentlich das Ergebnis auch einfach als gewichtete Linearkombination der Attribute vorhergesagt werden.

Es gibt viele Belege für den Erfolg von maschinellem Lernen in Geschäftsanwendungen [17]. In dieser Arbeit führt Langley etwa die folgenden Anwendungen im geschäftlichen Umfeld auf:

In einem frühen Beispiel hat American Express die Vergabe von Krediten an Personen mit Methoden des maschinellen Lernens verbessert [18]. In einer eher überraschender Anwendung verwendete der industrielle Versicherer „Hartford Steam Boiler“ eine Menge an automatisch erzeugten Regeln, um Ausfälle von ölfüllten elektrischen Transformatoren vorherzusagen. Dazu wurden allein gaschromatografische Untersuchungen des Öls der Transformatoren verwendet [19]. In einer aktuellen Anwendung aus dem letzten Jahr wird ein Teil eines industriellen Fertigungsprozesses allein mit Positionssensoren mit sehr großem Erfolg überwacht [1].

## 19.6 Kontextabhängige Kommunikationsparadigmen

Gerade im Bezug auf mobile Geschäftsanwendungen stellen sich in Zukunft ganz neue Herausforderungen an die Kommunikation. Dabei spielt insbesondere das Wissen um den Nutzerkontext eine wesentliche Rolle.

Betrachtet man die gegenwärtige Entwicklung im privaten Umfeld, so erkennt man einen klaren Trend, weg von den herkömmlichen Kommunikationskanälen wie SMS, Chat oder E-Mail, hin zur Verwendung sozialer Medien. Getrieben von den Entwicklungen rund um das Web 2.0 bieten soziale Medien ganz neue Möglichkeiten der Kommunikation, Kollaboration und des Wissensmanagements über multimediale Inhalte und vereinfachen damit die alltägliche Korrespondenz enorm. Im privaten Umfeld werden vor allem Webblogs, Microblogging, Instant Messaging und Foren als Technologie für soziale Medien eingesetzt. Eine Sonderstellung nehmen hier soziale Netze ein, deren Nutzung zuletzt ein nahezu exponentielles Anwachsen verzeichnete. Vertreter sozialer Netze wie Facebook, Twitter, MocoSpace, Google Latitude, Foursquare oder Flickr erlauben das Versenden privater oder öffentlicher Nachrichten, das Veröffentlichen persönlicher Kontextinformationen wie z. B. den gegenwärtigen Aufenthaltsort, das Publizieren und Teilen von multimedia-

len Inhalten wie z. B. Bildern oder Videos sowie das Kommentieren, Bewerten und Empfehlen solcher Inhalte.

Die enorm wachsende Zahl von Smartphone-Nutzern sowie die Verfügbarkeit entsprechender Anwendungen zur mobilen Nutzung sozialer Medien verstärkt diesen Trend.

Offensichtlich bietet die Verwendung mobiler sozialer Netze zum Austausch von Nachrichten, Kommentaren, Meinungen, Inhalten und allen erdenklichen Kontextinformationen einen enormen Mehrwert für die Nutzer, weswegen sich der Ruf nach sozialen Netzen in der Geschäftswelt nahezu aufdrängt.

Bisher hat eine Integration bzw. Adaption der Kommunikationsmöglichkeiten sozialer Medien in der Geschäftswelt jedoch kaum stattgefunden. So reduziert sich deren Einsatz primär darauf, ein Unternehmen und seine Mitarbeiter nach außen hin zu repräsentieren. Im Vordergrund stehen hier Online Marketing und Vertrieb [20] oder die Personalanalyse [21]. Zum Teil lässt sich dies auf eine mangelnde Nutzerakzeptanz zurückführen, die mit steigendem Alter der Mitarbeiter abnimmt [22]. So beschränkt sich die interne Nutzung sozialer Medien oft darauf, dass Mitarbeiter ihre Nutzerkonten dazu verwenden, ihre Kontaktdaten und Adressbücher komfortabel zu pflegen [23]. Zudem existieren in den meisten Unternehmen berechtigte Sicherheitsbedenken gegenüber dem Einsatz solcher Dienste: Zum einen besteht die Angst, dass Profilinformationen über Mitarbeiter oder firmeninterne Geheimnisse in den Besitz nicht autorisierter Dritter gelangen könnten. Zum anderen drohen Gefahren durch die Ausnutzung fehlerhafter Softwarekomponenten, die Attacken wie Cross-Site Scripting oder SQL-Injection ermöglichen.

Die Notwendigkeit zur Einbettung sozialer Medien in derzeitige Unternehmensstrukturen und der damit einhergehende Bedarf zur Integration mobiler Endgeräte verlangt nach passenden Lösungen und Konzepten.

Um von den positiven Effekten sozialer Medien in Bezug auf Kommunikation, Kollaboration und Wissensmanagement zu profitieren, müssen einige Anforderungen in Betracht gezogen werden, die sich für eine geschäftsorientierte Adaption bzw. Integration sozialer Medien ergeben.

Einige dieser Anforderungen wurden zum Teil auch schon früher vorgeschlagen, um soziale Netze im privaten Umfeld sicherer zu machen und die Privatsphäre seiner Nutzer zu stärken [24].

Einen wesentlichen Gesichtspunkt bei der Integration bzw. Adaption sozialer Netze in das Geschäftsumfeld stellt die Zuverlässigkeit eines verwendeten Kommunikationskanals dar. Hier dominieren derzeit asynchrone pull-basierte Verfahren. Um die Anforderungen moderner Anwendungen im Hinblick auf Zuverlässigkeit und Geschwindigkeit zu erfüllen, bedarf es der Aggregation der unterschiedlichsten Kommunikationskanäle wie SMS, E-Mail, Chat oder Twitter.

Die stark divergierende Verwendung herkömmlicher und sozialer Medien zur Kommunikation bewirkt schon heute eine starke Segmentierung in der Art der Informationsverbreitung. Diese Divergenz begründet sich primär in der unzureichenden Akzeptanz neuer Medien bei älteren Mitarbeitern. Eine wesentliche Anforderung an die Integration neuer Kommunikationskanäle besteht im Erreichen einer umfassenden Transparenz.

Die Bereitstellung einer medienübergreifenden Transparenz des Kommunikationsmodells impliziert die Notwendigkeit einer generischen Integrierbarkeit neuartiger Technologien. Um neue Medien nachträglich zu integrieren, sollte die Entwicklung mobiler Geschäftsanwendungen möglichst keine Abhängigkeiten zu einem bestimmten Kommunikationsmodell aufweisen.

Gerade im Bezug auf die Verwendung sozialer Medien hat sich in jüngster Vergangenheit immer wieder gezeigt, dass sich ausreichende Datensicherheit hier nur schwer garantieren lässt. Die Risiken drohen insbesondere durch fehlerhaft implementierte Benutzerschnittstellen, aber auch durch unverantwortliches Verhalten der Mitarbeiter selbst. Im Hinblick auf die Einbindung mobiler Geschäftsanwendungen werden diese Risiken zusätzlich verstärkt, da sowohl der Wert der Information gezielte Hackerangriffe wahrscheinlicher macht und kaum Sicherungsmechanismen im mobilen Umfeld umgesetzt werden können. Auch in diesem Rahmen werden in Zukunft Kontextinformationen wie Benutzeridentität, Aufenthaltsort, Konnektivität oder Geheimhaltungsrichtlinien stark dazu beitragen, die Sicherheit der Kommunikation mobiler Geschäftsanwendungen zu erhöhen.

In Zukunft wird die Integration mobiler Geschäftsanwendungen das Kommunikationsverhalten im Unternehmensumfeld stark beeinflussen. Dabei spielt insbesondere die Einbindung sozialer Medien für die interne Nutzung im Unternehmensumfeld eine besondere Rolle. Um die Nachhaltigkeit eines Konzepts zur Integration sozialer Medien in mobilen Geschäftsanwendungen zu gewährleisten, müssen zukünftige Lösungen sich stark an den eben skizzierten Anforderungen orientieren. Zum einen stehen hier die kontextsensitive Aggregation, transparente Bereitstellung und nachträgliche Integrierbarkeit herkömmlicher und neuartiger Kommunikationskanäle im Vordergrund. Zum anderen liegt der Fokus auf einer kontextbestimmten Absicherung der Kommunikation. Dies beinhaltet sowohl die Gewährleistung der Datensicherheit, als auch die Einhaltung der Richtlinien zur informationellen Selbstbestimmung der Mitarbeiter.

## 19.7 Fazit

Um die Leistungsfähigkeit von kontextabhängigen, mobilen Anwendungen für die Wirtschaft zugänglich zu machen, sind einige Hürden zu überwinden. Zunächst müssen die rechtlichen, sozialen und persönlichen Bedenken ausgeräumt werden. Dies kann zum Beispiel durch Pilotprojekte und eine ergebnisoffene Diskussion in Feldversuchen unter Beteiligung möglichst repräsentativer Anwendergruppen erreicht werden. Denn wenn es gelingt, Mitarbeiter von den Vorteilen der Verwendung von Kontextinformation zu überzeugen und den Arbeitsalltag nachhaltig zu vereinfachen, ist auch die Skepsis in Bezug auf die Verwendung der Daten nicht mehr groß. Ist allerdings der Sinn und Zweck der Sammlung irgendwelcher Daten den Mitarbeitern nicht klar, so entsteht Angst und Ablehnung. Die informationstechnologischen Grundlagen für kontextabhängige Anwendungen sind mittlerweile relativ gut erforscht und können ohne großen Aufwand im Tagesgeschäft einge-

setzt werden. Eine völlig neue Dimension von Kontextinformation, die sicherlich auch zu neuen Verfahren im gesamten Forschungsgebiet führen wird, besteht in der Zunahme sozialer Informationen in den Datenbeständen. Je mehr die Mitarbeiter über soziale Netzwerke kommunizieren, desto mehr Information lässt sich einer bestimmten sozialen Gruppe zuordnen. So wird es erstmals möglich, nicht auf eine Person zu personalisieren, sondern auf eine soziale Umgebung, d. h. eine Person mitsamt ihren engsten geschäftlichen Kontakten. Für die Erklärbarkeit von kontext-abhängig getroffenen Entscheidungen liegt hierin eine große Chance, weil nicht die speziellen Wünsche und Vorlieben einer einzelnen Person, sondern tatsächlich die Anforderungen eines Haupttätigkeitsbereiches durch kontextabhängige Applikationen unterstützt werden.

Zusammenfassend wollen wir festhalten, dass für den Einsatz von kontextabhängigen Systemen im Geschäftsumfeld und insbesondere für die Verwendung von mobilem Kontext aus Smartphones die wesentlichen Probleme in den Bereichen der Nutzerakzeptanz, des Datenschutzes und der Verlässlichkeit liegen. Hier werden wir eine spannende Entwicklung sehen und vielleicht schon sehr bald flächendeckend kontextabhängige Systeme verwenden.

## Literaturverzeichnis

1. Zinnen A, Wojek C, Schiele B (2009) Multi activity recognition based on bodymodel-derived primitives. *Location and Context Awareness*, pp 1–18
2. Bilton N (2011) Hackers claim to have playstation users card data. <http://bits.blogs.nytimes.com/2011/04/28/hackers-claim-to-have-playstation-users-card-data/> 25. Oktober 2011
3. Dey A, Abowd G (2000) Towards a better understanding of context and context-awareness. In: CHI 2000 workshop on the what, who, where, when, and how of context-awareness, vol 4, pp 1–6
4. Bettini C, Brdiczka O, Henricksen K, Indulska J, Nicklas D, Ranganathan A, Riboni D (2010) A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing* 6:161–180, doi: 10.1016/j.pmcj.2009.06.002
5. Strang T, Linnhoff-Popien C (2004) A context modeling survey. In: International Workshop on Ubiquitous Computing, Nottingham
6. Campbell AT, Eisenman SB, Lane ND, Miluzzo E, Peterson RA, Lu H, Zheng X, Musolesi M, Fodor K, Ahn GS (2008) The rise of people-centric sensing. *IEEE Internet Computing* 12:12–21, doi: <http://doi.ieee.org/10.1109/MIC.2008.90> 25. Oktober 2011
7. Lane N, Miluzzo E, Lu H, Peebles D, Choudhury T, Campbell A (2010) A survey of mobile phone sensing. *Communications Magazine, IEEE* 48(9):140–150
8. Wiesner K, Dürr M, Duchon M (2011) Private Pooling: A Privacy-Preserving Approach for Mobile Collaborative Sensing. In: Proceedings of 3rd International ICST Conference on Security and Privacy in Mobile Information and Communication Systems (MOBISec 2011), Aalborg, Denmark
9. Lane ND, Eisenman SB, Musolesi M, Miluzzo E, Campbell AT (2008) Urban sensing systems: opportunistic or participatory? In: Proceedings of the 9th workshop on Mobile computing systems and applications, ACM, New York, NY, USA, HotMobile '08, pp 11–16, doi: <http://doi.acm.org/10.1145/1411759.1411763> 25. Oktober 2011
10. Eisenman SB, Lane ND, Miluzzo E, Peterson RA, seop Ahn G, Campbell AT (2006) Metro-sense project: People-centric sensing at scale. In: In WSW 2006 at Sensys

11. Shin M, Cornelius C, Peebles D, Kapadia A, Kotz D, Triandopoulos N (2011) AnonySense: A system for anonymous opportunistic sensing. *Pervasive Mob Comput* 7:16–30
12. Abdelzaher T, Anokwa Y, Boda P, Burke J, Estrin D, Guibas L, Kansal A, Madden S, Reich J (2007) Mobicopes for human spaces. *IEEE Pervasive Computing* 6:20–29, doi: <http://doi.ieeecomputersociety.org/10.1109/MPRV.2007.38>, 25. Oktober 2011
13. Das T, Mohan P, Padmanabhan VN, Ramjee R, Sharma A (2010) Prism: platform for remote sensing using smartphones. In: *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*, ACM, New York, NY, USA, pp 63–76, doi: <http://doi.acm.org/10.1145/1814433.1814442>, 25. Oktober 2011
14. Miluzzo E, Lane N, Campbell A, Olfati-Saber R (2008) Calibre: A self-calibration system for mobile sensor networks. In: Nikoletseas S, Chlebus B, Johnson D, Krishnamachari B (eds) *Distributed Computing in Sensor Systems, Lecture Notes in Computer Science*, vol 5067, Springer Berlin / Heidelberg, pp 314–331
15. Miluzzo E, Papandrea M, Lane ND, Lu H, Campbell AT (2010) Pocket, Bag, Hand, etc.- Automatically Detecting Phone Context through Discovery. <http://www.cs.dartmouth.edu/~miluzzo/papers/miluzzo-phonesense10.pdf> 25. Oktober 2011
16. Russell S, Norvig P (2010) Artificial intelligence: a modern approach. Prentice Hall series in artificial intelligence, Prentice Hall
17. Langley P, Simon H (1995) Applications of machine learning and rule induction. *Communications of the ACM* 38(11):54–64
18. Michie D (1989) Problems of computer-aided concept formation. *Applications of expert systems* 2:310–333
19. Riese C (1985) Transformer fault detection and diagnosis using rule-master by radian. Radian Corp, Austin, TX, Paper
20. Mangold WG, Faulds DJ (2009) Social media: The new hybrid element of the promotion mix. *Business Horizons* 52(4):357–365, doi: 10.1016/j.bushor.2009.03.002
21. Cyganski P, Hass BH (2008) Potenziale sozialer Netzwerke für Unternehmen, vol 17, Springer Berlin Heidelberg, pp 101–120
22. Onyechi GC, Abeyasinghe G (2009) Adoption of web based collaboration tools in the enterprise: challenges and opportunities. *2009 International Conference on the Current Trends in Information Technology (CTIT)*, pp 1–6
23. Back A, Gronau N, Tochtermann K (2009) Web 2.0 in der Unternehmenspraxis
24. Dürr M, Werner M, Maier M (2010) Re-Socializing Online Social Networks. In: *Proc of GreenCom–CPSCom'10*, IEEE

---

# Kapitel 20

## Integration von Kontextinformationen in Smart Applications und Smart Workflows

Dirk Bade und Winfried Lamersdorf

**Zusammenfassung** In *Smart Mobile Applications* sind virtuelle und physische Welt in immer weiter zunehmendem Maße miteinander verknüpft: So werden u. a. insbesondere mobile Rechner und andere Geräte mit immer mehr Sensoren ausgestattet. Mit Hilfe derer können sie z. B. ihren eigenen Zustand, Attribute der physischen Umwelt, andere sie umgebende Objekte sowie Aktivitäten eines Benutzers wahrnehmen und für neuartige Anwendungen auswerten. Hierfür bedarf es der Integration von Kontextinformationen in solche „smartten“ Anwendungen, was eine Reihe von Herausforderungen, insbesondere für mobile Anwendungen, mit sich bringt. Im Rahmen dieses Artikels sollen daher verschiedene Möglichkeiten der Bereitstellung und existierende Ansätze der Integration betrachtet und hinsichtlich ihrer Eignung für mobile Systeme diskutiert werden. Um die Integration und den Umgang mit Kontextinformationen weitergehend zu flexibilisieren, stellen wir zudem einen Top-down-Ansatz der Integration als konzeptionelle Ergänzung zu existierenden Ansätzen sowie eine prototypische Systemunterstützung vor.

### 20.1 Einleitung

Durch die fortschreitenden Entwicklungen im Bereich des Mobile Computing erlangen Anwendungen in zunehmendem Maße die Fähigkeit ihre eigene Umwelt, bzw. den Kontext, in dem sie verwendet werden, wahrzunehmen. Dies ermöglicht diesen

---

Dirk Bade  
Universität Hamburg, Verteilte Systeme und Informationssysteme,  
E-mail: bade@informatik.uni-hamburg.de

Winfried Lamersdorf  
Universität Hamburg, Verteilte Systeme und Informationssysteme,  
E-mail: lamersd@informatik.uni-hamburg.de

sogenannten *Smart Applications* bzw. *Smart Workflows*, sich in „intelligenter“ Weise zur Laufzeit an die aktuelle Umgebungssituation anzupassen. Das Ziel hierbei ist es nicht nur, die sogenannte „User Experience“ zu erhöhen, sondern Benutzer auch durch eine weitergehende Automatisierung von Aufgaben zu entlasten. Zu diesem Zweck müssen lokal vorhandene oder entfernt abrufbare Kontextinformationen in Form von Datenströmen kontinuierlich an die jeweiligen Konsumenten übertragen und anwendungs- oder dienstseitig ausgewertet werden. Auf mobilen Geräten, die trotz begrenzter Ressourcen inzwischen oft bereits eine Vielzahl an Anwendungen nebenläufig ausführen, bedeutet dies jedoch einen großen Zusatzaufwand und eine hohe Auslastung der knappen Ressourcen, da jede dieser neuartigen Anwendungen und jeder zugehörige Dienst eine unter Umständen große Menge an unterschiedlichen Kontextdaten individuell bereitgestellt bekommt und anschließend auch handhaben und auswerten muss.

Eine mögliche Abhilfe stellen lokal auf einem Gerät laufende Kontextdaten-Dienste dar, welche die anwendungsübergreifende Erhebung, ggf. Persistierung und Vorverarbeitung lokaler Kontextdaten zu zentralisieren erlauben und somit sowohl die Komplexität als auch die Ressourcenanforderungen einzelner Anwendungen reduzieren. Einen ähnlichen Ansatz verfolgen auch sogenannte *Context-Provisioning-Systeme*, die in der Infrastruktur laufen und Kontextdaten von einer Vielzahl an Kunden sammeln, ggf. persistieren, vorverarbeiten und zum allgemeinen Abruf wieder zur Verfügung stellen. Für beide Arten der Bereitstellung von Kontextdaten existieren bereits eine Reihe von Frameworks, Toolkits und Systemen, die entweder auf eine spezielle Anwendungsdomäne zugeschnitten sind oder generische Funktionen zum Umgang mit Kontextdaten bereitstellen. Spezialisierte Systeme tragen dabei maßgeblich zur Reduzierung der Komplexität in Anwendungen und Diensten bei, dies jedoch auf Kosten der Generizität. Domänenunabhängige Lösungen hingegen unterstützen zwar die Kontextdaten-Verarbeitung, jedoch muss sich die Anwendung bzw. der Dienst entweder eng an die gebotene Lösung koppeln, was die Wiederverwendbarkeit einschränkt, oder einen Teil der Verarbeitungslogik selbst implementieren um sich an die verwendete Lösung zu adaptieren. Letzteres stellt insbesondere für Smart Workflows eine Herausforderung dar, weil Logik in das Prozessmodell integriert oder das jeweils verwendete Workflow-Management-System erweitert werden muss.

Je nach Anwendungsfall unterscheidet sich daher die Wahl einer Lösung zur Bereitstellung und Integration von Kontextdaten. In diesem Artikel stellen wir ein Konzept zur flexiblen, domänenunabhängigen Top-down-Integration von Kontextinformationen vor, welches die Injektion domänenabhängiger Verarbeitungsschritte in den Vorgang der Kontextdaten-Verarbeitung erlaubt und dabei sowohl komplett lokal als auch beliebig verteilt ausgeführt werden kann. Dadurch wird die Integration von Kontextinformationen in mobile Anwendungen dahingehend erleichtert, dass einzelne Smart Applications bzw. Workflows Kontextdaten in einer für sie jeweils passenden Form und Qualität beziehen und auf möglichst ressourcenschonende Weise verarbeiten können.

Im folgenden Teil dieses Artikels werden zunächst die Begriffe *Smart Application* und *Smart Workflow* anhand von Definitionen und Anwendungsbeispielen einge-

führt. Darauf folgend widmet sich der dritte Abschnitt der Frage, wie Kontextdaten als höherwertige Informationen überhaupt bereitgestellt, integriert und somit nutzbar gemacht werden können. Nachdem Probleme existierender Systemunterstützungen aufgezeigt wurden, präsentieren wir im vierten Abschnitt unser Konzept einer flexiblen Top-down-Integration. Abschnitt 5 schließt mit einer zusammenfassenden Diskussion der verschiedenen Ansätze.

## 20.2 Smart Applications und Smart Workflows

Mit der Eigenschaft „smart“ werden heutzutage geradezu inflationär Produkte oder Techniken attribuiert, auch wenn diese alles andere als wirklich intelligentes (oder „smartes“) Verhalten offenbaren. Aus diesem Grund soll einleitend das Verständnis insbesondere dieses Begriffes konkretisiert werden.

### 20.2.1 Begriffsdefinitionen

Das Adjektiv „smart“ wird nicht nur im Deutschen häufig mit dem etwas formelleren „intelligent“ gleichgesetzt. So wird z. B. das *Smart Grid* zum „intelligenten Stromnetz“, das *Smart Home* ermöglicht „intelligentes Wohnen“ und *Smart Objects* nehmen eine „intelligente Überwachung“ von Betriebsmitteln vor. Mit dem traditionellen Verständnis des Begriffes „Intelligenz“ (vom lat. *intellegere* „verstehen“) als einem Sammelbegriff für die kognitive Leistungsfähigkeit eines Subjekts oder Objekts, haben die neuen Wortschöpfungen allerdings wenig gemein. Betrachtet man die Charakteristiken der als „smart“ beworbenen Produkte und Techniken, so offenbart sich oft ein großes Spektrum an Eigenschaften, die von der einfachen Wahrnehmung einer physikalischen Kenngröße (z. B. *Smart Fan*, ein temperaturgesteuerter Lüfter) bis hin zur Fähigkeit, eigene Entscheidungen auf Basis eines kognitiven Modells zu treffen (z. B. „intelligente“ *Belief-Desire-Intention (BDI-)Agenten*), reichen. In der Literatur (z. B. [1]) spricht man daher der genaueren Abgrenzung halber auch von *verschiedenen Ebenen* der Intelligenz. Wir folgen in unserem Verständnis der Übersetzung des Begriffes „smart“ nach [2] als „knowledgeable“ (kenntnisreich, gut unterrichtet) und als „operating by automation“ (Ausführung durch Automation), denn Letzteres kann als kleinstes gemeinsames Ziel bei der Verwendung des Begriffes „smart“ angesehen werden, welches Ersteres, nämlich die Kenntnis der eigenen Umwelt, als eine Grundvoraussetzung hat. In [3] findet sich folgende zusammenfassende Definition: „*Smart Systems are those that are able to adapt their operations to the current context without explicit user intervention*“, welche auch die Grundlage des Verständnisses von „smart“ in diesem Artikel bildet. Wie ersichtlich ist, sind Informationen über die eigene Umwelt, bzw. den eigenen Kontext, von fundamentaler Bedeutung für Smart Applications. Der Begriff Kontext wird nach [4] als jedwede Information verstanden, die zur Charakterisierung der Situati-

on einer Entität verwendet werden kann. Für gewöhnlich unterteilt man Kontext in verschiedene Kategorien, wie persönlicher oder sozialer Kontext, Umwelt-Kontext etc. [5]. Allen Kategorien gemein ist die Tatsache, dass zur Erhebung von Kontextdaten eine Art Sensor benötigt wird. Dies muss nicht zwangsläufig ein physischer Sensor, z. B. zum Messen der Temperatur oder zum Aufnehmen eines Fotos, sein, sondern ist in vielen Fällen rein virtuell und durch Software realisiert, um z. B. die aktuelle CPU-Auslastung oder das Eintreffen einer neuen E-Mail zu „messen“.

### **20.2.2 Anwendungsbeispiele**

Kontextinformationen stellen also eine wesentliche Voraussetzung für Smart Applications dar. Es ist jedoch nicht unerheblich, wofür die Kontextinformationen letztlich in der Anwendung verwendet werden. Das Datenbank-Management-System eines Geoinformationssystems beispielsweise würde man nicht zwangsläufig als Smart Application bezeichnen, obwohl es auch Kontextdaten (z. B. Positionsdaten) verarbeitet. Um aber das Attribut „smart“ im hier verwendeten Sinn zu tragen, muss eine Anwendung Kontextinformationen interpretieren und ihre Funktionsweise diesen entsprechend anpassen. Ein einfaches Beispiel ist die *Smart Fan Control*, eine Steuersondware für den Prozessorlüfter eines Computers, welche die Drehgeschwindigkeit des Lüfters der aktuell gemessenen Prozessortemperatur anpasst. Ein anderes, viel zitiertes Beispiel ist ein *Smart Restaurant Finder* für mobile Geräte [6], welcher auf Basis des aktuellen Aufenthaltsortes, der persönlichen Präferenzen des Klienten, dessen Historie sowie der begleitenden Personen Vorschläge für geeignete Restaurants in der Umgebung macht.

Ein weiteres Beispiel kommt aus dem Bereich der Produktionsautomatisierung (engl. *Smart Manufacturing Control*). Bei der Herstellung von Gütern, z. B. Autos, durchlaufen diese in einem Produktionsprozess mehrere Stufen, in denen sie von speziellen Maschinen sequentiell bearbeitet werden. Hierbei gilt es, den Durchsatz von Gütern auf einem möglichst hohen Niveau zu halten. Durch Verschleiß oder sonstige Fehlfunktionen kann es jedoch zu temporären Ausfällen von Maschinen kommen, wodurch im ungünstigsten Fall eine komplette Produktionsstraße stillgelegt werden muss. In [7] wird deshalb vorgeschlagen, die gesamte Koordination der Güter, Transportbänder und Maschinen selbstorganisiert durch intelligente Agenten vorzunehmen. Die Agenten repräsentieren die einzelnen Teile des Systems und verhandeln untereinander den bestmöglichen Ablauf der Produktion, indem sie anhand ihres jeweiligen Wissens über den eigenen Zustand autonom Entscheidungen treffen. Anders als bei [8] greifen sie dabei jedoch nicht auf Informationen über ihre physische Umgebung zurück, sondern lediglich auf abstrakte Informationen ihrer virtuellen Umwelt (z. B. Anzahl an Elementen in der virtuellen Warteschlange einer Maschine). In [8] hingegen werden die Maschinen sowie das Personal mit Sensoren ausgestattet, sodass stetig Informationen z. B. über den Verschleißzustand oder den Aufenthaltsorts erhoben werden. Lässt sich anhand der Sensorinformationen

ableiten, dass eine Maschine bald auszufallen droht, wird präventiv ein Mitarbeiter verständigt, der dann zeitnah einen Austausch an der Maschine vornehmen kann.

Die vorgestellten Beispiele zeigen bereits die Notwendigkeit der Nutzung von Kontextdaten für Smart Systems. Dabei wird deutlich, dass heterogene Kontextdaten auf vielfältige Weise Verwendung finden. Aber allen Beispielen ist gemein, dass das übergeordnete Ziel die Anpassung einer Anwendung oder eines Prozesses an dynamische Umgebungen ist. Die Frage ist nun, wie eine Anwendung die notwendigen Informationen über den Kontext bereitgestellt bekommen und integrieren kann, was im folgenden Abschnitt näher betrachtet wird.

## 20.3 Bereitstellung und Integration von Kontextinformationen

Um Kontextinformationen in eine Smart Application oder einen Smart Workflow integrieren zu können, müssen diese zuerst aus ein oder mehreren Quellen beschafft werden. Aktuelle kontextbasierte Applikationen greifen auf Sensordaten meist über eine speziell auf die jeweilige Ausführungsplattform zugeschnittene Programmbibliothek zu und verarbeiten sie anschließend applikationsintern weiter. Kontextbasierte Workflows hingegen verlassen sich meist auf eine Erweiterung des Workflow-Management-Systems zur Beschaffung von Kontextinformationen oder rufen diese von einem speziellen Kontextdienst ab. Alle existierenden Lösungen sind nur bis zu einem gewissen Grad flexibel und wiederverwendbar, da bei einem Wechsel der Ausführungsplattform oder der Anwendungsdomäne oft erhebliche Anpassungen der Anwendungen oder Workflows notwendig sind.

Das übergeordnete Ziel sollte es daher sein, den intelligenten Anwendungen nur die für sie relevanten Kontextinformationen in geeigneter Art und Weise zu präsentieren. Man spricht in diesem Fall auch von Ereignissen auf Anwendungsebene (sog. *Application Level Events*) [9]. Auf derartige Ereignisse lassen sich zwei unterschiedliche Perspektiven einnehmen, welche den Grad der Domänenabhängigkeit beeinflussen: In der *Bottom-up-Perspektive* (oder auch Perspektive des Produzenten) werden, ausgehend von den rohen Messdaten, verschiedene Verarbeitungsschritte durchlaufen um diese Daten letztlich in einer gängigen Form, meistens einem bestimmten Standard folgend, über eine Schnittstelle zur Verfügung zu stellen. Ein Beispiel hierfür sind GPS-Sensoren, welche die aktuelle Position im *NMEA 0183*-Format bereitstellen. Je nach Anwendung kann diese entweder direkt mit den NMEA-Angaben in Bogen-Minuten und -Sekunden arbeiten oder in einem zusätzlichen Verarbeitungsschritt Längen- und Breitengrade oder Adressen berechnen.

Während also bei der Bottom-up-Perspektive die Frage im Vordergrund steht, wie die Daten für möglichst viele Anwendungsbereiche in sinnvoller Form bereitgestellt werden können, fokussiert die *Top-down-Perspektive* (oder auch Perspektive des Konsumenten) eine bestimmte Anwendung und konzentriert sich auf die Frage, welche Verarbeitungsschritte die Daten durchlaufen müssen, um den Integrationsaufwand für die Anwendung zu minimieren. Als Beispiel hierfür lässt sich ein Smart Workflow für die Wartung von Maschinen anführen, der bei Abnutzung

eines Werkzeugs den Austausch organisieren soll (vgl. Abschnitt 2.2). Hierbei kann die Abnutzung durch ein Zusammenspiel mehrerer Faktoren (Drehgeschwindigkeit des Bohrers, Abnutzung des Bohrkopfes, Material) bestimmt sein, die dem Workflow idealerweise als ein einmaliges höherwertiges Ereignis („Bohrkopf muss ausgetauscht werden“) bereitgestellt werden sollen. Da die Verarbeitungslogik zur Konstruktion solch höherwertiger Ereignisse in die unteren Schichten der Bereitstellung ausgelagert wird, reduziert sich anwendungsseitig der Integrationsaufwand für Kontextinformationen, dies jedoch zu Lasten der Domänenunabhängigkeit.

### **20.3.1 Bereitstellung von Kontextinformationen**

Die Bereitstellung von Kontextinformationen ist eine grundlegende Voraussetzung dafür, dass sich Smart Applications oder Smart Workflows an die aktuellen Gegebenheiten anpassen können. Bevor jedoch ein Ereignis in der Umwelt einer Entität als höherwertige Kontextinformation bereitgestellt werden kann, müssen unter Umständen eine Reihe von (komplexen) Verarbeitungsschritten durchlaufen werden [10]. Um das Erstellen intelligenter Anwendungen und Dienste zu vereinfachen, deren Komplexität zu reduzieren und die Wiederverwendbarkeit der Verarbeitungslogik zu ermöglichen, ist es daher sinnvoll, alle Verarbeitungsschritte von der Messung hin zur Bereitstellung aus der Anwendung auszulagern. Im Falle mobiler Geräte kann es zudem sinnvoll sein, Teile der Verarbeitung sogar in die Infrastruktur zu verschieben, um die Ressourcen des Gerätes zu entlasten. Gegebenenfalls stehen auf dem mobilen Gerät auch gar nicht alle Daten zur Konstruktion von Anwendungereignissen zur Verfügung und müssen aus externen Quellen abgerufen werden.

Generell unterscheidet man drei verschiedene Ansätze zur Abfrage von Kontextdaten [11]: Der einfachste Ansatz ist die *lokale Bereitstellung* bzw. Abfrage der Sensordaten über eine gebotene Programmierschnittstelle oder – im Falle externer Sensoren – über eine einfache Kommunikationsschnittstelle. Die Verarbeitung der Sensordaten obliegt in diesem Fall vollständig der jeweiligen Anwendung. Diese einfache Art der Abfrage bzw. Verarbeitung ist wenig mächtig, schränkt die Allgemeinheit und Wiederverwendbarkeit der Abfrage- und Verarbeitungslogik ein und erlaubt zudem lediglich die Verwendung lokal erhobener Kontextdaten. Interessieren sich mehrere Anwendungen auf demselben Gerät für dieselben Kontextdaten, kann es darüber hinaus vorkommen, dass manche Verarbeitungsschritte mehrfach ausgeführt oder die Daten redundant gespeichert werden. Eine andere Möglichkeit besteht darin, den Zugriff auf Kontextdaten auf externe Anbieter auszulagern. Hier unterscheidet man zwischen einem *zentralisierten* und einem *dezentralen* Ansatz: Beim zentralisierten Ansatz schicken die Sensoren ihre Kontextdaten an einen zentralen Dienst in der Infrastruktur. Dieser sammelt, persistiert und verarbeitet die Daten und stellt sie den Anwendungen wiederum zur Verfügung. Auf diese Weise müssen Daten lediglich einmal erhoben werden, können aber von mehreren – auch auf verschiedenen Geräten laufenden – Anwendungen genutzt werden. Auch lassen

sich in der Infrastruktur komplexere, ressourcenintensive Verarbeitungsfunktionen (z. B. Klassifizierung) realisieren, was die mobilen Geräte entlastet. Das Gleiche gilt auch für den dezentralen Ansatz, bei dem Kontextdaten durch das Aufspannen von Ad-hoc-Netzwerken von umliegenden Nachbarn angefragt werden können. Auch auf diese Weise kann die Datenerhebung und -verarbeitung auf verschiedene Knoten verteilt werden. Für alle Ansätze existieren bereits eine Reihe von Toolkits und Frameworks, deren Kernaufgabe die Erhebung, Verarbeitung und Bereitstellung von Kontextinformationen ist. Eine Auswahl dieser Arbeiten soll im Folgenden kurz vorgestellt werden.

### 20.3.1.1 Existierende Ansätze zur Bereitstellung von Kontextinformationen

Jede der beschriebenen Perspektiven und Verteilungsstrategien weist Vor- und Nachteile auf. Es hängt von dem jeweiligen Anwendungsfall und der jeweiligen Situation ab, welcher Ansatz sinnvollerweise Verwendung finden sollte.

Bei *Contory* [11] handelt es sich um eine Middleware für mobile Telefone, welche eine Reihe wohldefinierter Verantwortlichkeiten bzgl. der Bereitstellung von Kontextinformationen zusammenfasst. Auf unterster Ebene werden Sensoren über sogenannte *References* (auch entfernt) verfügbar gemacht. Die Messwerte der Sensoren werden in eine einheitliche Repräsentation überführt, durch Aggregatoren verarbeitet und können von Konsumenten unter Verwendung einer deklarativen Sprache angefragt oder abonniert werden. *Contory* nimmt eine Produzentenperspektive ein, bietet aber durch eine relativ mächtige Anfragesprache auch für Konsumenten eine einfache Möglichkeit zur Auswahl relevanter Kontextdaten. Ein letzter Anpassungsschritt obliegt jedoch den Anwendungen, da diese ihre weitergehende Verarbeitung an die von *Contory* verwendete Informationsrepräsentation anpassen müssen, was zu einer stärkeren Kopplung führt. Zudem sind die Möglichkeiten *Contorys* zur Verarbeitung von Kontextdaten stark eingeschränkt und Konsumenten haben keinerlei Möglichkeit, hierauf Einfluss zu nehmen.

In [10] beschreiben Schmidt und van Laerhoven ebenfalls eine Architektur zur Bereitstellung von Kontextdaten. Auf unterster Ebene nehmen Sensoren primitive Messwerte auf. Diese erfahren in der nächsthöheren Ebene durch sogenannte *Cues* einen Verarbeitungsschritt, wobei pro Sensor mehrere sequentielle *Cues* Verwendung finden können. Aus den Daten der *Cues* wird schließlich mit Hilfe von Regeln oder KI-Techniken eine abstrakte Kontextrepräsentation, z. B. „Person schläft“ abgeleitet und Anwendungen über eine Datenstruktur, z. B. einen Tuple Space, zur Verfügung gestellt. In ihrer Architektur nehmen die Autoren damit ebenfalls eine Produzentenperspektive ein, da sie die Daten ohne Kenntnis des intendierten Anwendungszwecks zuerst vorverarbeiten und schließlich den Konsumenten zur Verfügung stellen. Auch hier haben die Konsumenten keinen Einfluss auf die interne Verarbeitung der Daten durch die *Cues* und müssen sich an die verwendeten Datenstrukturen auf Anwendungsebene selbst anpassen.

Einen ähnlichen Ansatz verfolgt auch das *Context Toolkit* [12], das Kontextinformationen über sogenannte *Widgets* bereitstellt. Diese Widgets setzen sich aus

Generatoren (für die Anbindung von Sensoren), Interpretern (zur Abstraktion von Sensordaten hin zu höherwertigen Informationen) und Servern (für die Persistierung und Bereitstellung von Informationen) zusammen und können auch in einem Verbund zusammengefasst werden. Somit nehmen auch sie die Produzentenperspektive ein und auch für ihren Ansatz gelten dieselben, bereits oben beschriebenen Probleme der Anpassung für Anwendungen und Prozesse.

Ausgehend von der Feststellung, dass die Schnittstellen von Systemen zur Bereitstellung von Kontextinformationen zu komplex sind, um die Informationen einfach in Smart Workflows integrieren zu können, wird in [8] das *Context Integration Process Pattern* propagiert. Um aus Sicht eines Smart Workflows auf Kontextinformationen in geeigneter Weise zugreifen zu können, verwenden die Autoren einen hierarchischen Webservice-Stack, der auf oberster Ebene speziell auf eine Domäne zugeschnittene höherwertige Dienste anbietet. Diese greifen auf einen oder mehrere Dienste der nächst niederen Ebene des Stacks zu, welche zwar domänenunabhängig, aber bereits auf ein bestimmtes Context Provisioning System (in diesem Fall *Nexus* [13]) zugeschnitten sind. Mit dieser Vorgehensweise nähern sie sich dem Problem der Anpassung sowohl aus der Produzenten- als auch aus der Konsumenten-Perspektive. Die domänenunabhängigen Schichten, welche direkt über dem Produzenten angesiedelt sind, überbrücken die syntaktische Kluft zwischen der Bereitstellung von Kontextdaten und deren Nutzung, während die darüber liegende domänenabhängige Schicht für die Bereitstellung abstrakter Funktionen zuständig ist und den Smart Workflows genau die relevanten Informationen in geeigneter Form zur Verfügung stellt.

Nach dieser Auswahl existierender Ansätze zur Bereitstellung von Kontextinformationen werden im Weiteren Ansätze zur Integration dieser Informationen in bestehende Anwendungen und Workflows vorgestellt.

### **20.3.2 Integration von Kontextinformationen**

Die Herausforderungen an die Integration von Kontextinformationen unterscheiden sich wesentlich zwischen Smart Applications und Smart Workflows. Einer der Gründe hierfür ist die höhere Flexibilität im Umgang mit Daten, die Anwendungen gegenüber Workflows genießen. Ein Workflow kann empfangene Kontextinformationen nur bedingt weiterverarbeiten (dekodieren, konvertieren etc.), während eine Anwendung relativ flexibel in deren Verwendung ist.

Bei Smart Applications kann zunächst unterschieden werden, ob Kontextinformationen nur kurzzeitig relevant sind (z. B. Anzeigen der aktuellen Position auf einer Karte) oder ob die Informationen z. B. aggregiert oder in ein Kontextmodell integriert werden sollen. Ersteres ist relativ trivial, da bereitgestellte Kontextinformationen meist direkt für den intendierten Zweck verwendet oder mittels eines weiteren Verarbeitungsschrittes nutzbar gemacht werden können. Interessanter ist der zweite Punkt, den bereits zahlreiche Frameworks und Toolkits (vgl. Abschnitt 20.3.1.1)

unterstützen, sodass auch diese Art der Integration letztlich auf die beschriebene triviale Weise zurückgeführt werden kann.

Das Problem der Integration von Informationen in einen Smart Workflow besteht darin, dass dieser im Wesentlichen die zeitlich-logische Abfolge einer Reihe von Aktivitäten beschreibt [14]. Es ist nicht vorgesehen, dass Anwendungslogik in solche Prozesse integriert wird. Was bei der Informationsintegration in Anwendungen eine triviale Aufgabe ist, lässt sich in Workflows daher nicht ohne Weiteres realisieren. Aus diesem Grund kommt es bei der Integration von Kontextinformationen darauf an, dass diese in geeigneter Art und Repräsentation bereitgestellt werden, damit sie ohne zusätzlichen Verarbeitungsaufwand an die nächste Aktivität im Workflow weitergereicht werden können. Hier lassen sich zwei Ansätze der Systemunterstützung unterscheiden:

- Ein existierendes Workflow-Management-System wird dahingehend erweitert, dass es selbst Kontextdaten abfragen, verarbeiten und den laufenden Prozessen zur Verfügung stellen kann. Dies geht mit einer Erweiterung der Workflow-Beschreibungssprache um Konstrukte zum Austausch von Kontextinformationen zwischen einem Prozess und der ausführenden Workflow Engine einher.
- Der andere, nichtinvasive Ansatz beruht auf einem domänenabhängigen Kontextdienst, welcher Kontextinformationen je nach Erfordernis vorverarbeitet und schließlich selbst wieder bereitstellt. Prozesse können diesen Kontextdienst wie jeden anderen Dienst aufrufen und bekommen so die gewünschten Informationen in einer geeigneten Repräsentation, sodass diese ohne weitere Verarbeitung an nachfolgende Prozessaktivitäten weitergereicht werden können.

Im Folgenden werden beide Varianten anhand einer Auswahl existierender Ansätze vorgestellt und die jeweiligen Vor- und Nachteile diskutiert.

### 20.3.2.1 Existierende Ansätze zur Integration von Kontextinformationen

*COPAL* [15] adressiert sowohl die Bereitstellung als auch die Integration von Kontextinformationen in bestehende Anwendungen. Hierbei werden Sensordaten durch sogenannte Kontextprozessoren verarbeitet. Diese lassen sich beliebig konkatenieren und Konsumenten können eigene Prozessoren mittels der domänenspezifischen Sprache *COPAL-DSL* definieren und dem System zur Laufzeit bekannt machen. Auf diese Weise lässt sich die Datenverarbeitung weitestgehend individuell an die Bedürfnisse der Applikationen anpassen. Dieser Ansatz nimmt vom Grundsatz her zwar die Produzenten-Perspektive ein, damit aber Konsumenten die bereitgestellten Kontextinformationen möglichst einfach integrieren können, können diese eigene Prozessoren für den Verarbeitungsprozess bereitstellen und so Einfluss auf die Qualität und Form der Daten nehmen.

Abbas und Shaikh beschreiben in [3] ein *Smart Workflow Management Framework*, welches die kontextadaptive Ausführung sogenannter *Smart Activities* erlaubt. Diese repräsentieren eine Art Platzhalter für eine Sammlung einzelner Aktivitäten, die von einem als Dienst des Workflow-Management-Systems realisierten Context

Manager zur Laufzeit durch eine kontextadäquate Aktivität substituiert werden. Für die Auswahl einer geeigneten Aktivität führt der Context Manager eine durch die Smart Activity vorgegebene Kontextabfrage auf eine domänenabhängige Wissensbasis oder eine Menge von Sensoren aus, deren Ergebnis die Wahl der auszuführenden Aktivität beeinflusst. Die Integration erfolgt damit in Teilen nach Maßgabe des Konsumenten, da dieser durch die Kontextabfrage zumindest die Art der Informationen bestimmen kann, nicht jedoch die Repräsentation oder etwaige Verarbeitungsschritte wie bei COPAL.

CAWE [16] verfolgt einen ähnlichen Ansatz. Hier werden kontextsensitive Aktivitäten ebenfalls durch einen Platzhalter, eine *Abstract Activity*, repräsentiert. Ein kontextsensitiver *Workflow Manager* kapselt eine Workflow Engine zur Ausführung von Prozessen und ruft bei Ausführung einer abstrakten Aktivität ein sogenanntes Personalisierungsmodul auf, welches anhand von Kontextinformationen die konkrete, auszuführende Aktivität bestimmt. Die Integration von Kontextinformationen in einen Prozess dient hier im Wesentlichen der Auswahl kontextspezifischer Aktivitäten, wobei diese die Informationen durchaus auch global im Prozess bekanntmachen können. Jedoch ist der Prozessentwurf recht komplex, Kontextinformationen sind nur implizit repräsentiert und ein Prozess selbst kann keinen Einfluss auf die Verarbeitung von Kontextdaten nehmen.

*Sentient Processes* [17] realisieren kontextsensitive, prozessbasierte Applikationen, die aus einer Sequenz individueller Schritte bestehen. Schritte beschreiben Aktionen, deren Verhalten durch Angabe von Kontextbedingungen gesteuert werden können, wobei die Kontextdaten durch eine entsprechende Infrastruktur bereitgestellt werden müssen. Zur Modellierung der Prozesse und Bedingungen wird die *Sentient Process Command Language* verwendet, welche durch eine spezielle Workflow Engine interpretiert werden kann. Durch diesen konsumentenbasierten Ansatz wird zwar eine einfache Integration von Kontextinformationen in die Prozesse ermöglicht, allerdings erfordert dies eine Anpassung der Workflow Engine, damit diese die relevanten Informationen in einem geeigneten Format zur Verfügung stellen kann. Prozesse selbst können zur Laufzeit keinen Einfluss mehr auf die Verarbeitung der Kontextinformationen nehmen.

*Context4BPEL* [18] erweitert die *Business Process Execution Language* (BPEL) um zusätzliche Aktivitäten, damit der Einfluss von Kontextinformationen auf Workflows explizit modelliert werden kann. In der vorgestellten Arbeit sorgt die Nexus-Plattform [13] für die Bereitstellung von Kontextinformationen und erlaubt den Zugriff auf diese mittels Webservices, welche von der Workflow Engine aufgerufen werden können. Auf diese Weise lassen sich Kontextinformationen relativ elegant in das Workflow-Modell integrieren. Allerdings können die Verarbeitungsvorschriften für Kontextinformationen nicht spezifiziert werden, der Workflow muss sich also zu einem geringen Grad an das Context Provisioning System anpassen, wobei eine gewisse technische Abhängigkeit bzgl. der Nexus-Plattform auch im Modell verbleibt. Dies wirkt sich z. B. darin aus, dass der Zugriff auf bestimmte Kontextattribute speziell auf Nexus zugeschnitten ist.

Unabhängig von der Art des verwendeten Ansatzes, also entweder der Erweiterung von Prozesssprache und Workflow Engine oder der Nutzung eines speziellen

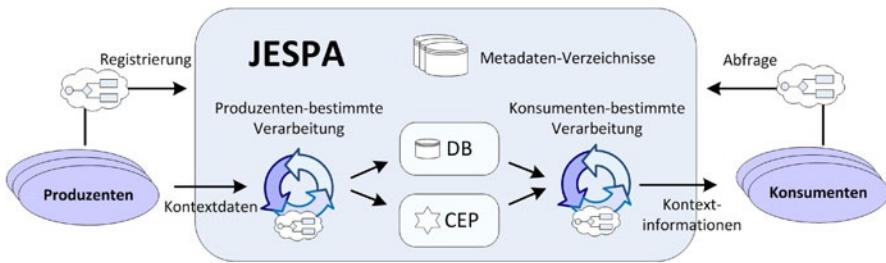
Kontextdienstes, werden bestimmte Anpassungen verlangt. Im ersten Fall müssen das Workflow-Management-System und die Prozessbeschreibungssprache erweitert werden, damit Kontextinformationen explizit bei der Modellierung und Ausführung des Prozesses Verwendung finden können. Dies schränkt einerseits die Interoperabilität ein, andererseits ergibt sich jedoch der Vorteil, dass der Umgang mit Kontext in der Prozessbeschreibung explizit geschieht, was die Modellierung und das Verständnis der Prozessbeschreibungen erleichtert. Im zweiten Fall, der Nutzung eines Kontextdienstes, verschiebt sich der Anpassungsaufwand in den Kontextdienst, da jener auf die jeweilige Anwendungsdomäne unter Umständen individuell zugeschnitten werden muss. Auch geschieht die Verwendung von Kontextinformationen nunmehr nur noch implizit, wodurch die Prozessbeschreibung komplexer und deren Lesbarkeit unter Umständen erschwert wird.

## 20.4 Vermittlung kontextbasierter Sichten

Im vorigen Kapitel wurden die Bereitstellung und die Integration von Kontextinformationen betrachtet, wobei als wesentliches Problem der existierenden Ansätze die fehlende Flexibilität der Bereitstellung und die nicht vorhandene Möglichkeit, konsumentenseitig Einfluss auf den Verarbeitungsprozess der Informationen zu nehmen, identifiziert wurde, was eine Integration wesentlich erschwert. Im Folgenden soll daher unser Ansatz für die Vermittlung sogenannter *kontextbasierter Sichten* vorgestellt werden, der sowohl die Produzenten- als auch die Konsumentenperspektive unterstützt, domänenunabhängig ist, aber zur Laufzeit an individuelle Bedürfnisse angepasst werden kann und beliebige Verteilungsstrategien für die Verarbeitung und den Zugriff auf Kontextinformationen erlaubt.

### 20.4.1 Realisierung eines Mediators

Der Ansatz basiert auf einem Mediator, der Kontextinformationen zwischen heterogenen Produzenten und Konsumenten in mobilen, ubiquitären Netzen vermittelt. Hierbei verschiebt sich ein wesentlicher Teil der Verarbeitungslogik für Kontextdaten weg von den Produzenten und Konsumenten hinein in den Mediator, was insbesondere ressourcenschwachen und mobilen Geräten zugute kommt. Realisiert wurde der Mediator durch die *Jadex Event Stream Processing Architecture* (JE-SPA) [19], einer Middleware zur Verarbeitung von Kontextereignissen, deren Funktionen sich beliebig von Produzenten und Konsumenten orchestrieren lassen (siehe Abb. 20.1). Die Orchestrierung wird hierbei mittels BPMN (*Business Process Model and Notation* [14]) beschrieben, einer grafischen Modellierungssprache, durch die der Daten- und Kontrollfluss während einer Prozessausführung spezifiziert werden kann. Auf diese Weise können zum Beispiel die Ausführungsreihenfolge von Verarbeitungsschritten, die Kommunikation mit anderen, externen Diensten oder etwaige



**Abb. 20.1** Abstrakte Architektur von JESPA

Maßnahmen zur Fehlerbehandlung beschrieben werden. Zur Laufzeit werden die BPMN-basierten Prozesse von einer integrierten Workflow-Engine interpretiert und ausgeführt, wobei die Verarbeitung jeder Prozessaktivität entweder durch von JESPA gebotene Dienste oder durch einen benutzerdefinierten externen Dienstleister erbracht wird.

JESPA umfasst bereits einen großen Umfang an Sensor-Adaptoren und Standard-Verarbeitungsfunktionen für Kontextdaten und ist zudem flexibel erweiterbar, so dass auch andere Frameworks und Toolkits zur Kontextdaten-Bereitstellung eingebunden werden können, wodurch eine größtmögliche Flexibilität hinsichtlich der Bereitstellung erzielt wird.

Produzenten registrieren sich bei JESPA mit einem Verarbeitungsprozess, welcher den Mediator anweist, nachfolgend versandte Kontextdaten dieses Produzenten nach einer durch den Prozess festgelegten Art und Weise zu verarbeiten, unabhängig davon, was mögliche Konsumenten bedürfen. So könnte z. B. ein Beschleunigungssensor den Mediator anweisen, alle nachfolgenden Beschleunigungsdaten in bestimmten Intervallen zu glätten um mögliches Rauschen zu entfernen, auszudünnen um die Datenmenge zu reduzieren und schließlich zu persistieren um eine Historie anzulegen. Diese erste Verarbeitungsstufe realisiert die Bottom-up-Sicht, also die Perspektive der Produzenten auf die Bereitstellung.

Nach dieser ersten Verarbeitungsstufe werden die Informationen den Konsumenten einerseits über Datenbanken (DB) zur Abfrage zurückliegender Ereignisse und andererseits über eine sogenannte *Complex-Event-Processing*-Komponente (CEP) [20] zur Detektion zukünftiger komplexer Ereignismuster zugänglich gemacht. In beiden Fällen können auch die Konsumenten einen Verarbeitungsprozess angeben, den die Abfrageergebnisse durchlaufen müssen. Unter Zuhilfenahme eines Metadaten-Verzeichnisses kann ein Konsument zudem relevante Daten auffinden, erhält Aufschluss über deren Beschaffenheit und kann somit nach eigener Maßgabe die finale Verarbeitung und Übermittlung der Daten durch die Middleware bestimmen, wodurch die Top-down-Sicht realisiert wird.

Grundsätzlich sind die Funktionen der Middleware auf die Verarbeitung und Detektion beliebiger komplexer Ereignisse spezialisiert. Durch die Verwendung von Verarbeitungsprozessen können Produzenten und Konsumenten jedoch das Verhalten der Middleware bei der Verarbeitung für sie relevanter Ereignisse beeinflus-

sen und somit auf ihre jeweilige Domäne abstimmen. Die Middleware bringt bereits eine Reihe gängiger Verarbeitungsfunktionen z. B. zur Aggregation (Gruppierung, Klassifizierung), Konvertierung (Einheiten, Formate), Translation (Übersetzung, math. Verschiebung), Transkription (Kodierungen), Reduktion (Bereinigung, Filterung, Kompression) sowie Augmentation (Anreicherung mit Informationen aus externen Quellen) mit sich, jedoch können beliebige externe Endpunkte in einem Verarbeitungsprozess benannt werden, welche spezialisierte Verarbeitungsschritte, z. B. die Konvertierung eines proprietären Formates ins XML-Format, ausführen können. Auf diese Weise können die Funktionen der Middleware nichtinvasiv zur Laufzeit erweitert werden.

Die Bestandteile der Middleware sind als Software-Agenten [21] bzw. aktive Komponenten [22] realisiert, was eine beliebige Verteilung von Aufgaben im Netzwerk erlaubt. Durch spätes Binden von Funktionen können insbesondere auch mobile Teilnehmer, die unter wechselnden Endpunkt-Adressen erreichbar sind, in den Verarbeitungsprozess integriert werden. Zudem weisen die einzelnen Komponenten eine starke Kohäsion auf, was letztlich ein flexibles Deployment des Systems ermöglicht. So wurde etwa in praktischen Experimenten gezeigt, dass neben einer Desktop-Variante auch bereits eine prototypische Implementierung der JESPA-Middleware für die *Android*-Plattform realisierbar ist, der lediglich einige Funktionen fehlen, die aber jeweils bei Bedarf über entsprechende andere Komponenten der Infrastruktur bereitgestellt werden können. Schließlich konnte im Rahmen einer frühen prototypischen Portierung der *Activiti*-Engine [23] zur Ausführung von BPMN 2.0-basierten Prozessen auch gezeigt werden, dass die im Folgenden beschriebene Integration von Kontextinformationen in Smart Workflows und die Ausführung dieser auch auf mobilen Geräten möglich ist.

## 20.4.2 Integrationsbeispiel

Um abschließend aufzuzeigen, wie Kontextdaten mittels der JESPA-Middleware in Smart Workflows integriert werden können, wird im Folgenden kurz auf die Verwendung der BPMN-basierten Verarbeitungsprozesse eingegangen, bevor das Prinzip anschließend anhand zweier Anwendungsbeispiele verdeutlicht wird.

Die grundlegende Idee ist, einen Anwendungs-Workflow um wenige Aktivitäten zur Kommunikation mit der Middleware (z. B. Registrieren, Abfrage von Kontextdaten etc.) zu ergänzen, die bei Ausführung des Workflows durch ein Workflow-Management-System mit den JESPA-Webservices kommunizieren. Des Weiteren kann innerhalb des Anwendungs-Workflows zusätzlich noch der JESPA-Verarbeitungsprozess spezifiziert werden, welcher bei Ausführung der Registrierungsaktivität als Nachrichteninhalt mit an JESPA geschickt wird.

Wir konzentrieren uns im Folgenden auf die Prozessmodellierung mit *BPMN 2.0*, da diese nunmehr auch direkt von einem Workflow-Management-System ausgeführt werden kann. Die Integration der Kontextdaten in Smart Applications oder z. B. BPEL-Workflows beruht auf demselben Prinzip der Angabe eines BPMN-basierten

Verarbeitungsprozesses, den ein Entwickler zur Entwurfszeit der Anwendung bzw. des Prozesses erstellt und den die Anwendung oder der Prozess zur Laufzeit in serialisierter Form einfach an die Middleware übergibt.

#### 20.4.2.1 Anwendungsszenario: Rettung einer gestürzten Person

In diesem Szenario soll einer älteren Person das Leben im eigenen Haushalt möglichst lange ermöglicht werden. Damit bei einem Sturz Angehörige und Ersthelfer alarmiert werden, bekommt diese Person einen Vitalsensor mit WLAN- und GSM-Chip um das Handgelenk, welcher kontinuierlich Beschleunigungsdaten, Blutdruck, Herzfrequenz etc., sowie die Signalstärken der umliegenden WLAN-Stationen (zur Positionsbestimmung) erhebt und an die JESPA-Middleware überträgt. Um nun einen Sturz festzustellen, sollen die bei JESPA eintreffenden Datensätze zuerst geglättet und dann klassifiziert werden. Deutet ein Beschleunigungsmuster auf einen Sturz hin, soll JESPA den Ort des Sturzes anhand des WLAN-Fingerprints bestimmen und dann die Rettungszentrale alarmieren.

Die Spezifikation des Prozesses, also die Verknüpfung der einzelnen Verarbeitungsschritte, kann mit Hilfe gängiger BPMN 2.0-Modellierungswerzeuge entweder z. B. durch einen Angehörigen, die Rettungszentrale etc. erfolgen. Auch können diese ihren unterschiedlichen Anforderungen entsprechend jeweils unterschiedliche Prozesse spezifizieren und bei JESPA registrieren. Da der Sensorchip keine weitere Logik zur Verarbeitung der Daten enthält, kann ein Dritter (z. B. die Rettungszentrale) den in Abb. 20.2 dargestellten Verarbeitungsprozess bei der JESPA-Middleware registrieren, welche diesen fortan für alle eintreffenden Kontextdaten ausführt. Der Vorteil dieser Variante ist, dass durch Entkopplung der Verarbeitungslogik von dem Chip Herstellungskosten gesenkt werden können und der Chip für eine Vielzahl weiterer Anwendungsfälle eingesetzt werden kann.

Auch Angehörige oder Nachbarn als potentielle Ersthelfer könnten alarmiert werden, sobald die ältere Person stürzt. Anstatt spezieller Hardware brauchen diese z. B. lediglich ein Smartphone, auf welchem eine Workflow Engine läuft. Eine BPMN-basierte Kollaborationsbeschreibung (Abb. 20.3) detailliert die Verarbeitungsschritte, die vom Smartphone bzw. JESPA ausgeführt werden sollen. In unserem Ansatz interpretiert die Workflow Engine das Diagramm, führt jedoch lediglich den Konsumentenprozess (oberer Pool) aus. Der zweite Pool beschreibt die Verarbeitungsschritte, die JESPA bei Eintreten eines bestimmten Ereignisses ausführen soll. Das Ereignis wird durch einen Ausdruck in der *Esper Event Query Language* [24] beschrieben, wodurch mittels CEP komplexe Kontextkonstellationen detek-

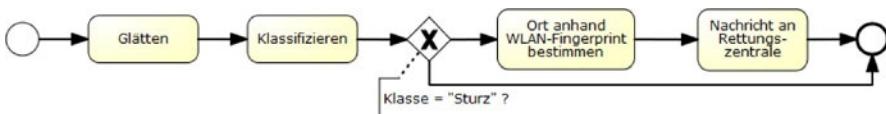


Abb. 20.2 Produzentenseitige Verarbeitungsvorschrift für JESPA

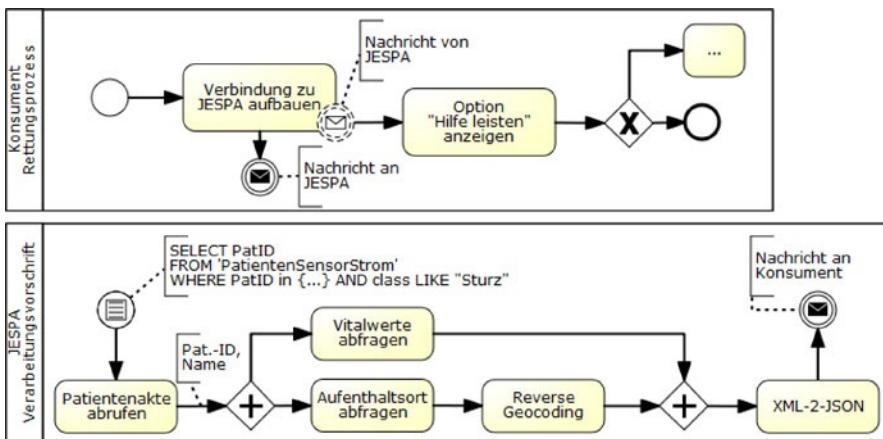


Abb. 20.3 Konsumentenseitige Verarbeitungsvorschrift für JESPA

tiert werden können. Dieser zweite Pool wird als Inhalt einer Registrierungsnachricht an JESPA geschickt und weist die Middleware an, eine Reihe von Aktivitäten auszuführen, sobald ein Sturz festgestellt wurde.

## 20.5 Zusammenfassung

Im Rahmen dieses Artikels wurden Ansätze zur Integration von Kontextinformationen in Smart Applications bzw. Workflows sowie deren Unterstützung durch geeignete generische Systemkomponenten diskutiert. Ausgehend von der Frage, wie Kontextinformationen für die Integration bereitgestellt werden können, wurden eine Reihe existierender Ansätze hinsichtlich ihrer dabei eingenommenen Perspektive (Produzenten vs. Konsumenten) sowie der Verteilungsoptionen von Daten und Verarbeitungsfunktionen untersucht. Daran anschließend wurden Möglichkeiten der Integration diskutiert und anhand bestehender Systemunterstützungen mögliche Probleme aufgezeigt. Wesentlich ist hierbei die geringe Flexibilität der Prozessausführung aus Sicht der Konsumenten, da diese sich meist eng an ein bestimmtes Toolkit oder Framework koppeln müssen, keinen Einfluss auf die Verarbeitungsschritte haben und ggf. in der Umgebung (an anderem Ort) verteilte Ressourcen nicht nutzen können. Auch wird der Wechsel der Ausführungsplattform oder Domäne erschwert, da die Integration eine Anpassung der Infrastruktur verlangt. Dadurch motiviert, wurde ein im Rahmen des JESPA-Projektes neu entwickelter Mediator vorgestellt, der individuelle Verarbeitungsschritte auf Kontextereignissen ausführen kann, die sowohl von Produzenten als auch von Konsumenten durch BPMN-basierte Prozesse vorgegeben sind. Dieser Ansatz steht nicht in Konkurrenz zu anderen, sondern soll eine konzeptionelle Ergänzung zur Erleichterung der Integration von Kontextinformationen bieten. Das ist insbesondere auch für mobile Systeme (und deren

Anwendungen) wichtig, da die Funktionalitäten des Mediators beliebig auch in der Infrastruktur verteilt werden können, was wiederum die (begrenzten) Ressourcen mobiler Geräte schont.

## Literaturverzeichnis

1. Meyer G, Främling K, Holmström J (2009) Intelligent Products: A survey. *Computers in Industry*, 60:137–148
2. Merriam-Webster Online Dictionary (2011) [www.m-w.com](http://www.m-w.com) 26. Oktober 2011
3. Abbasi A, Shaikh Z (2009) A Conceptual Framework for Smart Workflow Management, Int.Conf. on Information Management and Engineering. *IEEE Comp Soc*, pp 574–578
4. Dey A (2001) Understanding and Using Context, Personal Ubiquitous Computing, Springer-Verlag 5:4–7
5. Kofod-Petersen A, Cassens J (2006) Using Activity Theory to Model Context Awareness. In: TR Roth-Berghofer, S Schulz and DB Leake (eds) Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers, *LNCS*, Springer, Edinburgh, 3946:1–17.
6. Kogent Solutions Inc. (2007) Java Server Programming (J2EE 1.4) Black Book (2007 Platinum Edition), Dreamtech Press
7. Bussmann S, Schild K (2000) Self-organizing manufacturing control: an industrial application of agent technology, Fourth International Conference on MultiAgent Systems, pp 87–94
8. Wieland M, Nicklas D, Leymann F (2008) Managing Technical Processes Using Smart Workflows, ServiceWave, pp 287–298
9. EPCglobal Inc. (2011) The Application Level Events (ALE) Spec., Version 1.1.1
10. Schmidt A, van Laerhoven K (2001) How to build smart appliances?, Personal Communications, *IEEE* 8:66–71
11. Riva O (2006) Contory: A Middleware for the Provisioning of Context Information on Smart Phones, *Middleware*, Springer Berlin/Heidelberg 4290:219–239
12. Dey A, Salber D, Abowd G (2001) A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interact J* 16:97–166
13. Project Homepage SFB 627 (2011) [www.nexus.uni-stuttgart.de](http://www.nexus.uni-stuttgart.de) 26. Oktober 2011, Nexus Project
14. Business Process Management Initiative (2011) [www.bpmn.org](http://www.bpmn.org) 26. Oktober 2011
15. Li F, Sehic S, Dustdar S (2010) COPAL: An adaptive approach to context provisioning, In Proceedings of the 6th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)
16. Ardissono L, Furnari R, Goy A, Petrone G, Segnan M (2007) A framework for the management of context-aware workflow systems, Proc. of WEBIST 2007 – 3rd International Conference on Web Information Systems and Technologies, Spain, pp 80–87
17. Urbanski S, Becker C, Rothermel K (2006) Sentient processes – process-based applications in pervasive computing, Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 4:608–611
18. Wieland M, Kopp O, Nicklas D, Leymann F (2007) Towards Context-Aware Workflows, In Proc. of the CAiSEA 07 Workshop, Trondheim, Norway, Tapir Academic Press
19. Bade D, Lamersdorf W (2009) An Agent-based Event Processing Middleware for Sensor Networks and RFID Systems, Computer Journal, Special Issue on Agent Technologies for Sensor Networks
20. Luckham D (2002) The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems, Addison-Wesley Professional
21. Braubach L (2007) Architekturen und Methoden zur Entwicklung verteilter agentenorientierter Softwaresysteme, Universität Hamburg, Dissertation

22. Pokahr A, Braubach L (2011) Active Components: A Software Paradigm for Distributed Systems. In: Proc IEEE/WIC/ACM Int Conf on Intelligent Agent Technology (IAT)
23. Activiti.org (2011) Business Process Management Platform. [www.activiti.org](http://www.activiti.org) 26. Oktober 2011
24. EsperTech Inc. (2011) Esper – Event Stream Intelligence. [esper.codehaus.org](http://esper.codehaus.org) 26. Oktober 2011

---

# Kapitel 21

## Die Anwendung denkt mit auf Schritt und Tritt

Kurt Geihs

**Zusammenfassung** Damit sich Anwendungen auf mobilen Geräten „smart“ verhalten können, müssen sie sich an den sich verändernden Kontext des mobilen Benutzers anpassen können. Das Ziel der Anpassungsfähigkeit ist, dem Benutzer in jeder Situation die dafür optimale Unterstützung zu liefern. Geeignete Sensoren zur Erfassung der Umgebung sind in vielen mobilen Geräten bereits vorhanden, jedoch ist die Erstellung von adaptiven Anwendungen, die solche Sensoren nutzen und sich automatisch anpassen, eine sehr komplexe Aufgabe. In diesem Beitrag analysieren wir zunächst die unterschiedlichen Adoptionsanforderungen mobiler Anwendungen und zeigen somit ihre besonderen technischen Herausforderungen auf. Dann stellen wir eine konkrete Entwicklungsmethodik für adaptive mobile Anwendungen vor. Die Evaluation dieser Methodik anhand mehrerer Demonstratoren hat ihre Wirksamkeit gezeigt, aber auch neue, teilweise interdisziplinäre Forschungsfragen aufgeworfen.

### 21.1 Einleitung

Die zunehmende Akzeptanz und Verbreitung leistungsfähiger Smartphones, in die unterschiedliche Sensoren eingebaut sind, führen zu neuen mobilen Anwendungen, welche den nomadischen Benutzer begleiten und durch eine Vielzahl von Funktionen und Informationsangeboten unterstützen. Handelsübliche Smartphones besitzen bereits bis zu zehn integrierte Sensoren. Beispiele für die eingebaute Sensorik sind Umgebungshelligkeit, GPS, Beschleunigung, Mikrofon, Videokamera, Kompass, Gyroskop und Abstandsmessung. Manche dieser Sensoren sind gleich doppelt

---

Kurt Geihs  
Universität Kassel, Fachbereich Elektrotechnik/Informatik,  
E-mail: geihs@uni-kassel.de

vorhanden, z. B. Mikrofone und Videokameras. Bisher werden diese Sensoren vorwiegend zur Realisierung elementarer Funktionen wie Telefonieren, Navigation und Anpassung der Bildschirmhelligkeit verwendet.

Offensichtlich ergibt sich aber aus der Kombination von leistungsfähiger, mobiler Rechnerplattform und vielfältiger Sensorik ein enormes Potential für die Gestaltung neuer Anwendungen (Apps), wenn diese nicht nur Kontextdaten erfassen, sondern auch auf das aus dem Kontext abgeleitete Wissen unter Ausnutzung weiterer Informationsquellen wie Kalender oder Kontakte adaptiv reagieren können. Erste vorsichtige Schritte in diese Richtung sind bereits in einigen Apps erkennbar, beispielsweise in Form von adaptiven Aufgabenlisten und Terminverwaltungsprogrammen. Unter dem Begriff Kontext verstehen wir im Folgenden gemäß der Definition von [1]: *Kontext ist jede Information, die zur Charakterisierung der Situation einer Entität genutzt werden kann*. Der Kontext einer Anwendung umfasst also Angaben zur Umgebung eines Benutzers (z. B. Helligkeit, Temperatur, Geschwindigkeit, Vorhandensein bestimmter Objekte oder Dienste), zum Zustand des mobilen Computers, auf dem die App abläuft (z. B. Batteriekapazität, Übertragungsbandbreite, Prozessorauslastung) sowie zu den Aktivitäten und Präferenzen des Benutzers (z. B. aktuelle Prioritäten, Kalenderdaten, Adressbuch).

Es ist folglich nur konsequent den Schluss zu ziehen, dass zukünftige Smart Mobile Apps das „Wissen der Sensoren“ ausnutzen werden, um dem Benutzer automatisch einen seiner Situation und seinen Präferenzen angepassten Dienst zu liefern. Mit anderen Worten: Eine Mobile App ist dann „smart“, wenn sie auf Schritt und Tritt mitdenkt und sich zur Laufzeit an die dynamisch veränderliche Umgebung des mobilen Benutzers reaktiv oder proaktiv adaptiert.

Das wirft eine Reihe von komplexen Entwurfsentscheidungen und Fragen für den Entwickler von adaptiven Smart Mobile Apps auf: Welche Ziele werden mit der Adaption verfolgt? Welche Art von Adaption ist somit erforderlich? Welche Varianten sind für eine Anwendung von Interesse? Wie werden die Adoptionslogik und Zielfunktion spezifiziert? Wie sieht es mit der Benutzbarkeit und Robustheit solcher Anwendungen aus? Wie können externe Dienste in die Adoptionslogik einbezogen werden? Die Liste der offenen Fragen ist damit sicher nicht abgeschlossen. Eine systematische softwaretechnische Entwicklungsmethode und passende Laufzeitinfrastruktur für adaptive, Kontext-sensitive Smart Mobile Apps sind dringend erforderlich.

In diesem Beitrag diskutieren wir zunächst die Anforderungen an dynamische Anwendungsadaption und danach unterschiedliche Adoptionsmechanismen und ihre Anwendung anhand von konkreten Anwendungsfällen. Anschließend erläutern wir, wie das von uns mitentwickelte Rahmenwerk MUSIC diese Anforderungen umsetzt und damit die Entwicklung adaptiver Smart Mobile Apps beträchtlich erleichtert. Das MUSIC-Rahmenwerk ist das Resultat eines großen europäischen Projekts und für Android und Windows Mobile frei verfügbar. Im Rahmen des Projekts und danach wurde eine Reihe von adaptiven Anwendungen auf Basis von MUSIC realisiert und evaluiert. Die daraus gewonnenen Erkenntnisse werden gegen Ende dieses Beitrags präsentiert.

## 21.2 Anforderungen

Um die unterschiedlichen Anforderungen und Möglichkeiten der dynamischen Adaption zu erläutern, wählen wir ein durchgängiges Anwendungsbeispiel: die App *Meet-U*. Es handelt sich dabei um eine kontextsensitive, adaptive mobile Anwendung, die nach vorgegebenen Benutzeranforderungen entworfen und prototypisch realisiert wurde. Sie dient als „Werkstück“ zur Illustration und Evaluation verschiedener Adaptionsmechanismen. Für die in diesem Beitrag angestrebte Darstellung der gesamten Bandbreite von Adaptivität in mobilen Anwendungen erweitern wir den aktuell existierenden Meet-U-Prototyp gedanklich um weitere Funktionen und Eigenschaften. Alle diese Erweiterungen wurden im Rahmen des MUSIC-Projekts in anderen Anwendungen bereits eingesetzt und erprobt; hier übertragen wir diese Funktionen der Einheitlichkeit und Didaktik wegen in die erweiterte Anwendung Meet-U. Die so vorgenommenen Erweiterungen sind auch für Meet-U sehr sinnvoll, wurden aber noch nicht im vorhandenen Meet-U-Prototyp umgesetzt. Das gehört zu den zukünftigen Arbeiten.

### 21.2.1 Allgemeine Annahmen

Bevor wir die speziellen Anforderungen der Meet-U-Adaptivität erläutern, ist es zum Verständnis wichtig, die allgemeinen Annahmen für unsere Arbeiten an adaptiven Systemen zu konstatieren. Diese sind:

- Nomadische Benutzer mit (batteriebetriebenen) mobilen Geräten werden in ihren alltäglichen Aufgaben und Aktivitäten durch adaptive mobile Anwendungen in unterschiedlichen Umgebungen und Situationen unterstützt.
- Die Mobilität bringt zwangsläufig eine hohe Dynamik mit sich: Die Verfügbarkeit und Qualität von Netzverbindungen schwankt. Die Umgebungsbedingungen verändern sich. Benutzbare Dienste und Objekte erscheinen und verschwinden. Die Präferenzen des Benutzers ändern sich je nach Situation. Und vieles mehr. Diese Dynamik verlangt nach ständiger Anpassung der Anwendung.
- Mobile Anwendungen benutzen nach Möglichkeit externe, über das Internet zugängliche Dienste zur Verbesserung ihrer Funktionalität und Leistung.
- Mobile Anwendungen kooperieren je nach Verfügbarkeit und Erreichbarkeit mit externen Geräten, um dem Benutzer eine verbesserte Funktionalität und mehr Komfort zu bieten.

Diese allgemeinen Annahmen liegen allen unseren Arbeiten zugrunde, die wir mit diesem Beitrag ansprechen. Daneben gelten natürlich bewährte Prinzipien der Softwareentwicklung, wie Modularität, Wiederverwendbarkeit und Erweiterbarkeit.

## 21.2.2 Spezielle Anforderungen

Die Smart Mobile App Meet-U unterstützt den Benutzer in allen Situationen des Anwendungsszenarios „Treffen mit Freunden bei einer Veranstaltung“ und passt ihre Funktionalität und Darstellung automatisch an den jeweiligen Kontext an. Sie unterstützt somit den Benutzer bei der sozialen Vernetzung mit Freunden im Rahmen von privaten oder öffentlichen Veranstaltungen wie Konzerte und Kongresse. Die Anwendung deckt den kompletten Zyklus ab, von der Planung gemeinsamer Veranstaltungsbesuche, der Navigation und Fahrplanauskunft während der Fahrt zur Veranstaltung, bis hin zur ortsabhängigen Nutzung von Indoor-Navigationshilfen, dem Anbinden von und Zugang zu Hilfsdiensten der Veranstalter (z. B. Programm-informationen und Eintrittskartenverkauf) sowie dem Austauschen von Bildern und Videos zwischen den Freunden, die an der Veranstaltung teilnehmen.

In Reaktion auf Änderungen des Benutzerkontexts wird Meet-U durch die Adaptions-Middleware automatisch der aktuellen Situation angepasst. Im Planungsmodus unterstützt Meet-U den Benutzer, passende Veranstaltungen zu finden, die seinen eigenen und den Interessen seines Freundeskreises entsprechen. Dazu werden externe Dienste mit Veranstaltungsanzeigen und Soziale Netzwerke eingebunden, sofern verfügbar; auch können private Veranstaltungen manuell eingetragen werden. Sind keine externen Dienste zugreifbar, weil temporär kein Internet-Zugang besteht, wechselt Meet-U automatisch in den Offline-Modus, in dem nur bestimmte lokale Eingaben möglich sind.

Macht sich der Benutzer auf den Weg zu einer vereinbarten Veranstaltung, versetzt die Middleware die Anwendung in den Unterwegsmodus, um den Benutzer zum Veranstaltungsort zu führen: Eingebunden werden nun verfügbare Navigationsdienste (Outdoor und Indoor), verschiedene Fahrplaninformationsdienste und ggf. externe Ausgabegeräte wie der Bildschirm des Navigations- und Entertainmentsystems im Auto.



Abb. 21.1 Varianten von Meet-U

Am Veranstaltungsort angekommen, adaptiert sich Meet-U und präsentiert Zugang zum Kartenverkauf, liefert Informationen zur Veranstaltung und zu den Teilnehmern und erlaubt das spontane Austauschen von Bildern sowie Video- und Audioclip mit den Freunden. Anwendungsparameter wie Auflösung der Photos oder Datenrate werden in Abhängigkeit des Systemzustandes (Speicher, Batterie, Übertragungskapazität) automatisch angepasst.

Abbildung 21.1 zeigt zur Illustration vier verschiedene Varianten der Meet-U-App.

## 21.3 Mechanismen

Die zuvor beschriebenen Varianten und Funktionen von Meet-U illustrieren die verschiedenen Arten von Adaptionen, die wir bei der Gestaltung von intelligenten mobilen Anwendungen berücksichtigen und in unserem Adoptionsrahmenwerk MUSIC implementiert haben. Alle diese Adaptionen finden dynamisch zur Laufzeit als Reaktion auf Kontextänderungen statt [2–4]. In den nachfolgenden Beispielen nehmen wir auf die Beschreibung von Meet-U im vorigen Abschnitt Bezug.

- *Parametrische Adaption:* Veränderung von Parameterwerten der Anwendungskomponenten zur Feinabstimmung der Anwendung. Beispiel: Die Auflösung der von den Freunden empfangenen Bilder wird reduziert, weil die freie Speicherkapazität des mobilen Geräts einen bestimmten Schwellenwert unterschreitet.
- *Kompositionelle Adaption:* Modifikation der Anwendungarchitektur durch Austausch, Hinzufügen oder Entnehmen von Komponenten sowie durch Veränderung der Komponentenbeziehungen. Beispiel: Die Anwendung schaltet vom Planungsmodus in den Unterwegsmodus um.
- *Entdecken und Binden externer Dienste:* Erweiterung der Funktionalität lokaler Anwendungskomponenten durch extern angebotene Dienste, die z. B. über das Internet zugreifbar sind. Dazu gehört auch das Ersetzen bereits angebundener Dienstinstanzen durch „bessere“ Dienstinstanzen. Beispiel: Meet-U bindet einen lokalen, im Gebäude vorhandenen Lokalisierungsdienst in die Navigation ein, um dem Benutzer den Weg zum gewünschten Raum auf einem Grundriss des Gebäudes zu zeigen.
- *Verteilung der Komponenten:* Veränderung der verteilten Zuteilung der Komponenten und Dienste auf die Knoten des verteilten Systems. Beispiel: Die Konvertierung von Videoformaten wird vom mobilen Gerät auf einen leistungsfähigeren Rechner verschoben, falls ein solcher zur Verfügung steht.
- *Adaption des Geräts:* Adaption der Hardware des mobilen Geräts, auf dem die Anwendung läuft. Beispiel: Der lokale Bildschirm des Geräts wird ausgeschaltet, wenn ein größeres externes Display in Reichweite ist und die Ausgabe entsprechend umgelenkt werden kann.

## 21.4 Adoptionsmodell

Wir schlagen eine modellgetriebene Entwicklungsmethodik für Kontext-sensitive, adaptive, mobile Anwendungen vor, bei der die Belange der Anwendungslogik und der Anwendungadaptivität, soweit möglich, im Sinne eines „*Separation of Concerns*“ getrennt werden. Damit wird nicht nur ein klareres Design erreicht, sondern auch die Wiederverwendbarkeit von Adoptionsmodellen und Anwendungskomponenten erleichtert. Im Wesentlichen entspricht unsere Vorgehensweise der im Rahmen des MUSIC-Projekts entwickelten Methodik, wobei wir diese aber um eine intensivere Analysephase ergänzt haben [3].

Im Folgenden konzentrieren wir uns auf die Darstellung des Adoptionsmodells von Meet-U, das die verschiedenen Varianten modelliert, den Bezug der Varianten zu den Kontextparametern definiert und die Zielfunktion für die Adaption in Form einer Utility-Funktion vorgibt. Als Modellierungssprache wurde UML gewählt.

Das (zur besseren Verständlichkeit für diesen Beitrag vereinfachte) Adoptionsmodell zeigt auf oberster Ebene den Komponententyp „Meet-U“ sowie vier verschiedene Realisierungen dieses Komponententyps, welche den zuvor erwähnten Modi entsprechen (siehe Abb. 21.2): *Offline*, *Planning*, *OnTheWay*, und *AtEvent*. Auch die zur Meet-U-App zugehörige Utility-Funktion ist in diesem Modellausschnitt enthalten. Sie bildet den aktuellen Zustand des Benutzerkontexts (z. B. keine Internet-Verbindung oder die Lokation des Benutzers stimmt mit dem Veranstaltungsort überein) auf einen numerischen Wert zwischen Null und Eins ab, so dass der für diesen Kontext vorgesehene Anwendungsmodus den höchsten Wert erhält.

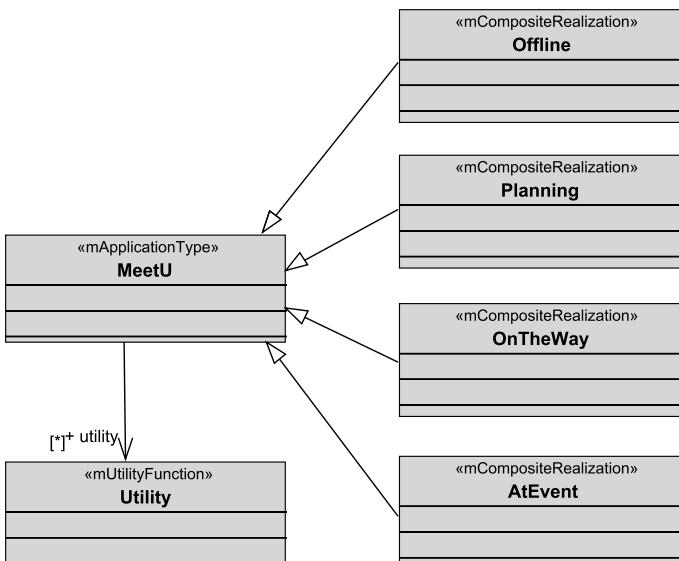


Abb. 21.2 Realisierungsvarianten von Meet-U auf oberster Stufe

Offensichtlich stellt das Modell in Abb. 21.2 eine kompositionelle Adaption dar, wobei der Schwerpunkt hier auf alternativen Funktionalitäten der Anwendung liegt. Die kompositionelle Adaption kann auch zum kontextabhängigen Austausch funktional gleichwertiger Komponenten eingesetzt werden; das ist in diesem Beispiel nicht dargestellt.

Jede dieser Anwendungsvarianten ist aus weiteren Komponenten zusammengesetzt. Dies wird durch den Stereotyp *mCompositeRealization* spezifiziert, im Gegensatz zu einer *mAtomicRealization*, die in dieser Abbildung nicht vorkommt. Zum Beispiel zeigt Abb 21.3 die Zusammensetzung der Realisierung *AtEvent*, die selbst wiederum aus fünf verschiedenen Komponententypen besteht. Falls wiederum mehrere Realisierungen dieser (Sub-)Komponenten vorhanden sind, ergibt sich eine kombinatorische Anzahl weiterer Variationsmöglichkeiten. Das Adoptionsmodell beschreibt also Variationspunkte, an denen zur Laufzeit verschiedene Optionen für die Anwendungskonfiguration ausgewählt werden können. Die rekursive Variation stoppt an dem Punkt, wo für eine Komponente nur noch eine *mAtomicRealization*, d. h. eine Realisierung ohne weitere Unterkomponenten, zur Wahl steht. Der Adoptionsmanager in der Middleware berechnet für jede zulässige Konfiguration die Utility nach der vorgegebenen Utility-Funktion und wählt diejenige aus, welche die höchste Utility besitzt. Somit wird genau die Applikationsvariante ausgewählt, die für die jeweilige Situation am besten passt. Danach setzt der Konfigurationsmanager der Middleware diese Adaption um.

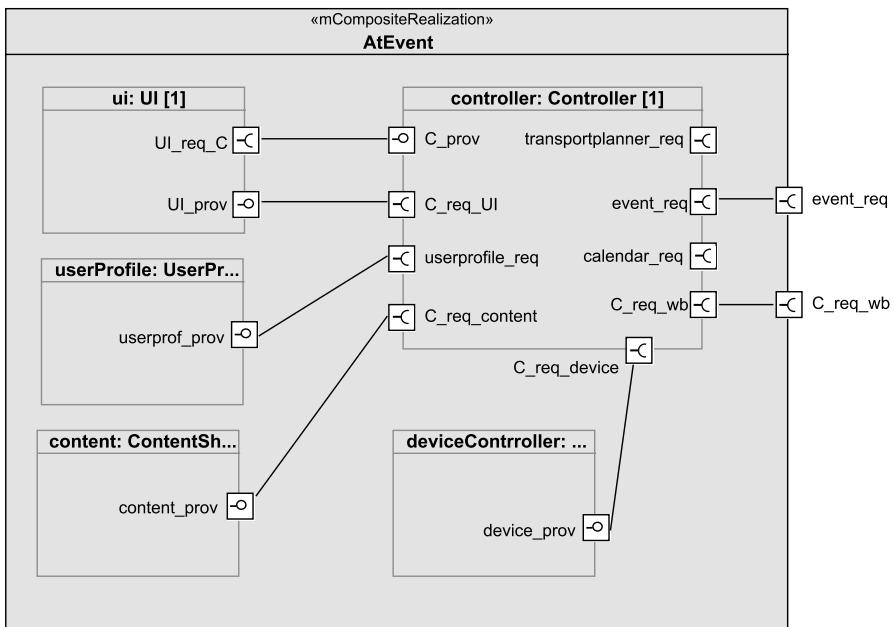


Abb. 21.3 Dekomposition der Realisierung *AtEvent*

Die Komponenten im Adoptionsmodell sind durch Ports miteinander verbunden. Eines der interessantesten Merkmale des Adoptionsmodells von MUSIC ist die Möglichkeit, dynamisch entdeckte externe Dienste in ubiquitären Umgebungen zur Laufzeit an die Anwendung anbinden zu können. Ports, an denen externe Dienste angebunden werden können, werden im Modell als unverbundene, nach außen zeigende Ports einer Realisierung dargestellt und mit Angaben zum gewünschten Dienst annotiert. Abbildung 21.3 zeigt zwei solcher Ports: *event\_req* und *C\_req\_wb*. Zum Beispiel besitzt der Port *event\_req* den Typ *EventServiceType* und verlangt nach einem externen Dienst, der Informationen zu der Veranstaltung und Eintrittskarten liefern kann.

Es soll hier explizit betont werden, dass das Adoptionsmodell nur die Architektur und Variabilität der Anwendung wiedergibt. In der aktuellen Version unserer Entwicklungsmethode werden die Anwendungskomponenten selbst nicht modelliert. Darauf wurde ganz bewusst verzichtet, da die Entwicklung von Softwarekomponenten eine orthogonale Aufgabe darstellt, für die erprobte softwaretechnische Unterstützung zur Verfügung steht. Zudem fördert diese Entscheidung das „Separation of Concerns“ und erlaubt den Entwicklern einer Anwendung, ihre jeweils bevorzugte Softwareentwicklungsmethode zu verwenden.

## 21.5 Realisierung

Gemäß der modellgetriebenen Vorgehensweise wird das Adoptionsmodell mit einer Modell-zu-Text-Transformation in Java-Quellcode übersetzt. Dieser Code erlaubt es der MUSIC-Middleware, die verschiedenen, in einem gegebenen Kontext möglichen Varianten der Anwendung zu ermitteln und für jede Variante die Utility zu berechnen und zu vergleichen. Der Aufwand für die Ermittlung aller Varianten und ihrer Utility ist ein Hauptbestandteil der Adoptionszeit, d.h. der Zeit, die von der Entdeckung einer Kontextänderung bis zum Ende der daraus resultierenden Adaption verstreicht. Diese Zeitspanne hängt ganz wesentlich von der Größe des Adoptionsmodells und der Anzahl der verschiedenen Realisierungen der Komponenten ab.

Die MUSIC-Middleware baut auf dem Komponentenrahmenwerk OSGi [5] auf und ist für mehrere Systemplattformen verfügbar, so auch für Google Android<sup>1</sup>. Die Meet-U-App läuft unter Android auf gängigen Smartphones. Typische Werte für die Adoptionszeit einer mobilen Anwendung mit 10 Varianten auf einem Smartphone mit 1 GHz CPU, 512 MB Speicher und Android 2.1 liegen im Bereich von 3 Sekunden. Ob diese Zeit dem Benutzer als angemessen oder als zu lang erscheint, hängt sicher vom Anwendungsszenario ab. Auch muss hierbei betont werden, dass der MUSIC-Middleware-Prototyp das Ergebnis eines Forschungsprojekts ist und bisher noch nicht explizit auf Performance „getrimmt“ wurde, da zunächst das Adoptionsverhalten im Vordergrund stand.

---

<sup>1</sup> Der Prototyp inklusive Anwendungen ist kostenlos verfügbar im Android Market.

Weitere Informationen zur Implementierung und Benutzung finden sich auf den Seiten des MUSIC-Projekts im Open Source Repository BERLIOS<sup>2</sup>.

## 21.6 Verwandte Arbeiten

Die Entwicklung von Smart Mobile Applications berührt eine Reihe von Forschungsthemen, z. B. Adaptivität, Kontextsensitivität, Robustheit und Benutzbarkeit, zu denen jeweils eine umfangreiche Menge an Vorarbeiten existiert. Im Folgenden präsentieren wir einen kleinen Ausschnitt davon.

Middleware und Entwicklungsumgebungen für mobile, adaptive Systeme wurden in einigen Projekten erforscht. Die Autoren von [1] beschreiben einen sog. *Context Toolkit*, der dem Entwickler einer kontextsensitiven Anwendung Abstraktionen für die Verarbeitung von Kontextdaten zur Verfügung stellt. Dazu gehören beispielsweise Widgets, Interpreter, Aggregatoren und Dienste. Der Context Toolkit konzentriert sich stark auf das Kontextmanagement, während – im Gegensatz zu MUSIC – die Unterstützung der Adaptivität der Anwendung weniger Beachtung findet. Zudem bietet die MUSIC Context Management Middleware moderne Plug-In-Mechanismen für Kontextsensoren, die im Sinne eines „*Context as a Service*“ eine dynamische Anpassung der Sensorik erlauben.

Das Middleware-Rahmenwerk TraX [4] unterstützt die Erstellung ortsbasierter Dienste und Anwendungen und konzentriert sich somit ganz auf die im mobilen Gerät eingebaute Positionssensorik. Die TraX-Middleware zeichnet sich ferner aus durch verschiedene Dienste zur Verarbeitung von Positionsdaten und ein umfangreiches Konzept zur Sicherung der Privatsphäre der Benutzer, deren Positionsdaten erfasst werden. Das MUSIC-Entwicklungsrahmenwerk hingegen ist nicht nur auf einen Sensortyp festgelegt, sondern kann mit beliebigen Sensoren arbeiten, die auch noch zur Laufzeit dynamisch eingebunden werden können.

CARISMA [6] ist eine Peer-to-Peer-Middleware für das Mobile Computing, die im Gegensatz zu MUSIC auf die Anpassung von Diensten der Middleware ausgerichtet ist. Die Adoptionsziele werden regelbasiert spezifiziert. CARISMA unterstützt nicht die dynamische Entdeckung und Einbindung von externen Diensten in Anwendungen, was wiederum in MUSIC eine wichtige Rolle spielt und in Anwendungen wie Meet-U intensiv ausgenutzt wird.

Biegel und Cahill präsentieren in [7] ein Rahmenwerk, dessen Ziel es ist, die Entwicklung kontextsensitiver Anwendungen in mobilen, ad hoc vernetzten Systemen zu unterstützen. Schwerpunkte der Methodik sind dabei die Fusion von Kontextinformation und die Beachtung der Unzuverlässigkeit von Sensordaten.

Alle zuvor genannten Arbeiten adressieren in unterschiedlich starker Form die Notwendigkeit der Anwendungsadaption und einer adäquaten methodischen Entwicklungsunterstützung. Uns sind aber keine Arbeiten bekannt, die in ähnlich umfangreicher und konsistenter Weise wie unser Rahmenwerk die Entwicklung von

---

<sup>2</sup> <http://ist-music.berlios.de/>

adaptiven, kontextsensitiven Anwendungen unterstützen. MUSIC bietet dem Anwendungsentwickler eine Reihe verschiedener Adoptionsmechanismen an, die in einen kohärenten modellgetriebenen Ansatz integriert sind, welcher durch eine umfangreiche Werkzeugkette effektiv unterstützt wird.

## 21.7 Erfahrungen und Ausblick

Damit sich Anwendungen auf mobilen Geräten „smart“ verhalten können, müssen sie sich an den sich dynamisch verändernden Kontext des mobilen Benutzers anpassen können. Das Ziel der Anpassungsfähigkeit ist, dem Benutzer in jeder Situation die dafür optimale Unterstützung zu liefern. Die Entwicklung solcher Anwendungen ist schwierig. Zur Beherrschung der Komplexität ist eine umfangreiche Entwicklungsunterstützung einschließlich Middleware und Werkzeugen dringend erforderlich. Das MUSIC-Entwicklungsrahmenwerk bietet hier eine Lösung für unterschiedliche Zielplattformen an. Es hat seine Tauglichkeit in mehreren Demonstratoren des MUSIC-Projekts unter Beweis gestellt.

Noch aussagekräftiger als die MUSIC-Demonstratoren sind die Erfahrungen mit MUSIC in Folgeprojekten an drei verschiedenen Orten, in denen das Rahmenwerk den Entwicklern neu war. In den anschließenden Befragungen zeigte sich übereinstimmend, dass (a) die Entwickler den Lernaufwand zunächst als sehr hoch einschätzten, (b) das Rahmenwerk nach dieser Lernphase als sehr positiv und effektiv für die Entwicklung mobiler Anwendungen mit komplexem adaptiven Verhalten bewertet wurde und (c) die prototypischen Implementierungen der Middleware und einiger Werkzeuge noch beträchtlichen Raum für Verbesserungen in puncto Stabilität und Benutzeranleitungen bieten. Letzteres ist nach unserer Erfahrung gerade für Ergebnisse von Forschungsprojekten nicht überraschend.

Aus technischer Perspektive sind – zum Teil aus Kapazitätsgründen – einige allgemeine Fragen offen geblieben, an denen in Zukunft gearbeitet wird. Dazu gehören die Skalierbarkeit des Adoptionsmodells und der Adoptionsentscheidung; die Robustheit und Stabilität der Adaption, wenn Fehler während der Rekonfiguration auftreten; die Testbarkeit und Validierung adaptiver Anwendungen; eine erweiterte Behandlung von Zustandstransfers zwischen Komponenten, die über die einfache Lösung im aktuellen Prototyp hinausgeht; die Sicherheit der persönlichen Kontextdaten und Fragen des Vertrauens in adaptive Systeme. Auch hat eine Benutzungsevaluation von Meet-U ergeben, dass der Benutzer in bestimmten Situationen manuell Einfluss auf die Adaption nehmen möchte. Wir arbeiten zurzeit an einer Lösung für diese Anforderung, die nicht nur speziell für die Anwendung Meet-U relevant ist, sondern auch prinzipiell im MUSIC-Rahmenwerk Berücksichtigung finden sollte.

Ein Großteil dieser offenen Fragen wird im Rahmen des VENUS-Projekts an der Universität Kassel untersucht [8]. Hierbei geht es insbesondere um eine interdisziplinäre Gestaltungsmethodik für anpassungsfähige, vernetzte Systeme. Neben den technischen Aspekten werden nicht-technische Aspekte wie Ergonomie, Recht

sowie Vertrauen und Akzeptanz von Beginn an im Entwicklungsprozess beachtet und in technische Vorgaben umgesetzt. Diese Vorgehensweise erscheint uns gerade für Smart Mobile Apps, die eine Fülle benutzerbezogener Kontextinformationen sammeln und verarbeiten, unerlässlich.

## Literaturverzeichnis

1. Dey AK (2001) Understanding and using context. *Personal Ubiquitous Computing* 5(1):4–7
2. Cheng BHC et al (2009) Software engineering for self-adaptive systems: A research roadmap. In *Software Engineering for Self-Adaptive Systems*, Lecture Notes in Computer Science, Springer 5525:1–26
3. Evers C, Hoffmann A, Saur D, Geihs K, Leimeister JM (2011) Ableitung von Anforderungen zum Adoptionsverhalten in ubiquitären adaptiven Anwendungen. In Proc. of Workshop on Self-Organising, Adaptive, Context-Sensitive Distributed Systems (SAKS), ECEASST 37
4. Küpper A, Treu G, Linnhoff-Popien C (2006) TraX: A Device-centric Middleware Framework for Location-based Services. *IEEE Commun Mag* 44(9):114–120
5. The OSGi Alliance. <http://www.osgi.org>. Zugriff 05. Juni 2011
6. Capra L, Emmerich W, Mascolo C, (2003) Carisma: Context-aware reactive middleware system for mobile applications. *IEEE Trans Soft Eng* 29:929–945.
7. Biegel G, Cahill V (2004) A framework for developing mobile, context-aware applications. Proceedings of the 2<sup>nd</sup> IEEE International Conference on Pervasive Computing and Communications, Washington, DC, USA.
8. VENUS Projekt. <http://www.uni-kassel.de/einrichtungen/iteg/venus/startseite.html>. Zugriff 5. Juni 2011
9. Dey AK, Abowd GD, Salber D (2001) A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction* 16.
10. Geihs K (2008) *Selbst-adaptive Software*. Informatik-Spektrum, Springer, 31(2):133–145
11. McKinley PK, Sadjadi SM, Kasten EP, Cheng BHC (2004) Composing Adaptive Software. *IEEE Comput* 37(7):56–64

**Teil VII**

**Technologische Sicht: Workflow &**

**Prozessintegration**

---

# Kapitel 22

## Agile Geschäftsprozesse durch integrierte mobile Kommunikation

Jürgen Totzke und Karl Klug

**Zusammenfassung** Nicht-Routine-Geschäftsprozesse erfordern ein hohes Maß an Interaktion. Deshalb wird Kommunikation zunehmend in die Geschäftsprozesse integriert. Derartig erweiterte Geschäftsprozesse werden Communication Enabled Business Processes (CEBP) genannt. Dabei werden heute vorwiegend Computer Telephony Integration (CTI)-Lösungen für die Kommunikation verwendet. CTI-basierte Lösungen sind jedoch hinsichtlich ihrer Funktionalität begrenzt und relativ statisch. Die flexible Einbindung von mobilen Endgeräten ist unzureichend.

Fortgeschrittene Business Process Modeling (BPM)-Plattformen hingegen erlauben eine wesentlich komfortablere Integration von Kommunikation in die Geschäftsprozesse. Besonders Service Oriented Architecture (SOA)-basiierende Implementierungen von Unified Communication & Collaboration (UCC) erlauben an den Geschäftskontext angepasste Integration von Kommunikationsvorgängen auch im mobilen Umfeld. So wird benutzerzentrische, kontext- und ortsbabhängige Modellierung von Geschäftsprozessen ermöglicht. Außerdem werden der Business Software ohne besonderen Integrationsaufwand die gesamte Funktionalität von UCC und besonders deren mobile Erweiterungen zur Verfügung gestellt.

Im Rahmen eines durchgeführten Innovationsprojektes wurden mobile Endgeräte über einen UCC-Kommunikationsserver an ein Business-Softwaresystem angebunden. Neben der Service-orientierten Anbindung der Kommunikationsdienste stand dabei vor allem eine Frage im Vordergrund: Wie wird sich die Einbindung von mobilen Endgeräten mit mobiler Erreichbarkeit und Lokalisations-Information auf die Agilität der durchlaufenden Geschäftsprozesse auswirken?

Das Ergebnis ist unerwartet positiv, weil klare Effizienzsteigerungen der betrachteten Geschäftsprozesse feststellbar sind; Eine Größenordnung von 35 Prozent Effi-

---

Jürgen Totzke  
Siemens Enterprise Communications GmbH & Co. KG, München,  
E-mail: juergen.totzke@siemens-enterprise.com

Karl Klug  
Siemens Enterprise Communications GmbH & Co. KG, München,  
E-mail: karl.klug@siemens-enterprise.com

zienzsteigerung ist dabei durchaus realistisch. Der daraus kalkulierbare Return on Investment (ROI) rechtfertigt mögliche Investitionen in derartige Plattformen und Systeme. Der im Innovationsprojekt realisierte Prototyp hat bei den Betrachtern sofort vielfältige Ideen für neuartige Anwendungen ausgelöst, was ein erhebliches Potenzial für mobile Business-Applikationen erwarten lässt.

## 22.1 Einführung

Bei Smart Mobile Apps denkt man vorrangig an in sich abgeschlossene Applikationen, die aus einem zentralen Repository (AppStore) heruntergeladen werden und auf dem mobilen Endgerät ablauffähig sind. Einige Anwendungen benötigen dabei keine Informationen aus Hintergrundanwendungen, andere greifen über mobile Datenverbindungen auf diese Informationen zu. Derartige Applikationen sind vorwiegend für den privaten und teilweise für den geschäftlichen Gebrauch verfügbar. In diesem Buch werden Aspekte von Business Apps für mobile Geschäftsprozesse behandelt. Naheliegend sind hierbei mobile Erweiterungen für bestehende Geschäftsprozesse, die Smartphones nur als entfernte Endgeräte nutzen. Der in diesem Beitrag bereits praktisch erprobte Ansatz stützt sich auf eine Kombination von Fähigkeiten mobiler Endgeräte und der sich bereits in breiter Anwendung befindlichen mobilen Unternehmens-Sprachkommunikation im Rahmen von Geschäftsprozessen.

Der Beitrag ist wie folgt gegliedert. Bevor das eigentliche Fallbeispiel behandelt wird, werden Entwicklungen und Trends von Komponenten aus den Bereichen der Unternehmensinformations- und Kommunikationstechnologien (IKT) dargestellt. Damit soll das Grundverständnis für Integrationsszenarien von Kommunikation in Geschäftsprozesse geschaffen und gezeigt werden, dass das technologische Umfeld für mobile Geschäftsprozesse mit integrierter Kommunikation vorbereitet ist. Prinzipiell werden zwei Integrationsformen von (mobiler) Kommunikation in Geschäftsprozesse vorgestellt, bevor das konkrete Fallbeispiel besprochen und bewertet wird.

## 22.2 Konvergierende Entwicklungen und Trends

Wesentliche Trends in der Telekommunikation und Informationstechnologie führen zur zunehmenden Verbreitung von Smartphones und Tablet-PCs sowie dafür geeigneten Anwendungen für den privaten und geschäftlichen Bereich. Die Entwicklung der Unternehmenskommunikationslösungen, der Unternehmens-IT und der Mobilfunknetze und deren Konvergenzen führen zwangsläufig zu Potenzialen für neue mobile Anwendungen im geschäftlichen Bereich und eröffnen den Marktteilnehmern wie Business-Software-Herstellern und Herstellern von Kommunikationslösungen, aber auch vielen Herstellern spezialisierter Querschnittsfunktionen neue

Geschäftsmöglichkeiten. Es ist zu erwarten, dass aus diesen Lösungen mit mobilen Erweiterungen Investitionsdruck auf die Unternehmen entstehen wird, um deren Wettbewerbsfähigkeit zu erhalten oder auszubauen. Deshalb ist die Kenntnis dieser konvergenten Entwicklungen und Trends für heutige und zukünftige Investitionsentscheidungen wichtig, um sich einen ebenfalls evolutionären Zugang zu mobilen Geschäftsanwendungen zu ebnen.

### **22.2.1 Mobile Endgeräte**

Private Konsumenten haben mobile Endgeräte wie Smartphones und Tablet-PCs zunehmend in ihren Alltag integriert und den Innovationsdruck auf die Hersteller enorm erhöht. Zunächst führte dies zur Nutzung von weiteren Medien auf diesen Geräten, wie die Musikwiedergabe oder die Aufnahme von Fotos. Im nächsten Schritt führte die kostengünstige Anbindung an das Mobile Internet und die zunehmende Präsenz von privaten Konsumenten in sozialen Netzen zu immer leistungsfähigeren mobilen Endgeräten und zur Nutzung von kleineren Applikationen. Und zuletzt reichte die Verwendung von Ziffernblock, Minitastaturen und Zeigerfunktionen für derartige Anwendungen nicht mehr aus und führte zur Entwicklung von neuen Benutzeroberflächen wie Touch-Screens mit haptischem oder optischem Feedback und Gestenerkennung. Die Bereitstellung von mobiler Navigations-Software führte zur Integration von Positionierungs-Informationen auf den mobilen Kommunikationsendgeräten und ermöglicht den Applikationsanbietern und Mobilfunkbetreibern neuartige Geschäftsmodelle. Die meisten Anwendungen zur vorwiegend privaten Nutzung werden den Benutzern über betriebssystem- oder herstellerspezifische Applikations-Marktplätze zum Herunterladen angeboten. Einige dieser Anwendungen werden auch im geschäftlichen Umfeld genutzt, z. B. Soziale Netzwerke wie Twitter, Yammer, LinkedIn etc. Weit verbreitet sind außerdem noch die sogenannten Personal Information Management (PIM)-Anwendungen für den Zugriff auf berufliche Daten wie Kontakte, Kalender und E-Mail, die mit den entsprechenden Backend-Systemen im Firmen-Intranet synchronisiert werden.

### **22.2.2 Business-Software**

Heute kann kein Unternehmen vom Großunternehmen bis zum Mittelstandsunternehmen mehr auf den Einsatz von Business-Software verzichten. Anfängliche Systeme waren monolithische Softwareeinheiten, die in Form von Siloanwendungen im Bereich Personal, Kunden- und Lieferantenbeziehungen, interne Ressourcenplanung usw. nebeneinander genutzt wurden. Hier kamen Business-Software-Anbieter zum Zuge, die mit ihrer Softwarelösung den individuellen Kundenerfordernissen am besten entsprochen haben. Dies führte auch zu einer großen Anbietervielfalt der verwendeten Software im Unternehmen. Mit dem Aufkommen

neuer Software-Architekturen, besonders der Service-orientierten (SOA) Software-Plattformen der IT-Hersteller, konnten neue Business-Software-Anteile mit dieser Technologie realisiert werden und alte Software-Bestandteile mit sogenannten Web-Service-Schnittstellen geeignet gekapselt werden. Das Ergebnis war die Möglichkeit, die Informationssilos aufzubrechen und Anwendungen überdisziplinarisch als auch herstellerübergreifend zu realisieren. Aber auch die SOA-basierte Business-Software kann die zunehmende Flexibilitätserwartung und kurzfristigen den Änderungs- und Anpassungsbedarf von Unternehmen im globalen Wettbewerb zunehmend nur unzureichend erfüllen. Deshalb entwickeln die Hersteller von Business Software ihre SOA-basierten Plattformen in Richtung Service Component Architectures (SCA) weiter. Dabei werden die Geschäftsprozesse mit geeigneten Werkzeugen, sogenannten Business Process Modeling (BPM) Tools aus Komponenten orchestriert. Dies ermöglicht neue Geschäftsmodelle wie z. B. Partnermodelle für Business-Software-Hersteller, die sich auf ihre Kernkompetenzen konzentrieren möchten, aber gerne die Anwendungsvielfalt wie branchenspezifische Anwendungen realisiert sehen. Diese flexiblen Architekturen erlauben aber auch vielfältige Möglichkeiten zur Integration von Mobilitätserweiterungen. Auch BPM wird nicht das Ende der Entwicklung darstellen, es werden bereits Plattformen der nächsten Generation (BPM 2.0) diskutiert, die den zunehmenden individuellen, fallbasierten Ablauf von Geschäftsprozessen ermöglichen sollen.

### **22.2.3 Kommunikationssysteme**

Parallel zu den öffentlichen Integrated Switched Digital Networks (ISDN) wurden in den privaten Firmennetzen entsprechende Systeme mit zunächst proprietären Signalisierungs-Protokollen auf Basis der Time Division Multiplexing (TDM)-Technologie als digitale Vermittlungssysteme entwickelt. Der Wettbewerb wurde über die Leistungsmerkmal-Vielfalt ausgetragen. Ende-zu-Ende-Interoperabilität mit Firmennetzwerken wurde über die in ECMA standardisierten Q-SIG-Protokolle für eine gemeinsame Untermenge dieser Leistungsmerkmale erreicht. Interoperabilität mit den öffentlichen Netzen wurde durch Interworking mit ISDN erzielt, wobei die proprietären Leistungsmerkmal-Signalisierungen transparent über die öffentlichen Netze transportiert wurden.

Über den Zwischenschritt der Asynchronous Transfer Mode (ATM)-Technologie erfolgten erste Erfahrungen der Übermittlung von echtzeitkritischer Sprache über Paketnetze, zunächst mit fester Paketlänge. Nachdem die ATM-Technologie in Firmennetzen schnell von der IP-Technologie abgelöst wurde, erfolgte die Übermittlung von Sprache mit Paketen variabler Paketlängen. Damit war ein konvergentes Netz für den Transport von Daten, Sprache und Video geschaffen. Dies war auch die Technologiebasis für die sogenannte „Converged PBX“, bei der die Hersteller ihre zum Teil proprietären Signalisierungsverfahren mit VoIP verknüpfen. Die Notwendigkeit einer eigenen TDM-Infrastruktur bis auf Gateway-Einrichtungen zum öffentlichen Festnetz oder Mobilfunknetz war damit hinfällig. Mit dem mobilen In-

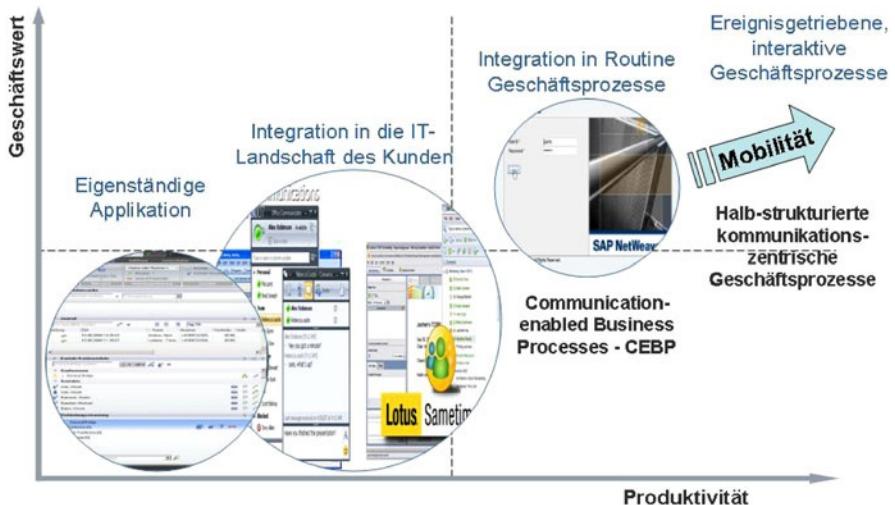
ternet und geeigneten Sicherheitskonzepten wie Virtual Private Networks (VPN) kann dieses konvergente Firmennetzwerk einfach in die mobile Domäne erweitert werden.

Anstelle der Leistungsmerkmale von TDM-basierten oder konvergenten Firmenkommunikationsnetzen setzte sich der Session Initiation Protocol (SIP)-Standard zum Signalisierungsprotokoll der Wahl auch in Firmennetzen durch. Nur einige wenige Leistungsmerkmale aus den Q-SIG-basierten Netzen wurden auch für SIP spezifiziert und implementiert. Neue Paradigmen im SIP-Umfeld, besonders die präsenzbasierte (Presence-) Kommunikation, führten auch zu abweichenden Implementierungen bekannter Leistungsmerkmale. Ein prominentes Beispiel hierfür ist der Rückruf bei besetzt. In Soft-BPX Lösungen wird dies durch Beobachtung des Media-Präsenzstatus Festnetztelefon des anderen Teilnehmers und Alarmierung des rufenden Teilnehmers erreicht (Tell-me-When ...). Auch die aus den TDM-Systemen bekannten Unified Messaging (UM)-Systeme entwickelten sich durch die Verwendung weiterer Medien neben der Sprache wie Video und dem Präsenz-Paradigma zum einheitlichen Kommunikationsmodell Unified Communication (UC) bzw. mit mobilen Erweiterungen und Bereitstellung von mobilen Clients zu Mobile Unified Communication (MUC) weiter.

#### ***22.2.4 Unified Communication (UC)***

Zunehmende Mobilitätsanforderungen, Integration in Groupware Software und Geschäftsprozesse zeigen den auf proprietären Signalisierungs-Verfahren basierten Converged-Systemen zunehmend Grenzen auf. Die bekannte Computer Telefonie Integration (CTI)/CSTA ist geräteorientiert und für personenorientierte Kommunikation nur bedingt transparent. Hier sind SIP-Standard-basierte UC-Systeme klar im Vorteil, da die personenbezogene Adressierung durch den sogenannten One Number Service (ONS) inklusive Mobilitätserweiterungen bereits integriert ist. Außerdem erlaubt eine Standard-basierte Lösung den anwendenden Firmen eine Multi-vendor Strategie bezüglich der Endgeräte und der Systemkomponenten wie Video.

Der Geschäftswert von UC-Lösungen ist auf verschiedene Ebenen gegliedert. Auf der untersten Ebene erfolgt die Einführung der VoIP-Infrastruktur in Verbindung einer UC-Anwendung. Dies können bereits Converged PBX-Systeme leisten. Auch die Erweiterung der Produktivität durch Integration von Video oder Kollaborations-Lösungen ist mit diesen Lösungen darstellbar. Allerdings ist dies bereits mit Bindung an einen Hersteller verbunden. Multi-vendor-Strategien auf dieser Ebene sind bereits unmöglich. Auf den höheren Ebenen des Geschäftswertes mit Integration in Groupware oder die Business Software Integration, also die Integration von Kommunikation in die Geschäftsprozesse (Communication Enabled Business Processes – CEBP), ist neben der Standardkonformität auch die Art der Realisierung der UC-Lösung erheblich. Service Oriented Architecture (SOA)-basierte Implementierungen erlauben hier die Integration mit unterschiedlichster Integrationstiefe und umgekehrt die Paketierung von Services zu sogenannten Composites, die



**Abb. 22.1** Integrationsvarianten von Unified Communication in die Unternehmens-IT

sich für die Orchestrierung auf sogenannten Business Process Modeling (BPM)-Plattformen der wichtigsten Hersteller von Business Software besser eignen. Weitergehende Untersuchungen in der Fortschreibung dieser BPM-Systeme in Richtung halb-strukturierter Kommunikation und Kollaboration auf Basis von Event Driven Architectures (EDA) als zukünftige Enterprise Architecture (EA) werden derzeit untersucht [3]. Zunehmende Mobilität wird durch ihre Anforderungen und neuen Möglichkeiten diesen Trend noch verstärken. Deshalb ist es wichtig, die Investitionsentscheidung von UC-Systemen nicht nur auf die Infrastruktur-Einsparungen zu reduzieren, weil damit weitergehende Integrationen auf der Ebene der Geschäftsprozesse, und dem damit verbundenen, erhöhten Geschäftswert von UC oder Mobile UC (MUC) erschwert oder gar unmöglich gemacht werden. Nachteilig kann sich auch die Auswahl von UC von Herstellern auswirken, die ihre Lösung ausschließlich an ihre sonstigen Enterprise-IT-Lösungen binden (Snap-in-Strategie), aber bei Weitem nicht das gesamte Spektrum der Unternehmens-IT abdecken oder für Lösungskomponenten keine oder nur wenig Akzeptanz beim Kunden finden. Dies gilt insbesondere auch für unzureichende oder fehlende Mobilitätserweiterungen der UC-Lösungen dieser Hersteller.

## 22.2.5 IT-Systeme

IT-Systeme folgten in der Vergangenheit mehrfach wechselnden Trends von zentralen und dezentralen Architekturen. Die zunehmende Endgeräte-Vielfalt und die damit zunehmende Komplexität des Software-Lifecycle-Managements bei den Endgeräten, besonders hinsichtlich der damit verbundenen Sicherheitsaspekte und der

Durchsetzung von firmenspezifischen Sicherheits-Policies, führten teilweise zu einer Renaissance von Client-Server-basierten Anwendungen bis hin zur Desktop-Virtualisierung. Client-Server-basierte Anwendungen wurden mit zunehmender Verbreitung des World Wide Web als Web-Applikationen realisiert. Zunehmende Ansprüche der Anwender hinsichtlich der graphischen Oberflächen werden durch sogenannte Rich Internet Applications (RIA) mit interaktiven Komponenten adressiert. Diese beinhalten meist auch die Unterstützung von mobilen Endgeräten.

Zunehmende Konzentration auf das Kerngeschäft führten zunehmend zu Outsourcing von IT-Dienstleistungen. Zeitgleich rasch fortschreitende Entwicklung von Virtualisierungs-Softwareplattformen eröffnet neue Betreiber- und Geschäftsmodelle wie Hosted Services. Nahezu alle softwarebasierten Systeme gängiger Betriebssysteme sind heute virtualisierbar. Der Trend Green-IT fördert diesen Trend, da virtualisierte Software auf Standard-Servern ablauffähig und bei Niederlastzeiten fast beliebig auf den Servern relokierbar ist und nicht beschäftigte Server entsprechend abgeschaltet werden können.

Virtualisierte Software-Systeme befähigen diese Anwendungen, auch in der Cloud ablauffähig zu sein. Im oder außerhalb des Unternehmens installierte Private, Hybrid oder Public Clouds adressieren die heute vorhandenen Sicherheitskriterien oder -einschätzungen der Unternehmens-IT. Besonders Public Clouds eröffnen neue Geschäftsmodelle wie Infrastructure-, Platform-, Software as a Service (IaaS, PaaS, SaaS). Das wesentliche Potenzial von Cloud-basierter Software, neben den Infrastruktureinsparungen und eingebauter Skalierbarkeit und Verfügbarkeit, liegt allerdings in der Möglichkeit zur Verknüpfung von Anwendungen zu sogenannten Composite Applications oder aufkommenden Cloud-of-Clouds-Angeboten, die auch immer verteilt und mobilen Endsystemen per se zugänglich sind.

### **22.2.6 Mobile Kommunikation**

Mit dem Durchbruch der Mobilfunknetze durch den GSM-Standard auf breiter Basis vom Geschäftskunden bis hin zum privaten Nutzer eröffneten sich Investitionen in die mobile Infrastruktur, dies führt zu kontinuierlicher Innovation durch entsprechende Weiterentwicklungen der mobilen Endgeräte. Mobile Sprachkommunikation ist durch die damit verbundene Erreichbarkeit der Mitarbeiter von Beginn an ein Wert für die Unternehmensprozesse und heute gar unverzichtbar.

Konsumentengetrieben und teilweise durch Subventionierung der mobilen Endgeräte durch die Mobilfunk-Betreiber wurden zunehmend andere Funktionen auf den Endgeräten integriert: Fotokameras, MP3-Player. Um multimediale Dokumente übermitteln zu können, wurde der SMS-Service zum Multimedia Messaging Service (MMS) fortentwickelt. Die zeitlich schritthaltende, zunehmende Nutzung von Web 2.0-Anwendungen, bei denen die Anwender multimediale Inhalte ins Netz stellen und die dafür gleichzeitige Hochrüstung der Mobilfunk-Netze (GPRS, Edge, UMTS, LTE) hinsichtlich der verfügbaren Bandbreiten zu erschwinglichen Kosten ermöglichen es, diese Anwendungen zunehmend mobil zu nutzen. Dies führt zur

Omnipräsenz von Informationen, sowohl für die Nutzung im privaten als auch geschäftlichen Bereich. In der Business-Anwendung ermöglichte dies beispielsweise die Anbindung an die Firmennetze zur Übermittlung von PIM-Diensten wie E-Mail-Push-Dienste.

Aus der mobilen Navigation, die die GPS-Technologie nutzt, werden dabei zunehmend Geo-Positionierungsdaten zur gezielten Bereitstellung von Informationen, sogenannte Location-based Services (LBS) oder ergänzte Realität (Augmented Reality) ermöglicht. Geo-Positionierung, virtuelle Präsenz, ortsunabhängige Erreichbarkeit ermöglichen aber auch neuartige oder erweiterte Business-Software-Anwendungen. Diese stellen jedoch zum Teil höhere Anforderungen an die Darstellung von Informationen und multimediale Eingabemöglichkeiten. Hier profitieren diese Anwendungsmöglichkeiten von dem Wettbewerbsdruck bei den Herstellern mobiler Endgeräte. Fotos, Kurzfilme, Geo-Positionierungsinformationen, Bewegungssensoren (horizontale und vertikale Beschleunigungssensoren), Touch-Screens z. T. mit Gestenerkennung (Multi-touch) oder haptischem Feedback eröffnen bisher nicht gekannte Anwendungsmöglichkeiten. Neue Smartphones werden sogenannte Near-field Communication (NFC)-Module unterstützen, mit denen Gerätidentifikation und Bezahlsysteme realisiert werden können.

Business-Software-Hersteller befähigen ihre zumeist SOA-basierten Software-Plattformen zur Einbindung mobiler Endgeräte. Zum Teil wird dies durch Middleware-Technologien dieser Plattformen erreicht, aber auch durch inhärente Fähigkeiten der RIA-Technologien wie Adobe Flash™, Microsofts Silverlight™ und andere. Sprach-Eingabesysteme erleichtern den Umgang mit diesen Geräten bei der Hands-free Operation unterwegs und im Auto. Es ist zu erwarten, dass die mobilen Anwendungen auch zunehmend als Telematikdienste im Automotive-Bereich Einzug halten.

### ***22.2.7 Neue Anwendungsmöglichkeiten für Smartphones und Tablets***

Smartphones und Tablet-PCs vereinen Funktionen zur Sprachkommunikation, zu multimedialer Dokumentenerstellung und -bearbeitung und mobilen Applikationen. Während früher diese Anwendungen exklusiv auf dem Endgerät betrieben werden konnten, erlauben aktuelle mobile Endgeräte sowohl die gleichzeitige Nutzung von Sprach- und Datenkanal als auch den Ablauf gleichzeitiger Anwendungen. Interaktionen mit den Benutzern können ebenfalls multimedial erfolgen. Ein wesentliches Merkmal am Markt erfolgreicher, heutiger Smartphones ist neben dem Design deren intuitive Bedienbarkeit und Spaß an der Bedienung. Dies wird eher weniger durch die Darstellung der graphischen Oberflächen in Browser-Fenstern als durch die native Verwendung der APIs der Gerätesoftware erreicht. Umgekehrt versuchen die Hersteller von Business-Applikationen, deren Benutzeroberflächen heute meist als Portalanwendungen realisiert sind, also in Internet Browzern dargestellt werden, diese auch auf mobilen Endgeräten verfügbar zu machen. Dazu werden RIA-

Technologien verwendet. Hierbei ergeben sich zurzeit noch Realisierungsbarrieren, da RIA wie Smartphone-Betriebssysteme und Browser herstellerspezifisch und nicht freizügig miteinander kombinierbar sind. Hier wird es noch einige Zeit in Anspruch nehmen, die Basistechnologien dafür zu standardisieren. Wer heute auf eine bestimmte Plattform und damit auf bestimmte RIA setzt, könnte morgen das Nachsehen haben, wenn sich diese Plattform oder RIA am Markt nicht durchsetzt. Die Realisierung von mobilen Business-Applikationen über mehrere RIA-Technologien ist jedoch nicht immer wirtschaftlich. Dies könnte die momentane Realisierung von mobilen Business Applikationen behindern oder verzögern. Neue alternative Browsertechnologien wie HTML5 und WebRTC werden derzeit standardisiert und werden einen erneuten Innovationsschub auslösen.

## 22.3 Integration von Kommunikation in Geschäftsprozesse

Bei der Analyse von Geschäftsszenarien für die Abbildung in automatisierte Geschäftsprozesse und genauerer Betrachtung stellt man fest, dass im Wesentlichen jeder Geschäftsprozess einen korrespondierenden Kommunikationsprozess hat. Für einige kommunikationszentrischen Geschäftsprozesse stellt man sogar fest, dass diese teilweise gar nicht IT-gestützt modelliert sind, weil geeignete Methoden der Modellierung fehlen. Damit kann auch mobile Kommunikation nur teilweise im Rahmen von automatisierten Geschäftsprozessen genutzt werden. Einige einfache Szenarien werden jedoch bereits mit Communication Enabled Business Processes (CEBP) realisiert.

Technisch gesehen werden CEBP-basierte Kommunikationsdienste entweder spezifisch über CTI integriert oder als Komponenten in den Geschäftsprozess-Workflow integriert. Während CTI oder Standard Web-Service Integration auf Applikationsportalebene erfolgt, sind Kommunikationsservices, die als sogenannte Communication Composites ausgestaltet sind, universeller, flexibler und wieder verwendbar über sogenannte Business Process Modeling (BPM)-Plattformen auf grafischer Ebene zu CEBP Composite Applications orchestrierbar. Diese BPM Plattformen erlauben auch die Übermittlung von Businesskontext zu Kommunikationskontext und umgekehrt. Dadurch werden Medienbrüche vermieden und es eröffnen sich effizienz- und effektivitätssteigernde Mehrwerte in der geschäftlichen Anwendung.

Für ein reales, im Rahmen eines Innovationsprojektes zu realisierendes Geschäftszenario wurde erkannt, dass es wohl unterschiedliche Integrationsformen geben muss. Bei der weiteren Analyse von BPM-basierten CEBP und Recherchen zum Stand der Technik [3] ergaben sich grundsätzlich zwei heute bekannte Kategorien von Integration abhängig vom gewählten Szenario:

- Strukturierte Aktivierung von menschlicher Kommunikation und Kollaboration (Workflow),
- Halb-strukturierte Aktivierung von menschlicher Kommunikation und Kollaboration.

### 22.3.1 Strukturierte Kollaboration

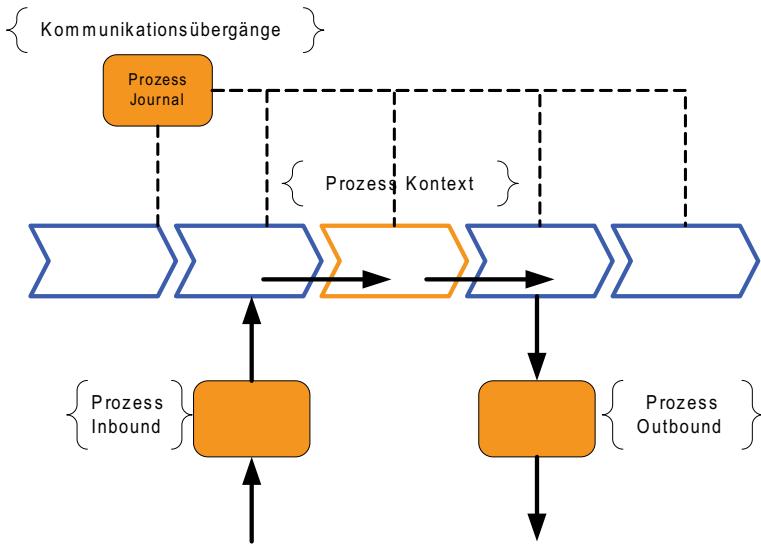
In dieser Kategorie werden Geschäftsprozess-Komponenten mithilfe von BPM zu einem automatisierbaren Gesamtprozess verknüpft, um ein bestimmtes geschäftliches Ziel IT-gestützt zu erreichen. Derartige Geschäftsprozesse werden mit Kommunikation angereichert, indem einfache oder komplexe Kommunikationsschritte in die Prozesskette integriert werden. Dabei wird beim Prozessdesign untersucht, wo fehlende Kommunikation zu potenzieller Verzögerung oder Abbruch des Prozessdurchlaufs führen können und somit Aktivierungen von Kommunikationsvorgängen vorteilhaft sind, um dies zu vermeiden. Dabei werden drei Integrationsvarianten unterschieden:

- Der Geschäftsprozess initiiert während der Ausführung den Kommunikationsvorgang (Process-outbound).
- Der Geschäftsprozess wird durch einen Kommunikationsvorgang geweckt (Process-inbound).
- Der Geschäftsprozess wartet auf den Abschluss eines Kommunikationsvorgangs (Process-context).

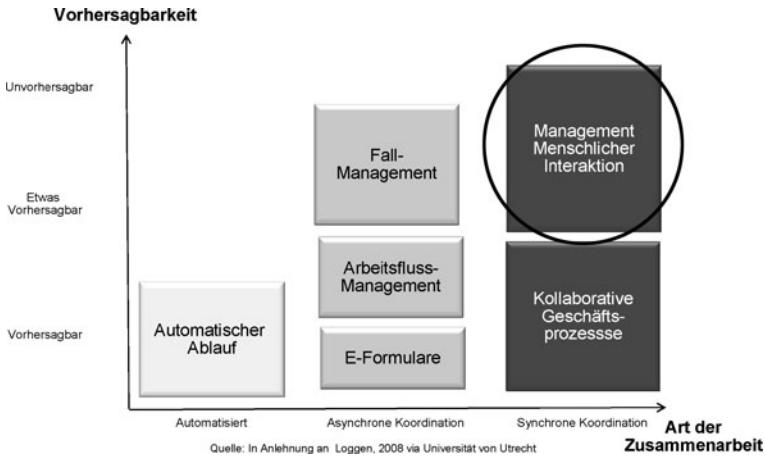
Abgehende Kommunikationsvorgänge (Process-outbound) können durch Informationen im Geschäftsprozess ausgelöst werden, um beispielsweise eine automatisierte Telefonkonferenz in Abhängigkeit vom Präsenzstatus oder der Tell-me-when-Indikation zu starten. Ankommende Kommunikationsvorgänge können die Fortsetzung oder den Start der Abarbeitung des Geschäftprozesses aktivieren oder kontrollieren, indem z. B. Informationen über ein Sprachportal in den Geschäftsprozesskontext bereitgestellt werden.

Einen für den Nutzer merklichen Vorteil durch derartige Integrationsformen in die Geschäftsprozesse bietet die Bereitstellung des relevanten Geschäftskontextes für die Kommunikation und Kollaboration. Dieser Kontext kann zur Designzeit der Geschäftsprozesse über die entsprechenden, teilweise graphischen BPM-Werkzeuge identifiziert und bereitgestellt werden. Dies ist eine sehr mächtige Form der Integration, da geschäftliche oder organisatorische Informationen verfügbar gemacht werden können, die in der Business-Software oder in der Unternehmens-IT vorhanden sind. Dadurch werden Medienbrüche vermieden, die sonst durch den Wechsel zu anderen IT-Verfahren entstehen würden. Beispielsweise können die Rufnummern der Teilnehmer einer Telefonkonferenz aus dem konkreten Geschäftsprozess herangezogen werden.

Strukturierte Kollaboration läuft unter der Kontrolle der System-Software ab, z. B. der Business Process Execution Language (BPEL)-Maschine. Dies führt zu Produktivitätsgewinnen für Routineaufgaben und deckt dank der Integration von Kommunikation auch einige Ausnahmefallbehandlungen mit ab. Diese Ausnahmefälle müssen allerdings bereits zur Designzeit antizipiert worden sein. Für nicht wiederkehrende Aufgaben, für die Bewertung von geschäftskritischen Ereignissen und die damit notwendige Bewältigung von Informationen in verteilten und geographisch verteilten Ressourcen und Projekten, also typische Managementaufgaben und Aufgaben von Wissensarbeitern, sind diese Rahmenbedingungen jedoch



**Abb. 22.2** Integrationsvarianten für CEBP in strukturierte Kollaboration



**Abb. 22.3** Kollaborationsformen in Geschäftsprozessen

zu einschränkend. Außerdem ist bekannt, dass mit Routineprozessen, die etwa 20 Prozent der aufgewendeten Arbeitszeit in Unternehmen ausmachen, circa 80 Prozent der Erlöse erwirtschaftet werden, während 80 Prozent der Aufgaben auf Nicht-Routineaufgaben entfallen, die aber nur zu 20 Prozent zum messbaren, monetären Geschäftserfolg beitragen. Hier liegt also ein erhebliches Effektivitäts- und Effizienzpotenzial für CEBP mit mobilen Erweiterungen brach und sollte unbedingt adressiert werden.

Unsere Analysen zeigen, dass diesbezügliche Geschäftsszenarien und -prozesse heute nicht automatisiert sind, da das Konzept der halb-strukturierten Kollaboration noch unzureichend in heutigen BPM-Systemen abgebildet werden kann. Die Kombination von BPM und CEBP unter benutzerzentrischer, also menschlicher Kontrolle besonders in mobilen Szenarien kann ein Wegbereiter für halb-strukturierte Kollaboration mit erheblichem Potenzial zur Unterstützung von Management und Wissensarbeitern mit klarem Return on Investment (ROI) und faszinierender, neuer komponierter Unternehmensanwendungen werden. Mobile Erweiterungen dazu werden zum einen unverzichtbar sein und zum anderen einen Multiplikator für den resultierenden Geschäftswert bieten.

### **22.3.2 Halb-strukturierte Kollaboration**

Um die Eigenschaften und Auswirkungen benutzerkontrollierter Geschäftsprozesse besser zu verstehen, haben wir einen Ausschnitt eines typischen Dienstleistungsprozesses, so wie er in unterschiedlichsten Unternehmensbranchen Anwendung findet, prototypisch implementiert. Dazu wurde in einer Hochschulkooperation der entsprechende Serviceprozess in unserer belgischen Landesgesellschaft benutzerzentrisch analysiert. Dabei wurden die Rollen der Prozessbeteiligten und deren Interaktion während Interviews der Prozessdesigner mit den Prozessbeteiligten dokumentiert. Für den betrachteten Ausschnitt sind dies der Ressourcenplaner und die ihm zugeordneten Techniker. Die Aufgabe des Ressourcenplaners ist die effiziente Einsatzplanung der Techniker und Einhaltung der mit den Kunden vereinbarten Service Level Agreements (SLA). Dabei wurde deutlich, dass obwohl ein Problemfall (Ticketing)-Workflow-Tool eingesetzt wird, ein erheblicher Anteil der Arbeitszeit des Ressourcenplaners für die Koordination der Techniker über das Kommunikationssystem aufgewendet wird, für die das Ticketing-System wenig Unterstützung bietet.

Das Analyse-Ergebnis wurde als Geschichte in einem Film festgehalten. Der Film dokumentiert alle relevanten Benutzungsfälle (Use Cases) dieses Szenarios. Ein Foliensatz illustriert die prototypische graphische Oberfläche der zukünftigen Anwendung. Dabei wurden noch nicht die Fähigkeiten der benutzten GUI-Technologie berücksichtigt. Film und Oberflächenentwurf wurden nochmals mit den Prozessbeteiligten vor der Realisierung abgestimmt. Im konkreten Fall wurde ein SAP-System auf der SAP-Netweaver™-Plattform mit der SAP Technologie WebDynpro verwendet.

Die resultierende Implementierung realisiert aber keinen Geschäftsprozess-Workflow im herkömmlichen Sinn, sondern ein Ressourcenplaner-Cockpit (Resource Planner's Dashboard) mit der Technikerübersicht im oberen linken Quadranten des Bildschirms. Wird ein Techniker ausgewählt, erscheinen im oberen rechten Quadranten die Details zu dem Techniker mit der Verfügbarkeitsinformation Präsenzstatus und Medienverfügbarkeit. Im unteren linken Quadranten wird die Ticketübersicht angezeigt und im rechten unteren Quadranten die Details zum aktuell ausge-

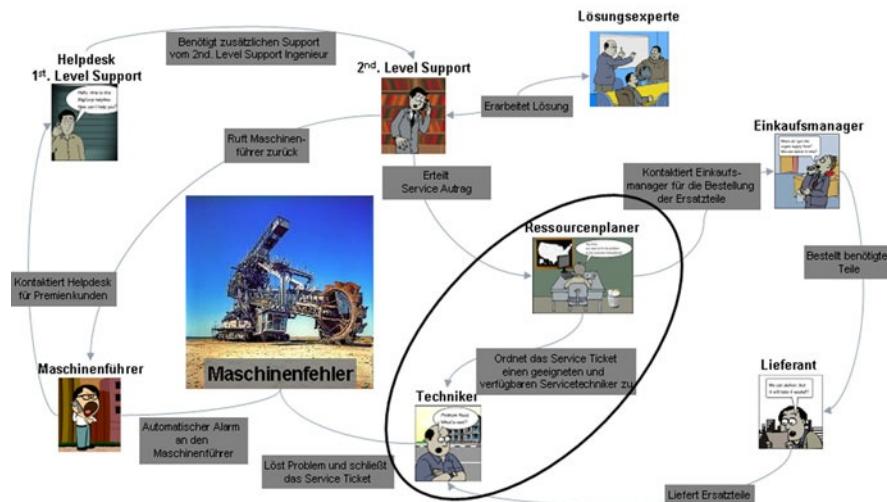


Abb. 22.4 Serviceprozess-Modellierung

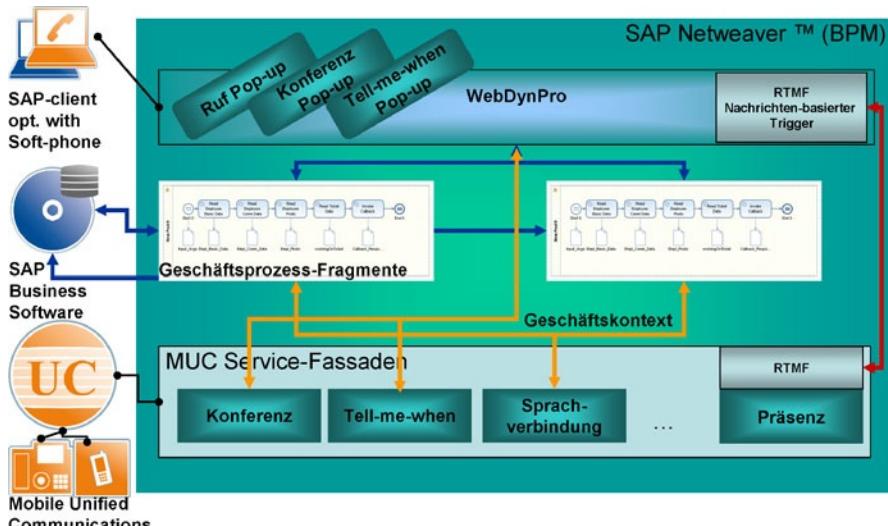


Abb. 22.5 Halb-strukturierte Kollaboration

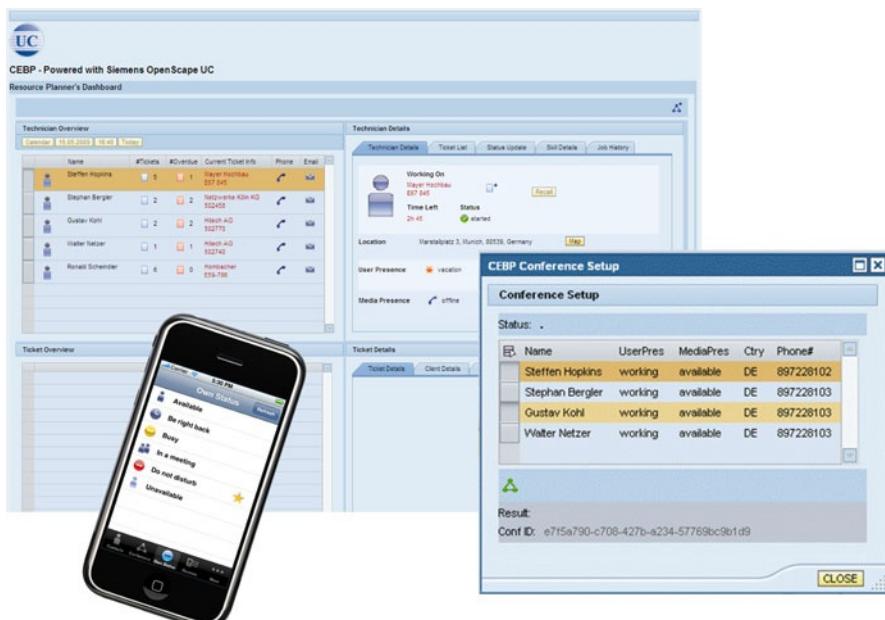
wählten Ticket. Für alle Quadranten wurden mithilfe des BPM-Werkzeuges entsprechende Geschäftsprozessfragmente zusammengestellt, die jeweils beim Aufbereiten der Ansicht oder nach entsprechenden Benutzereingaben in Verbindung mit den betroffenen SAP-Hintergrundsystemen durchlaufen werden und so den Geschäftskontext aktualisieren.

Die folgenden Use Cases wurden implementiert:

- Der Ressourcenplaner konsultiert den einzelnen Techniker um sich über den Bearbeitungsstand des Tickets zu informieren.

- Der Ressourcenplaner initiiert eine Telefonkonferenz mit geeigneten Technikern um den am besten geeigneten Techniker zuordnen zu können.
- Der Ressourcenplaner informiert sich über den zuletzt angefahrenen Einsatzort des Technikers und mögliche Folgeeinsatzorte mit der Möglichkeit der Konsultation des Technikers und des Kunden aus dem Geschäftskontext.

Im Gegensatz zu vorherigen CEBP-Implementierungen bleibt der Ressourcenplaner nach Auslösen von Kommunikationen in seiner Anwendung und kann den Kommunikationsakt aus der Business-Software heraus weiterhin steuern. Die Benutzeroberfläche stellt die im Anwendungsfall sinnvollen Bedienmöglichkeiten bereit. Über Präsenzinformation, die vom mobilen Endgerät des Benutzers über das UCC Mobilitätserweiterungssystem und das SAP Netweaver Real-time Messaging Framework (RTMF) durchgeleitet werden kann, wird dem Ressourcenplaner aktuell die Verfügbarkeit oder Unterbrechbarkeit des Technikers in seinem Cockpit angezeigt. Dadurch gewinnt dieser Geschäftsprozess an bisher ungekannter Agilität. Durch die eingebettete UCC in die Business-Software kann die gesamte Mächtigkeit der Funktionalität mit ihren Mobilitätserweiterungen einfach mitgenutzt werden. Deshalb ist mit Kommunikation auch immer mobile Kommunikation gemeint, ohne dies explizit mit erwähnen zu müssen. Über den One Number Service (ONS), der die Verbindung zum aktuell präferierten Endgerät herstellt, kann der Techniker nach eigener Wahl auf dem Mobiltelefon oder bei mangelndem Empfang in einem Technikraum im Keller beim Kunden über ein dort verfügbares Festnetztelefon erreicht werden.



**Abb. 22.6** Beispiel einer graphischen Oberfläche eines CEBP-Service-Prozesses

Das Szenario kann außerdem mit weiteren Mash-up-Internet-Applikationen angereichert werden. Im Prototyp wurde Google Maps zur Darstellung des letzten und des nachfolgenden Einsatzortes des Technikers integriert. Auch hier konnten Kommunikationsakte direkt aus dem Fenster kontrolliert werden.

Als besondere Eigenschaft der halb-strukturierten Kollaboration bleibt der Ablauf des Geschäftsprozesses vollständig unter der Kontrolle des Ressourcenplaners, der die dahinterliegenden Geschäftsprozessfragmente oder Kommunikationsprozesse während seiner Bedienung des Cockpits auslöst. Die Integration von UCC-Funktionalität mit ihren Mobilitätserweiterungen gepaart mit dem Business-Kontext erzielt erkennbare Produktivitätssteigerungen für den Ressourcenplaner selbst und der Einsatzplanung der Techniker. Das Szenario kann leicht für Sicherheits- oder Logistikbranchen angepasst werden.

### ***22.3.3 Erweiterte Kollaboration durch Mobilitätsfunktionen***

Als mögliche Erweiterungen des Prototypen würde sich die Integration von Google Latitude anbieten, um aktuelle Positionsdaten der Techniker zu übermitteln. In Verbindung mit der UCC-Funktion des Tell-me-when könnte dann ein Pop-up-Fenster im Cockpit des Ressourcenplaners aufscheinen, wenn der Techniker z. B. zwei Kilometer vom Einsatzort entfernt ist. Über die im Pop-up angebotenen Kommunikationsoptionen könnte der Ressourcenplaner dem Techniker letzte Hinweise zum Ticket geben oder dem Kunden die Ankunft des Technikers avisieren. Somit kann die Servicequalität gesteigert werden.

Durch die Möglichkeit der Verknüpfung von Business-Software und Kommunikations-Software können weitere Produktivitätshemmisse beseitigt werden. Stellt der Servicetechniker zwar sein präferiertes Gerät oder seinen Präsenzstatus beim Verlassen des Kunden zurück, hat aber vergessen das zugehörige Ticket zu schließen, kann das System proaktiv zum Schließen des Tickets nachfragen.

Diese einfachen Ergänzungsmöglichkeiten zeigen, wie bereits durch mobile Erweiterungen von UCC integriert in Business-Software agile halb-strukturierte Geschäftsprozesse realisiert werden können. In dem Prototypen wurde unser Standard Mobile UCC-Client verwendet. Für ein Produkt sollte jedoch ein an den Use Case angepasster mobiler Client verwendet werden, der die technischen Elemente einer anwendungsgerechten Semantik verwendet und geeignete Objekte des Ticketing-Systems mit einbindet, wie z. B. den aktuellen Bearbeitungsstand des Tickets oder die Übermittlung des Lösungsberichtes für eine zügigere Abrechnung der Serviceleistung.

### **22.3.4 Bewertung und Ausblick**

Wir haben den Prototypen einigen Prozessdesignern und Chief Information Officers (CIO) vorgeführt. Diese waren fasziniert von den neuartigen Möglichkeiten des halb-strukturierten Kollaborationsparadigmas und seiner mobilen Erweiterungen, sodass diese ad hoc weitere Anwendungsszenarien ableiten konnten und Investitionsbereitschaft in derartige Systeme und Infrastrukturen signalisierten.

In unserem Fallbeispiel wurde deutlich, dass bereits durch Integration von mobiler Kommunikation in Verbindung mit einer mobilen UCC-Client-Software und Location-based Services in CEBP-Geschäftsprozesse, die zudem dem kommunikationszentrischen Design folgen, erhebliche Effizienz- und Effektivitätssteigerungen für das Unternehmen realisiert werden können. Interessanterweise zeigten sich die sichtbaren Effekte in dem Fallbeispiel überwiegend im Back-Office. Die Effekte sind aber klar durch die mobile Erweiterung des Geschäftsprozesses begründet. Für das geschilderte Szenario wurden Effizienzsteigerungen von ca. 45 Minuten pro Techniker und Tag durch die mitwirkenden Mitarbeiter geschätzt. Mögliche Wegezeitoptimierungen kommen noch hinzu.

Im Rahmen einer aktuellen Studie zu mobilen Business-Applikationen haben wir die Rückmeldung erhalten, dass die in Unternehmen dafür Verantwortlichen die Notwendigkeit von Investitionen für derartige Lösungen kennen. Favorisiert werden Kommunikations- und Kollaborationslösungen sowie Customer Relationship Managementsysteme (CRM). Einige der Studienteilnehmer nannten darüber hinaus Lösungen aus den Bereichen Vertriebsunterstützung, Lieferüberwachung, Key Performance Indicator (KPI)-Informationsdienste, und andere Enterprise Ressource Planning (ERP)-Dienste. Jedoch nur wenige haben bereits eine konkret ausgearbeitete Roadmap, an deren konkreter Umsetzung sie arbeiten [4]. Hier gilt es weiter über aktuelle und zukünftige Möglichkeiten zu informieren, wozu dieser Artikel hoffentlich einen initialen Beitrag geleistet hat.

## **22.4 Zusammenfassung**

In diesem Beitrag wurden die konvergierenden Entwicklungen und Trends der notwendigen Lösungskomponenten für mobile Business Applikationen dargestellt. Wesentliche Treiber für Mobile Business Applikationen sind die erreichten Endgerätefähigkeiten und -funktionen durch integrierte Zusatzgeräte und die wirtschaftliche Verfügbarkeit der Mobilen Internets. Unterstützende Trends sind die Entwicklung der Unternehmens-IT zur Bereitstellung der Business-Applikationen im Datacenter oder in der Cloud sowie Web-Service Integration von Kommunikations- und Kollaborationslösungen (UCC) mit Business-Software, Business Intelligence Software, Social Software, Data-Warehouse-Systemen und Groupware-Systemen auf der Basis von SOA-basierten Implementierungen. Durch die Bereitstellung von Composite Business Applications auf der Basis von SCA, die Modellierung von

Geschäftsprozessen auf BPM-Plattformen und die Integration von Kommunikation in Geschäftsprozesse (CEBP) wird die Integration von mobilen Kommunikationsformen erleichtert und erschließt neuartige mobile Geschäftsprozesse.

Es gibt aber auch Hürden, die durch Wissenschaft und Industrie zu adressieren sind, wie proprietäre Betriebssystem-Plattformen von mobilen Endgeräten und proprietäre RIA-Technologien. Einige relevante Technologien wie HTML5 und WebRTC werden bald in der Breite verfügbar sein und zur weiteren Konsolidierung beitragen.

Es wird sicherlich auch einige kleine Helper-Applikationen im geschäftlichen Umfeld geben, die heute primär im privaten Umfeld genutzt werden. Der Geschäftswert von mobilen Business-Applikationen liegt aber für Unternehmen darin, ihren mobilen Mitarbeitern den gezielten Zugang zu relevanten Informationen zu ermöglichen und Kommunikations- und Kollaborationssysteme bereitzustellen, sodass ihre zunehmende Mobilität und das Verwischen ihres privaten und geschäftlichen Lebens nicht zum Nachteil für sie selbst und ihre internen und externen Kunden wird.

Wenn wir also von mobilen Business-Applikationen sprechen, und nur mobile Erweiterungen zu bestehenden Business-Workflows verstehen, greifen wir bei dem Thema zu kurz. Mobilität sollte primär Wissensarbeiter und Manager unterstützen, da genau diese Mitarbeitergruppen hohe Anforderungen an ihre Mobilität haben und weniger mit Routineaufgaben betraut sind.

## 22.5 Acknowledgments

Der Beitrag verwendet teilweise graphische Materialien von Herrn Dr. Philipp Pott und seinem Team, die im Rahmen eines Projektauftrages von der Siemens Enterprise Communications GmbH & Co. KG an die Siemens AG, Central Technologies erstellt worden sind.

## Literaturverzeichnis

1. Harrison-Broninski K (2006) A Software Framework for Human Interactions – Building Human-To-Human Systems, JAVAPOLIS Antwerpen
2. Ravesteyn P (2009) Human Interaction Management, Universität Utrecht
3. Lemahieu W (2010) Business Process Engineering and Communication Enabled Business Processes, JUST Tagung Brüssel
4. Weiß F, Bucherer E. Mobile Enterprise Study (Detecon Consulting Studie im Auftrag der Siemens Enterprise Communications GmbH & Co. KG), Veröffentlichung im Rahmen eines Whitenpapers geplant in 2011

---

# Kapitel 23

## Der Cloud-Broker: dynamische Orchestrierung von Cloud-Diensten zu Smart Mobile Apps

Vitalian A. Danciu, Nils gentschen Felde, Dieter Kranzlmüller, Michael Schiffers und Johannes R. Watzl

**Zusammenfassung** Anwendungen auf mobilen Geräten beschränken sich längst nicht mehr auf Adressbücher und lokale Kalender, sondern haben die Schwelle hin zu hohen Leistungsanforderungen an Rechen- und Übertragungskapazität überschritten. Viele Anwendungen erfordern den Zugriff auf große Datenbestände (etwa Kartenmaterial, Verzeichnisse) und benötigen Rechenkapazität jenseits der des mobilen Gerätes. Moderne mobile Applikationen, sogenannte *Smart Mobile Apps*, müssen auf skalierbare und dynamisch zuteilbare, in Rechenzentren bereitgestellte Ressourcen zurückgreifen, um entsprechende fortgeschrittene Dienstfunktionen erfüllen zu können. *Cloud Computing* bietet sich als Modell für die Realisierung eines solchen Betriebs an.

---

Vitalian A. Danciu  
Münchener Netz-Management Team (MNM-Team), Ludwig-Maximilians-Universität München,  
E-mail: danciu@mnm-team.org

Nils gentschen Felde  
Münchener Netz-Management Team (MNM-Team), Ludwig-Maximilians-Universität München,  
E-mail: felde@mnm-team.org

Dieter Kranzlmüller  
Münchener Netz-Management Team (MNM-Team), Ludwig-Maximilians-Universität München,  
E-mail: kranzlm@mnm-team.org

Michael Schiffers  
Münchener Netz-Management Team (MNM-Team), Ludwig-Maximilians-Universität München,  
E-mail: schiffer@mnm-team.org

Johannes R. Watzl  
Münchener Netz-Management Team (MNM-Team), Ludwig-Maximilians-Universität München,  
E-mail: watzl@mnm-team.org

## 23.1 Cloud Computing als Plattform für Smart Mobile Apps

Das Wachstumspotenzial einer neuen mobilen Applikation mit Bezug auf ihre Nutzerzahlen und die tatsächliche Nutzungsintensität ist sehr schwer prognostizierbar. Die entfernten Ressourcen, auf die eine Smart Mobile App (SMA) angewiesen ist, müssen bei einer etablierten, oft genutzten Applikation einer sehr hohen Anzahl gleichzeitiger Zugriffe standhalten. Gleichzeitig ist es riskant, eine solch hohe Kapazität dedizierter Ressourcen für eine neue Applikation vorzuhalten, deren zukünftige Nutzung nicht absehbar ist.

Durch Verteilung von SMAs über zentrale Verzeichnisse und Verkaufspunkte (sogenannte App Stores) kann manche neue SMA sehr schnell an Beliebtheit (und somit an Ressourcenhunger) gewinnen, während andere dauerhaft eine Nische einnehmen. Die schnelle Skalierbarkeit der rechenzentrumsseitigen Ressourcen wird somit bestimmt für die erfolgreiche Etablierung einer SMA. Ihre Bedeutung wird zusätzlich erhöht durch die Notwendigkeit der Bewältigung von Lastspitzen, die lokal auftreten können oder global, etwa ausgelöst durch plötzliche Erhöhung der allgemeinen Nachfrage (z. B. durch Bedarf an Information über ein internationales Ereignis). Gleichzeitig müssen SMAs möglichst kostengünstig durch rechenzentrumsseitige Ressourcen unterstützt werden, um sich gegenüber konkurrierenden Applikationen durchsetzen zu können.

Diesen Anforderungen an die Skalierbarkeit kann durch *Cloud Computing* begegnet werden, das eine für die geforderte Leistungselastizität geeignete wirtschaftliche und technische Plattform darstellt. Eine Cloud versteht sich als ein Reservoir an Ressourcen, die als Dienst schnell und feingranular zugeteilt werden können. Aus wirtschaftlicher Sicht ist Cloud Computing ein neues Geschäftsmodell zur feingranularen, kommerziellen Bereitstellung von Ressourcen eines Rechenzentrums (RZ) an Endkunden, in der Form von Diensten. Aus technischer Sicht erfolgt die Cloud-Dienst-Bereitstellung typischerweise unter Nutzung einer Middleware, die Ressourcen einer virtualisierten Infrastruktur bereitstellt.

Die öffentlich angebotenen Dienste beschränken sich derzeit meist auf Basisressourcen (Rechenzeit auf virtuellen Maschinen, netzbasierter virtueller Hintergrundspeicher) und werden als homogene Massendienste spezifiziert. Diese Ausprägung lässt sich durch die eingeschränkten Funktionen derzeitiger Middleware technisch begründen sowie durch die erst entstehende Nachfrage nach spezialisierten Diensten seitens der Dienstnehmer. Die derzeitigen Trends weisen aber auf eine Spezialisierung der Cloud-Dienste in naher Zukunft, bis hin zum Angebot anwendungsspezifischer, maßgeschneiderter Ressourcen. Man unterscheidet dabei zwischen verschiedenen Veredelungsstufen der Cloud-Ressourcen, wie in Abb. 23.1 angedeutet. *Infrastructure as a Service* (IaaS) bezeichnet die Bereitstellung der genannten Basisressourcen, die von ihren Nutzern unter Einsatz eigener Systemsoftware und eigener Applikationen betrieben werden. *Platform as a Service* (PaaS) ist eine Dienstklasse, bei der zusätzlich zu Rechen-/Speicherressourcen auch Basissoftware (z. B. Betriebssystem, Datenbanksoftware, Web-Server) oder auch eine Entwicklungsplattform, die Entwurf, Anwendung und Management neuer Anwendungen erleichtert, bereitgestellt werden. *Software as a Service* (SaaS) versteht sich als der Betrieb von

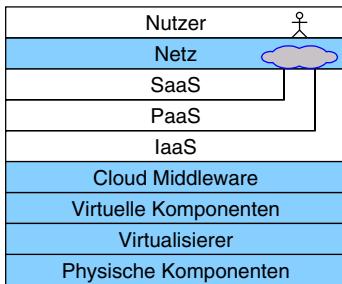


Abb. 23.1 Umfeld von Cloud-Diensten

Applikationen durch den Cloud-Anbieter, die durch die Dienstnutzer (per Internet-Portal) unmittelbar in Anspruch genommen werden können.

### 23.1.1 Beispiel einer Smart Mobile App

Betrachten wir anhand von Abb. 23.2 beispielhaft eine SMA, die die automatische Erkennung eines aktuellen Orts und der zugehörigen Sehenswürdigkeit anhand von Bilddaten ermöglicht. Als Eingabe wird ein von einem mobilen Endgerät aufgenommenes Foto erwartet, die Ausgabe der Applikation ist der Name der fotografierten Sehenswürdigkeit bzw. seines Standortes.

Die SMA verkettet eine Reihe von Funktionen in einen orchestrierten Ablauf:

1. Ein Nutzer fotografiert eine Sehenswürdigkeit mit einem mobilen Endgerät.
2. Das aufgenommene Bild wird als Eingabe rechenzentrumsseitiger Teildienste weitergereicht.
3. Das Eingabebild wird normalisiert, indem ein einheitliches Bildformat (bzgl. Größe, Auflösung, Datenformat etc.) erzeugt wird.
4. Am normalisierten Bild werden die charakteristischen Merkmale in Form eines sogenannten *Feature-Vektors* bestimmt.
5. Der *Feature-Vektor* dient als Suchterm einer Ähnlichkeitssuche in einer Datenbank markanter Orte und Sehenswürdigkeiten.

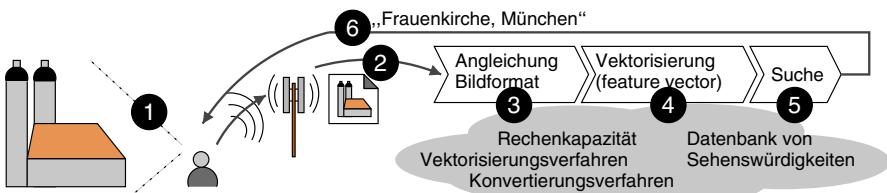


Abb. 23.2 Smart Mobile App zur Ortserkennung

6. Das Suchergebnis wird schließlich an die Smart Mobile App übermittelt, die es dem Nutzer geeignet darstellt.

Die Smart Mobile App greift für die Durchführung ihrer Teilschritte auf verschiedene Ressourcen zurück, die von dem Betreiber der Applikation bereitgehalten werden müssen (hier: Verfahren zur Bildkonversion und -vektorisierung wie auch Verfahren zur Ähnlichkeitssuche und die Verwaltung einer Datenbank von Orten und Sehenswürdigkeiten). Die Durchführung dieser Verfahren setzt entsprechend Rechenzeit und Speicherkapazität voraus. Dies erfordert als Grundlagen für eine Smart Mobile App die Erstellung der Verfahren selbst, die Zusammenstellung der Datenbank sowie das Bereithalten von Rechen- und Speicherkapazität in der Größenordnung der erwarteten Nutzung der Applikation.

### **23.1.2 Dynamische Allokation von SMA-Komponenten**

Die resultierende Struktur der Smart Mobile App ergibt sich somit zu der einer verteilten Anwendung. Derzeitige mobile Applikationen sind in der Regel statisch verteilt, indem ein Teil der Applikation auf dem mobilen Endgerät ausgeführt wird und ein weiterer unter Nutzung im Voraus definierter RZ-Ressourcen umgesetzt ist. Abb. 23.3 stellt die beiden Modelle der derzeitigen mobilen Anwendungen (links) und Cloud-gestützter Smart Mobile App (rechts) gegenüber. Die statische Verteilung einer Applikation ist nur dann zweckmäßig, wenn über ihre dem Endgerät entfernten Teile zuverlässige Aussagen gemacht werden können, erfordert aber die Auswahl und Reservierung von Ressourcen bereits vor Nutzung der Applikation. Dies widerspricht dem sowohl für Cloud als auch für Smart Apps vorstellbaren flexiblen Markt von Diensten. Es ist daher wünschenswert, die für die SMA erforderlichen Ressourcen der Nachfrage entsprechend, d. h. dynamisch, zu allozieren. Eine möglichst kostengünstige dynamische Allokation beinhaltet auch den Vergleich zwischen verschiedenen Dienstanbietern auf der Basis von Preisen und Leistungsparametern der Dienste.

Eine dynamische Allokation stellt ihrerseits die Herausforderung einer dynamischen Auswahl und Bindung von Anwendungsbestandteilen. Abbildung 23.3 zeigt die Struktur einer derart verteilten Anwendung: die Kernapplikation wählt funktionale Komponenten aus einem öffentlich bereitgestellten Angebot anhand applikations- und nutzergespebener Kriterien aus und wendet sie zur Erbringung des Gesamtdienstes an.

Die Unterstützung von Smart Apps erfordert also eine intelligente Verteilung von Applikationskomponenten zwischen dem Endgerät einerseits und RZ-Ressourcen andererseits. Die Auslagerung von Teilen einer Smart App „in die Cloud“ kann, je nach Applikationsklasse, anhand einer Vielzahl von Kriterien bewertet werden. Geht man von einer Zusammenstellung Cloud-seitiger Ressourcen zur Ergänzung einer Smart App aus, ist die Kompatibilität zu anderen Dienstkomponenten von Bedeutung. Auch die Netzlast während der Nutzung, die Latenz der Komponente, Garantien bezüglich Verlässlichkeit und Dienstgüte können beispielsweise zur

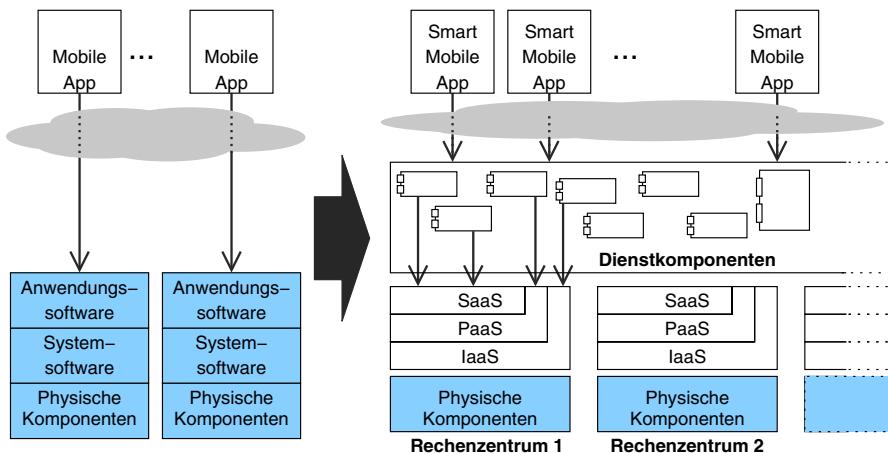


Abb. 23.3 Von der mobilen Anwendung zur dynamisch verteilten Smart Mobile App

Bewertung hinzugezogen werden. Schließlich ist, wie bei jeder Dienstleistung, die Bewertung anhand der finanziellen Kosten für die Nutzung für eine Auswahl von Anbieter und konkret angebotenem Cloud-Dienst relevant.

Bei den von einer SMA in Anspruch genommenen Ressourcen kann es sich allerdings um verschiedenartige Objekte handeln, die aufgrund solcher Kriterien nur schwer vergleichbar sind. Neben reiner Rechenleistung ist die Nutzung von Applikationsteilen bzw. Algorithmen denkbar, Indexierungsfunktionen des entfernten Speichers oder die abrechnungsfähige Nutzung existierender Datenbestände.

Die nicht-triviale Funktion der Auswahl von SMA-Komponenten erfordert daher neben einem Überblick des Angebots Kenntnis über die Anforderungen der Applikation, Erfahrungswerte mit Bezug auf die tatsächliche Leistung einzelner Komponenten sowie Vergleichsverfahren zur Entscheidungsfindung: ihre manuelle Durchführung durch den Nutzer ist somit auszuschließen. Gleichzeitig implizieren die Automatisierung dieser Tätigkeiten und ihre Übernahme durch Software-Komponenten die Forderung nach bislang nicht hinreichend untersuchten wissenschaftlichen und technologischen Bestandteilen.

## 23.2 Herausforderungen bei der Orchestrierung von Cloud-Diensten

Das Beispiel in Abschn. 23.1.1 weist eine Reihe von Herausforderungen auf, denen sich ein Lösungsansatz stellen muss. Soll ein Dienst (z. B. ein Dienst zur bildbasierten Orts- und Sehenswürdigkeiterkennung oder auch einer seiner Teildienste) automatisiert verarbeitbar beschrieben werden, so erfordert dies mindestens die Beschreibung der angebotenen Dienstgüte (z. B. als formalisierte *Service Level Agreements* (SLAs)), der Bepreisung des Dienstes (z. B. pro Anfrage, pro Nutzungsdauer,

pauschal), die Beschreibung von Dienstzugriffspunkten sowie die Interoperabilität des Dienstes mit anderen (Teil-) Diensten. Auswahl, Komposition und Aggregation geeigneter Teildienste zu einem höherwertigen Dienst müssen automatisiert darstellbar sein. Hierbei sind Fragen nach dem Auffinden geeigneter (Teil-)Dienste (*service discovery*), Kriterien und/oder Metriken zur Auswahl sowie geeignete Algorithmen dazu zu entwickeln. Neben den Anforderungen an die Darstellbarkeit eines Dienstes müssen im Hinblick auf seine technische Umsetzung auch Funktionen der Benutzerverwaltung (inklusive einer *Authentication and Authorization Infrastructure* (AAI)) sowie Verantwortlichkeiten und Rollen zur Abrechnung (*accounting & billing*) etabliert werden. Bei letzteren ergeben sich organisatorische Herausforderungen, da Verantwortungsbereiche und Zuständigkeiten meist unter einer Vielzahl beteiligter organisatorischer Einheiten koordiniert werden müssen.

Im Folgenden werden die grundlegenden Fragestellungen von Cloud-basierten SMAs erarbeitet, insbesondere

1. Vergleichbarkeit von Cloud-Diensten,
2. Auswahl von Ressourcen, Diensten und Dienstanbietern,
3. Zusammensetzung von Diensten aus Teildiensten und
4. Aspekte eines Marktplatzes für Cloud-Ressourcen.

### Vergleichbarkeit von Cloud-Diensten

Um Cloud-Dienste vergleichen zu können, müssen Informationen über diese vorhanden sein, die in einer standardisierten Form abrufbar sind. Die Elemente dieser Informationen sind Objekte, denen ein Wert zugeordnet ist. Für jede Klasse dieser Elemente müssen Einheiten und Skalen sowie Vergleichsoperationen definiert sein. Sollen z. B. Aussagen über die Performanz eines Dienstes, die verwendeten Ressourcen, Nutzungsschnittstellen (APIs) und Ein- und Ausgabeformate getroffen werden, müssen seine Attribute seitens des Dienstanbieters bereitgehalten werden. Informationen zu unserer Beispiel-SMA könnten also folgende Attribute beinhalten:

**Tabelle 23.1** Bereitgestellte Informationen über einen Dienst

Attribut	Wert
Dienst	Bilderkennung
Bildformat	JPG
Maximale Wartezeit auf Ergebnis	5 s
:	:

Die Bereithaltung dieser Informationen kann durch die Dienste selbst geschehen (Selbstbeschreibung), durch ein Verzeichnis oder durch Einsatz einer Erkennungstechnik.

Im Lichte der Heterogenität der Dienstrealisierung verschiedener Anbieter stellt sich zur Dienstvergleichbarkeit insbesondere die Herausforderung einer standardisierten und zweckmäßigen (bzw. erweiterbaren) Ressourcen- und Dienstbeschreibung für Cloud-Dienste einerseits und andererseits der Definition von Vergleichsoperationen, die Aussagen der Art „ist besser als“ bzw. „ist schlechter als“ über ein Elementepaar der Cloud-Dienst-Information erlauben.

### Auswahl von Ressourcen, Teildiensten und Dienstanbietern

Seitens der mobilen Applikation wird festgelegt, welche (Teil)Dienstfunktionalität benötigt wird und zu welchen Konditionen der Dienst genutzt werden soll. Auf Basis der Informationen, die von Seiten des Dienstanbieters zur Verfügung gestellt werden, legt der Nutzer einer Smart Mobile App Parameterwerte fest, die Auswahlkriterien für Teildienste darstellen. Die Wahl des Dienstanbieters erfolgt sowohl auf Basis von Informationen über die Cloud-Dienste, als auch anhand der vom Nutzer geforderten Parameter des Dienstes. Seitens des Dienstbetreibers können Informationen zur Nutzung, der Güte, dem Preis und zum (nutzerseitigen) Management des Dienstes bereitgestellt werden. *Nutzungsinformationen* beschreiben den Dienst in einer standardisierten Form, einschließlich seiner Nutzungsschnittstelle (API). *Dienstgüteinformationen* geben den *Service Level* des Dienstes an und sollten ein Maß für die Dienstverfügbarkeit einschließen. Informationen zum *Nutzungspreis* deklarieren den für die Nutzung des Dienstes zu entrichtenden Betrag einschließlich der Einheit der Dienstnutzung. Als *Managementinformationen* werden Angaben zum Betrieb, zur Abrechnung und Sicherheit eines Dienstes eingeschlossen.

Automatisierte Entscheidungen auf Basis solcher Informationen ergeben, formal betrachtet, ein mehrdimensionales Optimierungsproblem, welches im Allgemeinen mehrere Lösungen hat. Zur Auswahl genau einer Lösung muss eine Bewertungsfunktion herangezogen werden, die anhand von (für die Smart Mobile App spezifischen) Gewichtungen eine eindeutige Entscheidung ermöglicht. Es stellen sich folglich bei der Auswahl von Diensten und ihren Anbietern Herausforderungen bezüglich der formalen und zweckmäßigen (d. h. standardisierbaren) Darstellung von Informationen zu Dienstgüte und Preis (als Teil der *service level*) sowie zur Festlegung von Kriterien zur Gruppierung und Einordnung von Cloud-Diensten, um eine Auswahl aus ähnlichen Diensten zu unterstützen. Darüber hinaus werden Algorithmen und Gewichtungsschemata gesucht, die eine Optimierung der Auswahl erlauben.

### Aggregation von Diensten

Transparente Kombinationen von Diensten ermöglichen die Bereitstellung neuer (höherwertiger) Dienste oder Workflows für eine mobile Applikation. So z. B. kann ein Dienst, der Bildern einen rechenaufwendigen Effekt hinzufügt und sie auf ei-

ne bestimmte Auflösung umsetzt, bei Nichtverfügbarkeit mit der Reihung von zwei Diensten (je einer für Effekt und Verkleinerung) ersetzt werden. Für die mobile Applikation macht es keinen Unterschied, da das Ergebnis dem von der Applikation erwarteten entspricht.

Die programmatische Ermittlung gültiger Kombinationen von Ressourcen und Teildiensten, ihrer resultierenden Funktion sowie der Äquivalenz (sowohl syntaktisch, d. h. bezüglich Schnittstellen und Datenformaten, als auch semantisch, d. h. mit Bezug auf inhaltliche Aspekte) verschiedener Kombinationen stellen große Herausforderungen, wie sie bereits z. B. im IT-Management oder der kontextsensitiven Datenverarbeitung formuliert wurden. Die Erfahrungen in diesen Gebieten legen nahe, dass eine *a priori* erfolgte Einschränkung der erlaubten Kombinationen durch Standardisierung eine plausible Möglichkeit zur Erreichung dieses Ziels bietet.

### Aspekte eines Marktplatzes für Cloud-Ressourcen

Betrachtet man typische Management-Teildisziplinen wie die FCAPS (*Fault- & Configuration-Management, Accounting, Performance- & Security-Management*), so findet man in jedem dieser Bereiche nicht-triviale Herausforderungen. Durch ein multiorganisationales Umfeld, das sich aus jeweils eigenständigen und für sich selbst verantwortlichen Parteien zusammensetzt, sind Verantwortlichkeiten kompliziert bis fast gar nicht festzulegen. In Fehlerfällen gilt es einen Ansprechpartner zu lokalisieren bzw. *a priori* festzulegen. Außerdem muss eine Aufzeichnung von Dienstnutzungen und deren Abrechnung mehrere Organisationen überspannend etabliert werden. Aggregierte Dienste, die sich aus vielen Teildiensten, die durch unterschiedliche Partner erbracht werden, darstellen, sind nur sehr schwierig in ihrer Güte zu bewerten (sprich SLAs festzulegen). Im Bereich der IT-Sicherheit gilt es nicht nur die Vertraulichkeit, Integrität und Authentizität von Informationen zu gewährleisten, sondern auch eine übergreifende Benutzerverwaltung mit entsprechenden Authentifizierungs- und Autorisierungsmechanismen zu gestalten. All diese und viele weitere Fragen sind zum Teil Inhalt aktueller Forschung.

## 23.3 Abstraktion durch funktionale Schichtung als Lösungsansatz

Die grundlegende Idee zur Lösung des in Abschn. 23.1.1 beispielhaft beschriebenen Szenarios ist die Abstraktion und Vermittlung vorhandener Cloud-Dienste durch einen sogenannten *Cloud Broker*. Die Komponente des Brokers ermöglicht es, eine Kombination aus beliebigen Cloud-basierten Diensten transparent zur Nutzung an eine mobile Applikation bereitzustellen, er übernimmt also die Aufgaben der

dynamischen Auswahl und Bindung geeigneter Applikationskomponenten wie in Abb. 23.3 dargestellt ist.

Bildet man diese Idee auf einen konkreten Anwendungsfall ab, so interagiert der Endanwender mit einer Kernkomponente, die auf einem mobilen Gerät ausgeführt wird. Diese Komponente stellt unter anderem die graphische Benutzeroberfläche zur Verfügung. Die Smart Mobile App selbst bzw. deren Logik kann einerseits aus lokal (auf dem mobilen Gerät) ausgeführten Komponenten und andererseits aus Komponenten, die „in der Cloud“ liegen, zusammengesetzt sein. Die Komponenten (ob lokal oder in der Cloud) werden an die mobile Applikation gebunden und ergeben die Gesamtfunktionalität der Smart Mobile App.

Die Orchestrierung und Choreographie von Diensten geschieht immer im Rahmen von Workflows, die allerdings im Kontext dieser Arbeit weder statisch sind, noch a priori festgelegt werden können. Vielmehr müssen diese dynamisch auf Veränderungen des Umfeldes und möglicher Nutzeranforderungen anpassbar sein. Die Notwendigkeit, Dienstfunktionalitäten in adäquate Workflow-Komponenten zu kapseln, ergibt sich dabei nicht nur aus der Anforderung, einfache Teildienste (z. B. Objektbestimmung, Ermittlung von Feature-Vektoren, Angleichung von Bildformaten oder Geo-Tagging (Hinzufügen von geographischen Metadaten) von Objekten) zu integrieren, sondern zunehmend aus dem Erfordernis, auch komplexe mathematische Modelle bereitzustellen und diese zu koppeln, Wissen zu aggregieren oder aufwändiges Rendering (Bilderzeugung auf Basis eines Modells) im Rahmen von Visualisierungen zu ermöglichen.

Komponenten-orientiertes Design ist seit Jahren ein aktives Forschungsgebiet (siehe z. B. Allan et al., 2006 [1] für den Bereich wissenschaftlicher Workflows) mit vielfältigen Forschungsfragestellungen, die insbesondere die Ausgewogenheit von Flexibilität und Performanz sowie von lokalen und verteilten Komponentenausführungen, Fragen zur optimalen Architektur, zur Adäquatheit von Frameworks zur Implementierung der Architekturen und zur Tauglichkeit der zu Grunde liegenden Infrastruktur betreffen. Eine systematische Behandlung dieser Fragen erfordert eine genaue Kenntnis der Schnittstellendefinitionen und der Mechanismen zur Dienstkopplung.

Fragestellungen zum Komponenten-orientierten Design umfassen aber nicht nur klassische Workflow-Aspekte und Trade-Off-Diskussionen. Ein zunehmendes Gewicht – auch für den Rahmen dieses Kapitels – erhält der Bereich adaptiver Implementierungen. Die Grundidee ist dabei, unterschiedliche Implementierungen über eine gemeinsame Schnittstelle anzubieten. Adaptives Dienstverhalten wird dann durch die dynamische Auswahl der situativ „am besten geeigneten“ Implementierung erreicht, basierend auf spezifischen Dienstparametern oder auf der Verfügbarkeit (oder allgemeiner: der Verlässlichkeit) von Ressourcen, wie sie z. B. von einem Cloud-Broker ermittelt werden. Ein typisches Beispiel stellt der Aufruf einer Komponente über eine einzige Schnittstelle dar, die Ausführungsumgebung wird jedoch dynamisch ausgewählt, abhängig von der Größe der Eingabedatensätze und/oder der Verfügbarkeit von Rechenressourcen. Ein schönes Anschauungsbeispiel dazu findet sich bei Castranova und Goodall [6].

### 23.3.1 Verallgemeinerte funktionale Betrachtung

Die zur Lösung der angeführten Herausforderungen notwendigen Teifunktionen lassen sich in ein Schichtenmodell bestehend aus neun Schichten, die in vier Ebenen untergliedert sind, einordnen (siehe Abb. 23.4).

Die unterste Ebene stellt die Ressourcenebene dar, die auf der Seite der Dienstanbieter Virtualisierungsfunktionen (Schicht (i)) auf der zugrunde liegenden Hardware und Ressourcen-Management, Funktionen (Schicht (ii)) bereitstellt. Darauf aufbauend liegt eine Ebene, die Einzelkomponenten (z. B. zur Teildiensterbringung) betrachtet. Schicht (iii) ist dabei für die Informationsakquise verantwortlich. Diese kann einerseits durch Selbstbeschreibung der Einzelkomponenten oder durch eine dafür eigens vorgesehene Komponente, die Beschreibungen von Diensten erstellt, durchgeführt werden. Auf Schicht (iv) werden Teildienste bestimmten Applikationskomponenten zugeordnet. Schicht (v) stellt die Funktionalität für die Auswahl dieser Komponenten zur Verfügung.

In der nächsten Ebene werden komplexe Komponenten betrachtet. Eine komplexe Komponente ist ein aus Teildiensten (Applikationskomponenten) zusammengesetzter Dienst. Die beiden Schichten in dieser Ebene sind verantwortlich für die Interkomponentenbindung (Schicht (vi)), die die Teildienste zu der komplexen Komponente zusammensetzt, und Schicht (vii), die für die Auswahl der Komponenten, die die Smart Mobile App benötigt, zuständig ist.

Auf der obersten Ebene – der Ebene der Smart Mobile App – gibt es die Schicht, die die Funktionalität bereitstellt, Komponenten an eine Smart Mobile App zu binden (Schicht (viii)), und zuletzt Schicht (ix), die die Nutzungsschicht darstellt. Ist ein von der Smart Mobile App verlangter Cloud-Dienst durch eine Einzelkomponente darstellbar, geht Schicht (iv) direkt in Schicht (vii) über.



Abb. 23.4 Aufeinander aufbauende Funktionen induzieren geschichtetes Modell

### 23.3.2 Funktionale Aspekte

Eine mobile Applikation möchte eine bestimmte Aufgabe mit Hilfe von Cloud-Diensten lösen. Dazu besitzt sie als Informationen die Daten zur Weiterverarbeitung und die Vorgaben an den Dienst. Im Beispiel des Ortserkennungsdienstes sind die Eingabedaten ein Bild in einem bestimmten Format. Die Vorgaben stellen Informationen über die gewünschte Dienstbeschaffenheit dar. Exemplarisch können Vorgaben der mobilen Applikation wie folgt aussehen:

**Tabelle 23.2** Minimale Anforderung an den gewünschten Dienst

Dienst	Ortserkennung
Eingabeformat	JPG
Bildgröße	1024, 768
Maximale Wartezeit auf Ergebnis	7 s
Maximaler Preis	2 \$

In diesem Beispiel könnten dann Beschreibungen der zusammengesetzten Applikationskomponenten folgende sein:

**Tabelle 23.3** Angebotene Dienste

	Dienstangebot 1	Dienstangebot 2	Dienstangebot 3
Dienst	Ortserkennung	Ortserkennung	Ortserkennung
Zugangspunkt	URL1	URL2	URL3
Eingabeformat	TIFF	JPG, TIFF, BMP	JPG
Bildgröße	1920, 1080	1920, 1080	800, 600
Maximale Wartezeit auf Ergebnis	2 s	5 s	1 s
Preis	3 \$	1.8 \$	0.5 \$

Mittels der Vorgaben werden in Frage kommende Applikationskomponenten gesucht, und der beste oder die besten davon ausgewählt (vgl. Schicht (vii)). Dienstangebot 1 kann aufgrund der fehlenden Unterstützung für das JPG-Format und des höheren Preises nicht verwendet werden. Die unterstützte Bildgröße schließt Dienstangebot 3 aus. Dienstangebot 2 wird ausgewählt, und der mobilen Applikation wird die Zugangspunktinformation mitgeteilt (Schicht (viii)).

Um diese zusammengesetzten Dienstangebote bereitstellen zu können, muss eine Instanz, die Dienste aggregiert, auf die Beschreibungen von Teildiensten Zugriff haben. Diese Beschreibungen müssen vom Dienstanbieter oder vom Dienst selbst bereitgestellt werden (Schicht (iii)). Sie muss außerdem in der Lage sein, die Teildienste – im Falle unseres Beispiels *Sehenswürdigkeit bestimmen*, *Feature-Vektor ermitteln* und *Bildformat angeleichen* – einer Komponente (Ortserkennung) zuzuordnen (Schicht (iv)) und aus einer Liste zu ermitteln (Schicht (v)). Das Auswählen der

Applikationskomponenten und Dienste zur Aggregation geschieht durch die Funktionalitäten der Schichten (v) und (vii). Nach der Durchführung aller zur Diensterbringung notwendigen Berechnungen muss das Ergebnis an die mobile Applikation übermittelt und dort dem Benutzer zur Verfügung gestellt werden (Schicht (ix)).

### 23.3.3 Rollen und Aufgabenzuteilungen

Durch die vorgeschlagene Schichtung bedingt ergeben sich eine Menge unterschiedlicher Rollen und Aufgabenzuordnungen. Auf der Hand liegend sind mindestens die drei Rollen des *Cloud-Dienstanbieters*, des *Cloud-Brokers* sowie des *Endanwenders*. Letzterer wird in der Regel durch eine Smart Mobile App repräsentiert.

Ein Cloud-Broker stellt bei einer solchen Dreiteilung eine sehr zentrale und komplexe Rolle dar, die sich in mehrere Aufgabengebiete untergliedern lässt. Die Aufgabengebiete lassen sich mindestens in die Dienst- und Providerverwaltung, einen Selektor für Dienstkomponenten, die Dienstaggregation, eine Benutzerverwaltung sowie ein Abrechnungssystem untergliedern.

Der Broker muss in der Lage sein Informationen über Dienste und Dienstanbieter in eine Datenbank einzupflegen und diese zu verwalten (Dienst- und Anbieterverwaltung). Mit einer Anfrage eines Nutzers wählt ein Selektor bzw. eine Selektionslogik Dienstkomponenten aus, die von der Smart Mobile App benötigt werden. Diese Dienstkomponenten vermittelt der Broker sodann an den Nutzer. Eine weitere Teilrolle ist der Dienstaggregator. Dieser stellt aus Teildiensten einen neuen, höherwertigen Dienst zusammen, erstellt eine Beschreibung und trägt den neuen Dienst in die Datenbank der bestehenden Dienste ein. Um sicherstellen zu können, dass die Zuordnung von Dienst auf Nutzer eindeutig und die Nachvollziehbarkeit der Dienstnutzung gewährleistet ist, muss eine Benutzerverwaltung existieren. Diese ist auch für das Abrechnungssystem erforderlich. Das Abrechnungssystem protokolliert die Nutzung von Diensten pro Nutzer und verrechnet die angefallenen Kosten.

Die Rolle des Cloud-Dienstanbieters zu guter Letzt ist vergleichsweise einfach. Die Herausforderungen in diesem Kontext beziehen sich vielmehr auf die Dienstbeschreibungen sowie das dynamische Binden von zu verwendenden Dienstkomponenten. Ansonsten agiert die Rolle des Dienstanbieters autonom und steht in keinem direkten Abhängigkeitsverhältnis.

### 23.3.4 Kommunikationspfade

Wie bereits der vorhergehende Abschnitt gezeigt hat, ist die Gestaltung der Broker-Komponente nicht-trivial. Abbildung 23.5 illustriert mögliche Kommunikationsbeziehungen zwischen Teilfunktionen eines Cloud-Brokers. Die Verwaltung von Dienstanbietern und deren Diensten ist neben einer Dienstaggregation eine zentrale Aufgabe. Ein Dienstaggregator erhält von allen teilnehmenden Dienstanbietern Be-

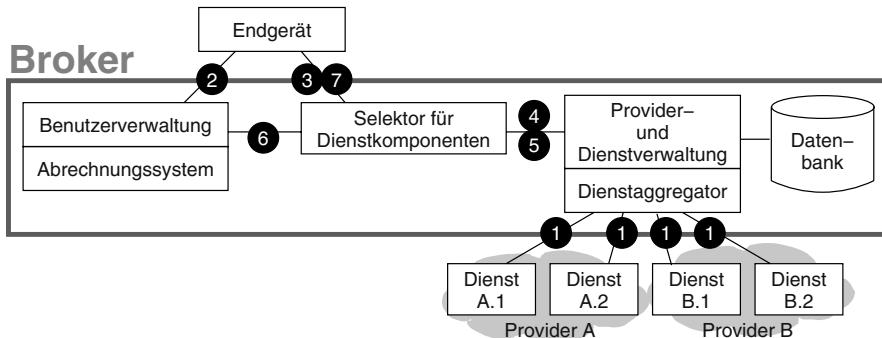


Abb. 23.5 Kommunikationsbeziehungen

schreibungen über sich und ihre angebotenen Dienste. Dies erfolgt periodisch und bei Änderungen (vgl. Abb. 23.5, (1)). Besondere Herausforderungen ergeben sich dabei insbesondere durch die Beschreibung von Dienstanbietern und -angeboten sowie dem Modellieren adäquater Informations- und Datenmodelle und deren Abbildung auf ein korrespondierendes Datenbankschema.

Eine Smart Mobile App authentifiziert sich bzw. den Benutzer der Applikation gegenüber dem Broker unter Nutzung der Benutzerverwaltungskomponente (2). Eine besondere Herausforderung stellt dabei eben diese Benutzerverwaltung dar, da in einem solchen multiorganisationalen und hoch dynamischen Szenario ebenfalls hochwahrscheinlich eine auf mehrere organisatorische Einheiten verteilte Nutzerbasis vorliegt. Eine entsprechende Kopplung zu einer gemeinsamen Infrastruktur inklusive einer *Authentication and Authorization Infrastructure* (AAI) ist an dieser Stelle notwendig.

Nach der erfolgreichen Benutzeranmeldung werden Anfragen seitens der Smart Mobile App an den Selektor für Dienstkomponenten gestellt (3). Eine Herausforderung dabei ist einmal mehr die Notwendigkeit der Beschreibung. In diesem Fall muss eine Beschreibung der Anforderungen an einen ausführenden Teildienst im Sinne einer „Job-Beschreibung“ in einem standardisierten Format festgelegt werden.

Der Selektor wiederum schickt der Provider- und Dienstverwaltung korrespondierende Anfragen (4), die diese im günstigen Falle mit einer oder mehreren Dienstbeschreibungen beantwortet (5). Bei der Selektion geeigneter Teilkomponenten sind insbesondere die Auswahllogik sowie die Kriterien zur Auswahl geeigneter Teildienste von entscheidender Relevanz. Zum einen muss eine Vergleichbarkeit von Diensten gegeben sein (um z. B. zum kostengünstigsten Anbieter zu wechseln), zum anderen müssen Optimierungsmethoden zur Bestimmung der „besten“ Teildienstkomponenten festgelegt werden.

Bei erfolgreicher Suche nach Diensten und Aggregation dieser zu Applikationskomponenten wird die Aufzeichnung der Nutzung dieser Dienste initiiert. Dazu wird vom Selektor die Beschreibung der Dienste an die Abrechnungsverwaltung übermittelt (6) und dem Nutzer bzw. der Smart Mobile App der Zugriffspunkt für

den Dienst mitgeteilt (7). Ähnlich wie bei der Nutzerverwaltung ist auch die tatsächliche Abrechnung eines genutzten Dienstes inkl. seiner verwendeten Teildienste nicht-trivial. Dies resultiert vor allem daraus, dass sich ein solcher Anwendungsfall (ähnlich wie auch die Benutzerverwaltung) in einem hoch dynamischen und interorganisationalen Umfeld bewegt, so dass es vor allem gilt Zuständigkeiten festzulegen und zuzuweisen.

### **23.3.5 Informationsbeschreibung und -modellierung**

Insbesondere müssen die Beschreibungen von Dienstanbietern und ihren Diensten modelliert werden. Der Broker nimmt auf Seiten der SMA Anfragen und auf Seiten der Dienstanbieter Dienstbeschreibungen entgegen. Die Dienstanbieter stellen dem Broker Informationen über die angebotenen Dienste und Ressourcen zur Verfügung. Sinnvoll ist es, für die auszutauschenden Daten Formate zu verwenden, die an bestehende Formate angelehnt sind. Ausgewählte Beispiele dazu, die sich bereits in anderen Anwendungsbereichen bewährt haben, finden sich in Abschn. 23.3.6. Probleme, die bei diesen Beschreibungen auftreten können, sind vielfältiger Natur. Ist die Benennung von Diensten, die die gleiche Funktionalität anbieten, nicht gleich, kann programmatisch nicht mehr trivial erkannt werden, dass diese Dienste die gleiche Funktionalität anbieten.

Einmal mehr zeigt sich auch an dieser Stelle, dass sich viele – wenn auch nicht alle – Probleme, die sich im Rahmen dieser Arbeit bewegen, auf die Beschreibung und Vergleichbarkeit von Dienstangeboten reduzieren lassen. Diese Problematik der semantischen Ungleichheiten ist nicht neu und wird auch in vielen themennahen Forschungs- und Anwendungsfeldern behandelt. Einige prominente Versuche von Lösungsansätzen lassen sich z. B. im Bereich des Grid Computing (vgl. hierzu auch das nachfolgende Abschn. 23.3.6) finden.

### **23.3.6 Relevante Technologien und Standards**

Betrachtet man die vorgestellte Schichtenbildung in Abb. 23.4, so lassen sich anhand dieses Ordnungsrahmens themenverwandte Arbeiten, die als Basis für eine Umsetzung des vorgeschlagenen Lösungsansatzes dienen können, einordnen.

Die IETF [8] hat mit einem Internet-Draft die Spezifikation eines *Cloud Reference Framework* begonnen. Ziel dieser Spezifikation ist das Bilden einer generischen Basis, um standardisierte Schnittstellen und Dienstzugriffspunkte zu den funktionalen Schichten einer Cloud schaffen zu können. Das Cloud Reference Framework besteht aus vier horizontalen Schichten – namentlich dem *Application/Service Layer* (ASL), dem *Resource Control Layer* (RCL), dem *Resource Abstract and Virtualization Layer* (RAVL) und dem *Physical Resource Layer* (PRL) – und einer vertikalen Schicht. Die vertikale Schicht beinhaltet vorwiegend managementrele-

vante Funktionalitäten wie das Konfigurationsmanagement, Registrierungsdienste, Auditierung, Sicherheitsmanagement und Dienstgütemanagement. Diese Standardisierungsbemühungen bilden eine gesunde Basis für die Darstellung der Ressourcenebene mit ihren Schichten (i) „Virtualisierungsfunktionen“ und (ii) „Ressourcen-Management“.

In *A Broker for OWL-S Web Services* [9] zeigen Massimo Paolucci et al. einen Broker für Web Services, die mithilfe einer *Semantic Web Service Description Language* auf Basis der *Ontology Web Language* (OWL) beschrieben werden (vgl. Schicht (iii)). Die Arbeit beschreibt Ansätze, die die Ebene der Einzelkomponenten aus Abbildung 23.4 tangieren. Das vorgestellte Modell kennt dabei drei Rollen: Kunde, Broker und Provider. Web Services, die in OWL-S beschrieben sind, können vom Broker gefunden und weitervermittelt werden. Der Kern der Arbeit liegt auf einer Erweiterung der Beschreibungssprache, um einen Ausdruck, der es erlaubt, Prozessmodelle zu erstellen.

Für Schicht (iii) lassen sich auch noch weitere Arbeiten, insbesondere im Bereich der Standards aus dem Gebiet des *Grid Computing*, finden. Das *Grid Laboratory Uniform Environment* (GLUE) [2] ist ein vom Open Grid Forum spezifiziertes Informations- und Datenmodell, das eine einheitliche Beschreibung von Ressourcen im Grid zum Ziel hat. Die aktuelle Version 2.0 von GLUE basiert auf seiner Vorgängerversion 1.3. GLUE 2.0 findet derzeit seine hauptsächliche Verbreitung in der Welt der Grids und es existieren Implementierungen z. B. auf Basis von XML, SQL oder auch LDAP.

Wie bereits aus dem Namen ersichtlich wird, konzentriert sich die Arbeit *WS-Aggregation: Distributed Aggregation of Web Services Data* [7] vor allem auf datenintensive Dienste. Diese Arbeit lässt sich allerdings auch im Kontext der in diesem Buchkapitel behandelten Thematik gut einordnen um die Ebene der komplexen Komponenten des vorgestellten Schichtenmodells zu behandeln. Es wird ein Rahmenwerk beschrieben, das einem Anwender sogenannte *Gateway Services*, die von einem Aggregator nach zu definierenden Kriterien zusammengefügt werden, zur Verfügung stellt. Zudem wird die *Web Service Aggregation Query Language* (WAQL) eingeführt, die es ermöglicht, Templates für Anfragen zu erstellen, und beschreibt, wie die Ausgabedaten zusammengestellt werden.

Zur Auswahl konkreter Smart-App-Komponenten (vgl. Schicht (vii)) bedarf es zusätzlich Beschreibungssprachen, die die Anforderungen einer Applikation an ihre Ausführungsumgebung spezifizieren. Auch in diesem Gebiet finden sich insbesondere Ansätze aus dem Bereich des Grid Computing. Die *Job Submission and Description Language* (JSDL) [3] ist eine XML-basierte Spezifikation des Global Grid Forum. JSDL ist darauf ausgelegt, die Anforderungen eines Jobs, der typischerweise an Höchstleistungsrechner bzw. deren Batch-Scheduler gerichtet, aber nicht darauf beschränkt ist, an seine Rechenressourcen zu beschreiben. Dabei werden neben dem Namen und einer Beschreibung ebenfalls Ressourcenanforderungen wie z. B. benötigter Arbeitsspeicher, benötigte CPU (z. B. Typ und Anzahl) oder das geforderte Betriebssystem angegeben. Neben vielen weiteren Parametern können auch Laufzeitbeschränkungen wie Höchstgrenzen für die Bearbeitungszeit des Jobs oder maximal erlaubter/erwünschter Arbeitsspeicherverbrauch spezifiziert werden.

Das *World Wide Web Consortium* (W3C) bildet mit seiner *W3C Unified Service Description Language Incubator Group* eine Gruppe, die an der Spezifikation der *Unified Service Description Language* (USDL) [5] arbeitet. Ziel von USDL ist es, „eine Sprache zu definieren, die Hauptmerkmale und generische Bestandteile von technischen und wirtschaftlich genutzten Diensten beschreibt, damit diese handelbar werden“ (vgl. <http://www.w3.org/2005/Incubator/usdl/>). Noch hat USDL nicht die Reife und den Status eines Standards erlangt, dennoch finden sich in den bereits existierenden Arbeiten eine Menge brauchbarer Ansätze, die es zu beobachten gilt.

Schlussendlich existieren auch Vorarbeiten im Bereich des Bindens von Smart-App-Komponenten (vgl. Schicht (viii) aus Abb. 23.4). In ihrer Arbeit *CloudSim: A toolkit for Modeling and Simulations of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms* [4] stellen Buyya et al. ein Konzept für ein verteiltes Cloud-Netz basierend auf vorhergehenden Arbeiten im Grid-Umfeld vor. Die Autoren führen ein marktbares Modell ein, das als zentrales Element eine Börse besitzt. Um mit dieser interagieren zu können, braucht jede Cloud einen Koordinator. Sogenannte *Broker*, die zwischen den Kunden (in diesem Fall z. B. ein *Software-as-a-Service*-Anbieter) und dem angebotenen Cloud-Dienst vermitteln, versuchen für die Kunden passende Angebote an der Börse zu finden, die sie dann an die Kunden weitergeben.

## 23.4 Zusammenfassung und Ausblick

Smart Mobile Apps erweitern die Funktionalität von mobilen Endgeräten in Bereiche, die bisher leistungsstarken Rechnern vorbehalten waren. Aufgrund der Beschränkungen von mobilen Endgeräten müssen eben einzelne Teile dieser Anwendungen auf Rechnerstrukturen im Hintergrund abgebildet werden. Mit Hilfe der Cloud ergibt sich eine mächtige Lösung, die derartige Anwendungsszenarien überhaupt erst ermöglicht. Gleichzeitig zeigt sich, dass heutige Cloud-Konzepte nicht ausreichen, sondern aus der Kombination von Smart Mobile Apps mit Clouds neue Fragestellungen entstehen.

In diesem Kapitel wurden diese Fragestellungen aufgelistet, wobei Vergleichbarkeit, Auswahl und Kombinierbarkeit schließlich einen Markt für heutige Cloud-Dienste definieren. Um diesen Markt zu realisieren, wird die Funktion eines Cloud-Brokers vorgestellt. Der Cloud-Broker stellt dabei die Vermittlungsschicht zwischen den Smart Mobile Apps und den Cloud-Diensten dar. Die vielfachen Möglichkeiten zum Kombinieren dieser beiden Dienstarten erfordern einen mehrdimensionalen Optimierungsansatz, der die für den Nutzer günstigste (bezogen auf z. B. Preis, Dienstqualität, ...) auswählt oder dem Nutzer zur Auswahl vorgibt. Dennoch, das Postulat der Notwendigkeit eines solchen Cloud-Brokers gepaart mit dessen Komplexität bringt auch die Forderung nach einer skalierbaren und ausfallsicheren Lösung mit sich, der im Rahmen dieser Arbeit bisher nicht Rechnung getragen wurde. Die zukünftigen Arbeiten am Cloud-Broker werden sich u. a. auch

mit dem verteilten Aufbau und der Interoperabilität von Cloud-Brokern sowie der Kombination von verschiedenen SLAs zu einem auseinandersetzen.

Die in diesem Kapitel beschriebene Lösung stellt erst einmal einen Ansatz dar, der sowohl die notwendige Funktionalität, als auch die erforderliche Flexibilität ermöglicht. Gleichzeitig ist es ein erster Ansatz, der viele neue Möglichkeiten ergibt, die sowohl für die Anbieter von Smart Mobile Apps, als auch für die Cloud-Anbieter interessante Möglichkeiten auftun. So sind für die Auswahl der Dienste Ansätze aus der Betriebswirtschaft denkbar, die neue Geschäftsfelder und -modelle mit sich bringen. Vorstellbar ist ein automatisierter „App-Store“, der von einem Broker durchsucht wird. Die Möglichkeit der automatischen Kombination von Diensten zu neuen komplexeren Diensten oder Workflows ergibt ungeahnte Möglichkeiten in Bezug auf die Funktionalität der Smart Mobile Apps. Den Entwicklern der Apps und der Cloud-Dienste sind keine Grenzen in Bezug auf die Funktion gesetzt, und es wird spannend, diese Möglichkeiten in naher Zukunft im täglichen Einsatz zu sehen.

## Literaturverzeichnis

1. Allan BA, Armstrong R, Bernholdt DE, Bertrand F, Chiu K, Dahlgren TL, Damevski K, Elwasif WR, Epperly TGW, Govindaraju M, Katz DS, Kohl JA, Krishnan M, Kumfert G, Larson JW, Lefantzi S, Lewis MJ, Malony AD, McLennan LC, Nieplocha J, Norris B, Parker SG, Ray J, Shende S, Windus TL, Zhou S (2006) A Component Architecture for High-Performance Scientific Computing. International Journal of High Performance Computing Applications 20:163–202, doi: 10.1177/1094342006064488, <http://portal.acm.org/citation.cfm?id=1125980.1125983>, 26. Oktober 2011
2. Andreozzi S, Burke S, Ehm F, Field L, Galang G, Konya B, Litmaath M, Millar P, Navarro J (2009) GLUE Specification v. 2.0
3. Anjomshoa A, Brisard F, Drescher M, Fellows D, Ly A, McGough S, Pulsipher D, Savva A (2005) Job Submission Description Language (JSIDL) Specification, Version 1.0
4. Calheiros RN, Ranjan R, Beloglazov A, De Rose CAF, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience 41(1):23–50, doi: 10.1002/spe.995, <http://dx.doi.org/10.1002/spe.995>, 26. Oktober 2011
5. Cardoso J, Barros A, May N, Kylau U (2010) Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments
6. Castranova AM, Goodall JL (2010) A generic approach for developing process-level hydrologic modeling components. Environmental Modelling & Software 25:819–825, doi: <http://dx.doi.org/10.1016/j.envsoft.2010.01.003>, 26. Oktober 2011
7. Hummer W, Leitner P, Dustdar S (2011) WS-Aggregation: Distributed Aggregation of Web Services Data. In: 26th ACM SIGAPP Symposium On Applied Computing (SAC), SOAP Track, [http://www.infosys.tuwien.ac.at/staff/hummer/docs/2011\\_sac\\_ws-aggregation.pdf](http://www.infosys.tuwien.ac.at/staff/hummer/docs/2011_sac_ws-aggregation.pdf), 26. Oktober 2011
8. Khasnabish B, Chu J, Ma S, Meng Y, So N, Unbehagen P (2010) Cloud Reference Framework – Internet-Draft
9. Paolucci M, Soudry J, Srinivasan N, Sycara K (2004) A Broker for OWL-S Web Services. In: First International Semantic Web Services Symposium, AAAI Spring Symposium Series

---

# Kapitel 24

## Smart Apps aus der Wolke

### Das Beste aus beiden Welten

Holger Sirtl und Frank Koch

**Zusammenfassung** Mobile Endgeräte wie Smartphones und Tablet-Computer haben in den letzten Jahren starke Verbreitung erlangt. Lange Laufzeiten, Mobilität und einfache Bedienung stellen oft genannte Vorteile dieser Geräte dar, ihre beschränkte Rechenleistung einen der größten Nachteile. Kombiniert mit Cloud Computing ergeben sich jedoch völlig neue Nutzungsmöglichkeiten. Im Zusammenspiel mit mobilen Endgeräten können über die Cloud vernetzte Anwendungen optimiert und gänzlich neue Anwendungstypen geschaffen werden. Hierzu sind durchgängige Entwicklungs- und Technologieplattformen mit Cross-Platform-Werkzeugen, Konzepte zum sicheren Betrieb und Vernetzung von Cloud- und Device-Komponenten erforderlich, um neben der gewünschten Funktionalität auch Sicherheit, Datenschutz und Compliance zu gewährleisten. Der Buchbeitrag gibt einen Überblick über den aktuellen Stand und die Möglichkeiten, mit Hilfe der Microsoft-Plattform effizient Cloud-basierte, verteilte mobile Anwendungen für verschiedene mobile Plattformen zu erstellen und zu betreiben.

#### 24.1 Smart Apps als Tor zu einer vernetzten Welt

IT-Funktionen können heute von einer Vielzahl an Gerätetypen aus genutzt werden. Längst ist der PC nicht mehr der einzige Zugang zu IT. Anwender erwarten, immer und überall auf ihre gespeicherten Daten zugreifen, mit anderen Anwendern kommunizieren und andere IT-Dienste nutzen zu können. Das Einsatzszenario bestimmt das jeweils bestgeeignete Gerät. Mobilien Endgeräten wie Smartphones oder

---

Holger Sirtl  
Microsoft Deutschland GmbH, Unterschleißheim,  
E-mail: holger.sirtl@microsoft.com

Frank Koch  
Microsoft Deutschland GmbH, Unterschleißheim,  
E-mail: frank.koch@microsoft.com

Tablet-Computern kommt hierbei eine wichtige Rolle zu. Sie sind transportabel, dadurch fast immer verfügbar und konnten in den vergangenen Jahren nicht nur ihre Leistungsfähigkeit steigern, sondern auch um gänzlich neue Funktionen erweitert werden.

### **24.1.1 Neue Möglichkeiten mobiler Endgeräte**

Durch neue Ausstattungsmerkmale mobiler Endgeräte öffnen sich gänzlich neue Möglichkeiten für Anwendungen. Einige davon können wie folgt umrissen werden:

- Online-Connectivity – entsprechende Sprach- und Datentarife erlauben es Nutzern, immer und überall „online“ zu sein. Dies kann somit beim Entwurf von Smart Apps angenommen werden. Apps können um entfernte Services erweitert werden und müssen nicht mehr vollständig auf dem Endgerät ausgeführt werden. Beschränkungen der Geräte in Bezug auf Rechenleistung, Speicherkapazität etc. können so zum Teil aufgehoben werden.
- Multitouch – neuartige Eingabemöglichkeiten machen Smart Apps neuen Benutzergruppen zugänglich. Dies erweitert den Anwenderkreis, was wiederum die Attraktivität des Gesamtsystems erhöhen und damit wiederum neue Anwender gewinnen kann.
- Geo-Location – durch die Möglichkeit zur Bestimmung der Position eines Anwenders ergeben sich neue Optionen zur Personalisierung von Anwendungen. Neben dem Benutzerprofil kann auch die Position herangezogen werden um ortsbabhängige Dienste (z. B. Restaurantempfehlungen) anzubieten.
- Lagesensoren – neben der Position kann auch die Lage (Neigungswinkel, Beschleunigung etc.) in Anwendungen verarbeitet werden.
- Aufnahmemöglichkeiten für Audio, Foto und Video – Mediendateien können auf dem Gerät erzeugt und verarbeitet werden. Durch die genannte Connectivity kann Speicherung und Verarbeitung auch auf entfernten Servern erfolgen.

### **24.1.2 Cloud Computing als Rückgrat mobiler Anwendungen**

Cloud Computing hat sich innerhalb weniger Jahre von einem Hype-Thema zu einer ernst zu nehmenden Alternative für die Bereitstellung von IT-Leistungen gewandelt. Noch hat sich in der Literatur allerdings keine einheitliche Definition herausgebildet. Häufig wird der Begriff normativ, d. h. über bestimmte Eigenschaften definiert, die man im Allgemeinen mit Cloud Computing verbindet. Armbrust et al. identifizieren in [1] folgende Aspekte, die neu für die Cloud sind:

- *Die Illusion unendlicher Rechenressourcen, die bei Bedarf zur Verfügung stehen* – damit vereinfachen sich Kapazitätsplanungen deutlich.

- *Die Ausschaltung von Vorab-Investitionen bei Cloud-Anwendern* – die Menge der bereitgestellten Cloud-Ressourcen lässt sich zu jedem Zeitpunkt flexibel an den jeweiligen Bedarf anpassen.
- *Die Möglichkeit, die Nutzung bereitgestellter Ressourcen kurzfristig und rein verbrauchsabhängig zu bezahlen* – damit wird es möglich Kosten zu senken, indem nicht benötigte Ressourcen freigegeben und bei Bedarf wieder hinzugeschaltet werden.

Unter anderem in folgenden Einsatzszenarien verteilter Software-Systeme kann die Cloud ihre durch die genannten Wesensmerkmale geprägten Stärken besonders ausspielen:

- Temporärer Ressourcenbedarf – Durch die Elastizität der Cloud können kurzfristig große Mengen an IT-Ressourcen genutzt werden, die nach der Nutzung wieder deaktiviert werden können. Mobile Endgeräte können so leicht ihre begrenzte Rechenkapazität massiv aufstocken, um größere Rechen- und Speicheraufgaben (z. B. für Bildverarbeitung) auszulagern. Anwendungen wie Photosynth (siehe Abb. 24.1 und [2]) sind Beispiele, in denen dieser Vorteil genutzt wird.
- Hohes mögliches Wachstum der Last – Diese Flexibilität der Ressourcenbereitstellung hat insbesondere dann Vorteile, wenn der Erfolg einer Smart App, d. h. die Zahl der Nutzer und deren Nutzungsverhalten schwer abzuschätzen ist. Mit der Cloud im Hintergrund kann zunächst eine kleine Installation vorgesehen werden und im Erfolgsfall können schnell Ressourcen zugeschaltet werden. Das unternehmerische Risiko kann aufgrund der vereinfachten Kapazitätsplanung erheblich reduziert werden.
- Regelmäßige und unregelmäßige Lastspitzen – Speziell bei Smart Apps, bei denen die Nutzung nicht gleichmäßig über die Zeit verteilt ist (weil sie z. B. abhängig von Tageszeiten oder bestimmten externen Ereignissen ist), ergeben sich durch die flexible Ressourcenbereitstellung Vorteile. IT-Ressourcen müssen nicht mehr entsprechend der prognostizierten Maximallast vorgehalten werden, sondern können im Bedarfsfall schnell hinzugeschaltet werden. Zwischen den Lastspitzen können die Ressourcen entsprechend wieder reduziert werden, um Kosten zu sparen. Xbox Live ist ein solcher Cloud Service, bei dem Lastspitzen zu Zeiten, in denen Anwender Freizeit haben (Wochenende, Abendstunden), auftreten.
- Zugriff immer und überall – Durch die Mobilität der Endgeräte ergeben sich spezielle Anforderungen an IT-Ressourcen, möglichst an beliebigen Orten möglichst zu jeder Tages- und Nachtzeit verfügbar zu sein. Cloud Computing kann durch entsprechende Versprechen hinsichtlich Hochverfügbarkeit, Datenreplikation etc. die Voraussetzungen schaffen, IT-Funktionen für Smart Apps zuverlässig, weltweit und durchgängig bereitzustellen.

## 24.2 Szenarien für Smart Apps

Smart Apps führen die Flexibilität mobiler Clients mit der Leistungsfähigkeit von Cloud Services zusammen. Diese Kombination macht faszinierende Anwendungssysteme möglich, von denen einige hier kurz erwähnt werden sollen.

### 24.2.1 Spiele

Immer und überall nutzbare Smart Apps sind ideale Softwarekonzepte für den Spielmarkt. Microsoft hat mit Einführung der Windows-Phone-Plattform diese in die Xbox-LIVE-Umgebung, den zur Xbox gehörenden Cloud Service, integriert. Dies ermöglicht es, auf Handy-Spiele zuzugreifen oder rund um die Uhr Funktionen von Xbox LIVE nutzen zu können, so dass Anwender stets mit Freunden in Kontakt bleiben und gemeinsam mit anderen spielen können (siehe [3]).

Diese Möglichkeit, im Falle von Xbox LIVE, mit bereits über 35 Millionen registrierten Anwendern spielen zu können, macht Online-Spiele in ganz neuen Dimensionen möglich. Hierfür hat sich der Begriff „Massively Multiplayer Online Game“ (MMOG) etabliert (siehe [4] und [5]). Diese erfreuen sich wachsender Beliebtheit. Unter [6] findet sich eine umfangreiche Liste aktueller MMOGs, die das Wachstum des Marktes dieser Kategorie von Smart Apps zeigt.

Mit der CipSoft GmbH [7] hat sich in diesem Umfeld auch ein deutscher Softwarehersteller erfolgreich positioniert.

### 24.2.2 Edutainment

Spielekonzepte halten auch im Weiterbildungsbereich Einzug. Ein Beispiel hierfür ist die Quiz-Anwendung „Quizza“ (siehe Abb. 24.1 und [8]), bei der ein Spieler Fragen beantworten muss und entsprechend seiner Antworten bewertet wird. Vernetzt mit der Cloud sind ständig neue Inhalte verfügbar. Darüber hinaus kann der Spieler gemeinsam mit anderen Spielern Fragen beantworten.

Ein weiteres Beispiel ist Sportmanager Fußball (siehe Abb. 24.1 und [9]), bei dem Spieler als Manager von Fußballmannschaften auftreten, die gegen Mannschaften anderer Spieler antreten können.

### 24.2.3 Medienverarbeitung

Medieninhalte, die über mobile Geräte erzeugt werden, können gesteuert durch den mobilen Client in der Cloud verarbeitet werden, um nicht auf die begrenzten Ressourcenkapazitäten des mobilen Geräts limitiert zu sein. Ein Beispiel hierfür ist Pho-

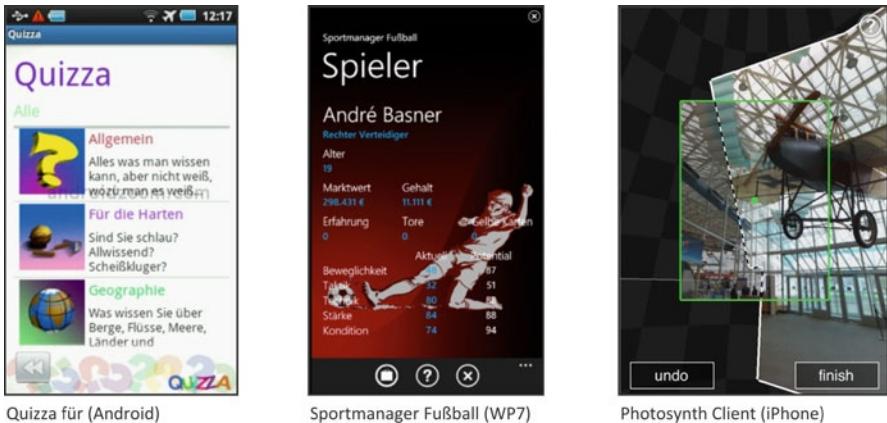


Abb. 24.1 Smart Apps für Edutainment und Medienverarbeitung

tosynth, ein Cloud Service, mit dessen Hilfe aus einzelnen Fotos dreidimensionale Modelle eines Objektraums erzeugt werden können. Entsprechende Apps sind für verschiedene Mobilplattformen verfügbar (siehe Abb. 24.1 sowie [10] und [2]).

#### 24.2.4 Augmented Reality

Mit Hilfe der Sensorik der mobilen Geräte können Bilder oder Videos, die über die Geräte gewonnen werden, mit Zusatzinformationen aus der Cloud oder virtuellen Objekten mittels Einblendung/Überlagerung ergänzt werden. Dieses als „Augmented Reality“ bezeichnete Konzept ist ein weiteres Szenario für Smart Apps. Microsoft stellt für Windows Phone Werkzeuge zur Verfügung, mit deren Hilfe entsprechende Silverlight-basierte Smart Apps erstellt werden können (siehe [11]).

### 24.3 Technische Grundlagen für Smart Apps

Das technische Fundament für Smart Apps wird durch das Zusammenspiel von Client-Anwendungen für mobile Endgeräte und leistungsfähigen Cloud Services gebildet. Die Vorteile der mobilen Einsatzszenarien und quasi unbegrenzten IT-Ressourcen müssen dabei gegen die Nachteile der nötigen Connectivity abgewogen werden. Vielfach können hybride Ansätze mit eingeschränkter lokaler Funktionalität für den Offline-Einsatz interessante Kompromisse darstellen. So kann eine Smart App für den Geschäftsbereich auf dem mobilen Gerät z. B. die wichtigsten Kunden-daten wie Adresse und Telefonnummer, Ansprechpartner mit Bild und aktuelle Kun-denanfrage vorhalten. Über die Cloud könnten zusätzlich noch Informationen aus

CRM- und ERP-Systemen für die Kundenhistorie aufbereitet sowie wichtige, aktuelle Wirtschaftsnachrichten für das zu besuchende Unternehmen angezeigt werden, um sich optimal auf den Kundentermin vorzubereiten und Geschäftsentscheidungen fundierter begründen zu können.

### **24.3.1 Cloud Services**

Aus den oben beschriebenen Wesensmerkmalen der Cloud geht noch nicht unmittelbar hervor, welche konkreten Dienste über den Cloud-Ansatz bereitgestellt werden. Betrachtet man Cloud Computing aus technischem Blickwinkel, so lassen sich Cloud Services in Anlehnung an Ausführungen in [12] und [13] in die folgenden drei Schichten kategorisieren.

- *Infrastructure as a Service* – infrastrukturnahe Dienste wie Storage, Rechenleistung, Virtualisierung, die von Administratoren beispielsweise dazu genutzt werden können, eigene virtuelle Maschinen oder Speicherdienste in der Cloud zuverlässig zu betreiben.
- *Platform as a Service* – Dienste, die zur Entwicklung und dem Betrieb von Cloud Services genutzt werden können. Administrative Fragen zum Aufbau und Betrieb hierzu benötigter virtueller Maschinen bleiben im Verantwortungsbereich des Cloud Anbieters.
- *Software as a Service* – Anwendungssysteme und -dienste, die von Benutzern ohne Installationsaufwand aus der Cloud genutzt werden können, um informationsverarbeitende Aufgaben zu erledigen.

Für Entwickler von Smart Apps sind insbesondere die Dienste aus dem PaaS-Umfeld interessant, da sie die Basis für Entwicklung und Bereitstellung eigener in der Cloud betriebene App Services, also eigener SaaS-Dienste, darstellen.

#### **24.3.1.1 Cloud Services von Microsoft**

Als Technologieanbieter ist Microsoft auf allen Ebenen mit entsprechenden Angeboten vertreten. Diese sind in Tabelle 1 aufgelistet.

#### **24.3.1.2 Die Windows Azure Platform**

Innerhalb Microsofts Cloud-Plattform stellt die Windows Azure Platform das Platform-as-a-Service-Angebot dar, d. h. diese Dienste können für Entwicklung und Bereitstellung Cloud-basierter App Services eigener Smart-App-Lösungen genutzt werden. Azure bietet drei Gruppen von Cloud Services an, die einzeln oder in Kombination genutzt werden können.

**Tabelle 24.1** Cloud Services von Microsoft

Kategorie	Cloud Service	Beschreibung
SaaS	Office 365	Office-Client- und Server-Produkte (Exchange, SharePoint, Lync) als Cloud Service bereitgestellt
	Windows Live	Cloud Services für Privatanwender für E-Mail, Kalender, Speicher etc.
	Microsoft Intune	Managementdienste zur Verwaltung eigener Hardware-Ressourcen
PaaS	Windows Azure Platform	Cloud Services für Entwickler (Laufzeitumgebung, Speicher etc.) zu Erstellung und Betrieb eigener Cloud-Anwendungen
	Windows Azure Platform Appliance	Azure, das – auf spezieller Hardware vorinstalliert – von Dienstleistern oder Unternehmenskunden betrieben werden kann
IaaS	Global Foundation Services	Infrastrukturdiene, die in Microsofts Rechenzentren bereitgestellt werden

Windows Azure ist das Betriebssystem für Microsofts Cloud. Es stellt mit dem Windows Azure Compute Service eine Ausführungsumgebung für Cloud Services zur Verfügung. Mit dessen Hilfe können selbst entwickelte Anwendungen in Microsofts Cloud-Infrastruktur betrieben werden. Der Dienst bietet ein automatisiertes Management von Cloud Services mit Überwachung der ausgeführten Cloud Services, das Patch- und Update-Management für die eingesetzten virtuellen Maschinen sowie die automatische, lastabhängige Skalierung der Umgebung. Grundsätzlich lässt sich auf Windows Azure jede Software ausführen, die auch auf einem Windows-Server ausgeführt werden kann. Dies ist unabhängig von der Implementierungstechnologie. So können sowohl .NET- als auch Java- oder PHP-basierte Anwendungen auf Azure betrieben werden. Windows Azure stellt darüber hinaus Speicherdiene (Windows Azure Storage Services) zur Verfügung. In diesen können strukturierte Daten in Form von Entity Sets (Table Storage) und große Binärdaten (Binary Large Objects = BLOBs) im sogenannten BLOB-Storage abgelegt werden. Asynchroner Nachrichtenaustausch kann über Windows Azure Queues abgewickelt werden. Mit Azure Drives stellt die Plattform auch ein NTFS-basiertes Dateisystem zur persistenten Speicherung von Daten zur Verfügung. Die Speicherdiene können über Standardschnittstellen (SOAP, REST, XML), also von jeder Technologie, die diese Standards unterstützt, genutzt werden.

Wer ein relationales Datenbanksystem in der Cloud benötigt, findet mit SQL Azure den passenden Dienst. Dieser unterstützt ein Subset der Funktionen eines SQL Servers und stellt seine Funktionen über einen TDS-Endpunkt (TDS = Tabular Data Stream) zur Verfügung. Dieser ist kompatibel zu entsprechenden Endpunkten eines SQL Servers, womit die Funktionen von SQL Azure über die gleichen Technologien und Werkzeuge genutzt werden können, die auch für einen SQL Server zum Einsatz kommen. Mit Hilfe des SQL Azure Sync Service kann eine Cloud-Datenbank mit anderen Datenbanken, die Daten beispielsweise auf mobilen Geräten

speichern, synchronisiert werden. Damit ist der Aufbau offline-fähiger Smart Apps möglich (siehe auch Abb. 24.6).

Die Windows Azure Platform AppFabric ist ein Kommunikationsdienst, über den verteilte Webservices, unabhängig von Implementierungstechnologie und Ausführungsort über einen Service-Bus kommunizieren können. Dabei können die beteiligten Kommunikationspartner den Nachrichtenaustausch absichern und Teile der Authentifizierungslogik an den Access Control Service auslagern, der Funktionen zur Claims-basierten Zugriffskontrolle bereitstellt. Damit steht eine leistungsfähige Integrationstechnologie für die Kommunikation zwischen mobilen Clients untereinander und mit der Cloud zur Verfügung.

### **24.3.2 Mobile Plattformen**

Der Markt für mobile Plattformen ist aufgrund des hohen Innovationstempos der einzelnen Anbieter ständigen Umwälzungen unterworfen, die Auswirkungen auf Marktanteile und damit die Bedeutung der einzelnen Plattformen haben. Das Marktforschungsinstitut IDC wagt in [14] eine Prognose der Entwicklung der wichtigsten Plattformen. Für das Jahr 2015 werden als dominante Mobilplattformen Android (45,4 % Marktanteil), Windows Phone 7 (20,9 %) und iOS (15,3 %) angesehen. Mit einer Anwendung, die auf diesen drei Plattformen verfügbar ist, kann also mehr als 80 % des Marktes adressiert werden.

Wenngleich derartige Prognosen immer mit einer gewissen Unsicherheit behaftet sind, ist doch zu erwarten, dass es zu einer Aufteilung des Marktes auf mehrere Plattformen kommen wird. Entsprechend wichtig ist es für Anbieter von Smart Apps, mehrere Plattformen zu bedienen, um eine möglichst hohe potenzielle Nutzerzahl zu erreichen. Dabei steht der Entwickler vor der Wahl, entweder für jede mobile Plattform jeweils einen eigenen nativen Client zu entwickeln (für iPhone mit Objectif-C, für Android mit Java, für Windows Phone mit .NET/Silverlight), oder eine plattformübergreifende Technologie wie HTML5 zu wählen und das Frontend als Web-Anwendung bereitzustellen, die im jeweiligen Endgerät über einen Browser angezeigt wird. Beide Ansätze bieten Vor- und Nachteile, wie im Folgenden aufgezeigt wird.

## **24.4 Entwicklung Cloud-basierter Smart Apps**

Beim Entwurf einer Anwendung, die Anwendungskomponenten auf mobilen Endgeräten mit Komponenten aus der Cloud kombiniert, muss in einem ersten Schritt die Verteilung der Anwendungsbestandteile auf die beiden Ausführungsumgebungen festgelegt werden. Wie diese Verteilung konkret aussieht, hängt vom jeweiligen Anwendungsfall ab. In einem Extremfall liegt die Anwendung primär als SaaS-Lösung in der Cloud, und das mobile Endgerät dient nur der Anzeige mittels eines

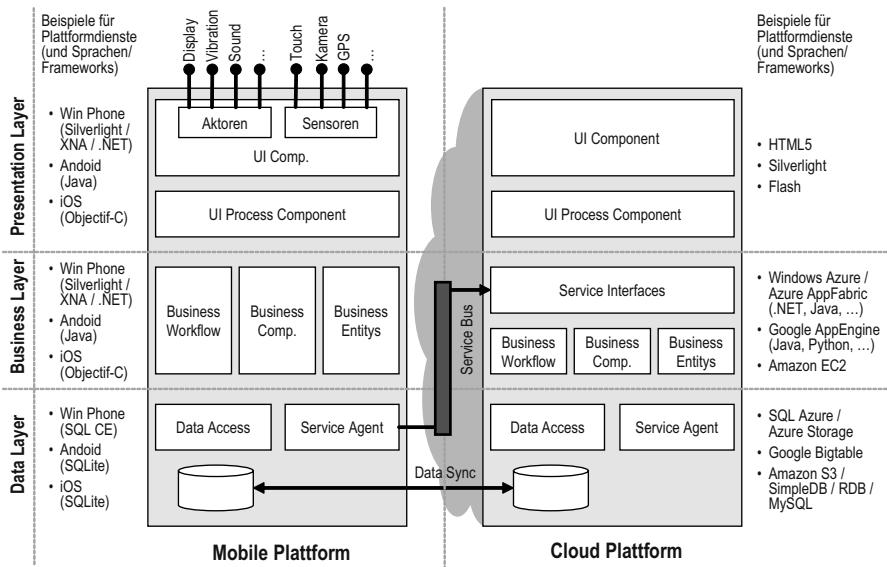


Abb. 24.2 Interaktion einer mobilen Anwendung mit der Windows Azure Platform

Internet-Browsers. In einem anderen Fall kann die Smart App als native Anwendung auf dem mobilen Gerät installiert werden. Aus der Cloud werden dann nur punktuell einzelne Dienste (z. B. zur Kommunikation oder Datenspeicherung) genutzt.

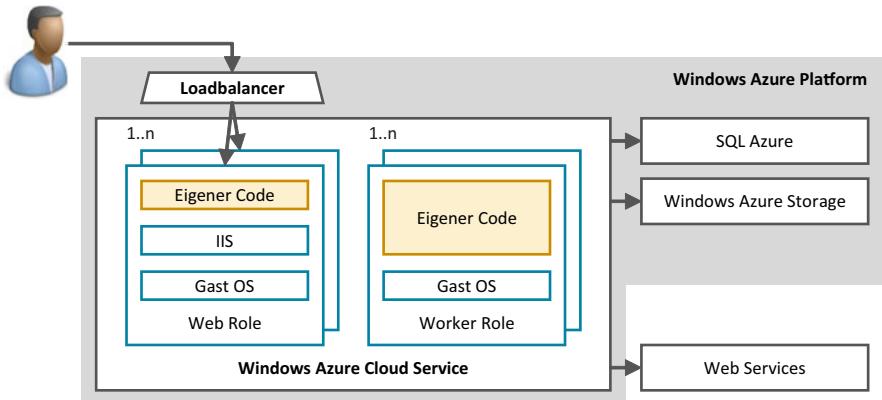
Abbildung 24.2 zeigt eine aus der in [15] verwendeten Basisarchitektur abgeleitete Referenzarchitektur für Smart Apps, bei der Teile der Anwendung auf einer mobilen Plattform und Teile in der Cloud betrieben werden.

Auf beiden Seiten kann eine Unterteilung in die üblichen Schichten (Data, Business und Presentation Layer) erfolgen. In konkreten Anwendungsszenarien müssen nicht alle dort abgebildeten Komponenten vorhanden sein. In der Abbildung sind darüber hinaus Beispiele für verfügbare Plattformdienste und Sprachen verschiedener Anbieter aufgeführt. In den Implementierungsalternativen der folgenden Abschnitte wird die Windows Azure Platform stellvertretend für Cloud-Plattformen verwendet.

#### 24.4.1 Implementierung Windows-Azure-basierter Services

Ein Windows-Azure-basierter Cloud Service ist durch eine Service-Grenze sowie die enthaltenen Service-Komponenten definiert. Der konzeptionelle Aufbau ist in Abb. 24.3 zu sehen.

Jegliche Kommunikation von außen mit dem Cloud Service erfolgt über einen Loadbalancer. Nur explizit vom Entwickler geöffnete Ports sind zum Empfang freigegeben. Der Service selbst kann beliebig mit der Außenwelt, d. h. mit SQL Azu-



**Abb. 24.3** Aufbau eines Windows-Azure-basierten Cloud Service

re, Windows Azure Storage oder anderen Web Services kommunizieren. Service-Komponenten werden in jeweils eigenen virtuellen Maschinen ausgeführt und werden „Roles“ genannt. Dabei wird unterschieden zwischen Web Roles und Worker Roles. In Web Roles ist ein Internet Information Server (IIS) vorinstalliert. Sie eignen sich deshalb für die Implementierung von Webseiten und Webservices. Worker Roles fehlt dieser IIS. Worker Roles werden zur Implementierung von Hintergrundprozessen bzw. in Szenarien verwendet, in denen der Entwickler einen eigenen App-Server (z. B. Tomcat für Java Apps) betreiben möchte.

Sowohl für Visual Studio als auch für Eclipse sind kostenlose Erweiterungen verfügbar, die die Erstellung Azure-basierter Cloud Services stark vereinfachen. Entwickler können damit bestehende Werkzeuge als auch bekannte Programmiersprachen weiterverwenden.

Auf die Dienste kann direkt über den Loadbalancer zugegriffen werden. Alternativ können diese Dienste aber auch bei einem von der Windows Azure AppFabric bereitgestellten Internet Service Bus registriert und die Kommunikation über den Bus abgewickelt werden. In diesem Fall kann die AppFabric weitere Dienste wie Nachrichtenpufferung und -verteilung, Caching, Authentifizierung etc. einbringen.

#### 24.4.2 Zugriff auf Windows-Azure-basierte Apps und App Services

Für die Implementierung des Clients einer Smart App stehen zum heutigen Zeitpunkt zwei grundsätzliche Alternativen zur Verfügung, die in Abb. 24.4 zu sehen sind.

Im ersten Szenario (1) greift der Anwender über einen Browser per HTTP auf die Anwendung auf Windows Azure zu. Die Anwendungslogik wird primär in der Cloud bereitgehalten. Das mobile Endgerät dient also nur als Anzeigemedium für Webseiten. Diese Seiten können beispielsweise über HTML5 implemen-

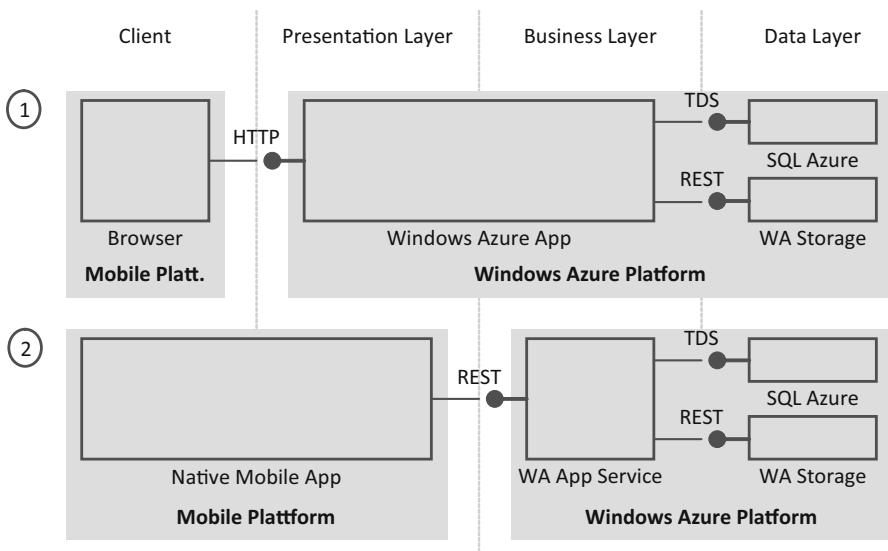


Abb. 24.4 Windows-Azure-basierte Apps und App Services

tiert sein. Vorteil dieses Szenarios ist es, dass die Anwendungsoberfläche auf allen HTML5-fähigen Anzeigesystemen sofort dargestellt werden kann. Die Entwicklung der Smart App als Web-Anwendung unterscheidet sich nicht wesentlich von der Entwicklung klassischer Web-Anwendungen. Entsprechend können bestehende Entwicklerwerkzeuge und -technologien weiterverwendet werden. Eine Abhängigkeit zu einer Mobilplattform wird vermieden. Allerdings beschränken sich die Möglichkeiten hierbei auf den Funktionsumfang von HTML5.

Im zweiten Szenario (2) übernimmt eine native Anwendung die Anzeige der Oberfläche auf dem Endgerät. Bei der Entwicklung nativer Clients (Szenario (2) in Abb. 24.4) kommen spezielle Entwicklerwerkzeuge für die jeweilige Mobilplattform zum Einsatz. Im Falle von Windows Phone 7 können Erweiterungen für Expression Studio bzw. Visual Studio eingesetzt werden, um Smart Apps zu entwickeln. Im Fall von Visual Studio kann also ein einziges Werkzeug sowohl zur Entwicklung des Clients als auch der Azure-basierten Cloud-Komponenten verwendet werden. Je nach Anwendungstyp kann der Windows Phone 7 Client entweder auf Basis von Silverlight oder Microsofts XNA Framework implementiert werden. Silverlight erlaubt die Entwicklung leistungsfähiger Oberflächen für Smart Apps. XNA kommt bei der Umsetzung von 2D- bzw. 3D-basierten Spielen zum Einsatz. Speziell für eine bestimmte Mobilplattform implementiert, können auch spezielle Funktionen und Ausstattungen einzelner Endgeräte genutzt werden, wodurch die Leistungsfähigkeit der Anwendung steigen kann. Über REST-Schnittstellen lassen sich auch hier Windows-Azure-basierte App Services integrieren. Bei dieser Variante müssen gegebenenfalls für alle relevanten Mobilplattformen eigene Client Apps geschrieben werden.

Des Weiteren stellt Microsoft Windows Azure Toolkits für Android und iOS zur Verfügung.

### 24.4.3 Daten in der Cloud

Neben der Bereitstellung von Anwendungslogik kann die Cloud auch als Datenspeicher genutzt werden. Dies hat den Vorteil, dass die Daten von dort aus vielen Nutzern von Smart Apps zur Verfügung gestellt werden bzw. Datenaustausch zwischen den Nutzern erfolgen kann. Abbildung 24.5 zeigt die Möglichkeiten, die Azure in diesem Zusammenhang bietet.

In Szenario (3) greift eine native Smart App auf SQL Azure als Datenspeicher zu. Die Kommunikation mit der Cloud-Datenbank kann dank der Schnittstellen-Kompatibilität mit SQL Server über bekannte Zugriffstechnologien (z. B. ADO.NET aus C#, JDBC aus Java, ...) erfolgen, die die Zugriffe auf TDS-Befehle abbilden. Soll zu Datenablage bzw. -austausch Windows Azure Storage genutzt werden, wie in Szenario (4) zu sehen, kann auf die REST-Schnittstelle des Storage zurückgegriffen werden.

Eine Smart App, bei der Datenspeicherung ausschließlich in der Cloud erfolgt, ist für ein Funktionieren von einer bestehenden Netzwerkverbindung in die Cloud abhängig, wenn Datenzugriffe erfolgen. Bei unterbrochener Verbindung stehen entsprechende Funktionen also nicht zur Verfügung. Soll ein unterbrechungsfreier Betrieb gewährleistet werden, können Daten aus SQL Azure mit Hilfe des Sync Service mit lokalen Datenbanken, die auf dem mobilen Client laufen, synchronisiert werden. Damit können die Smart Apps mit den lokalen Datenbanken arbeiten, die auch bei unterbrochener Verbindung zu SQL Azure zur Verfügung stehen, und ihre Daten bei wiederherstellter Netzwerkverbindung mit der Cloud-Datenbank syn-

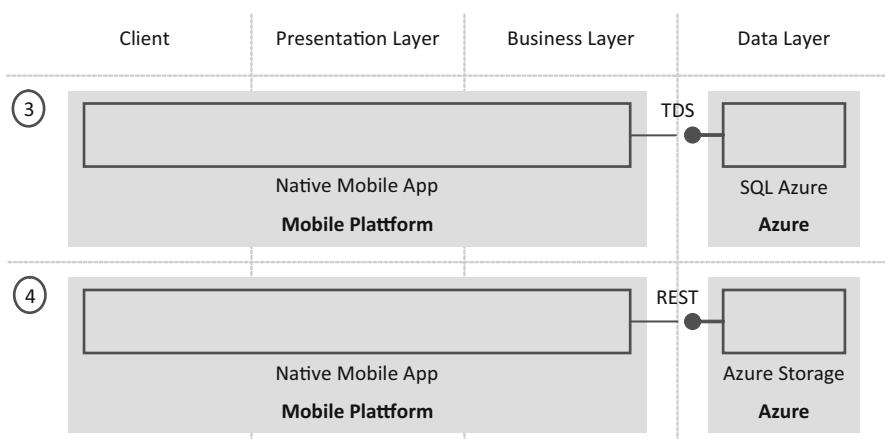


Abb. 24.5 Datenspeicherung in der Cloud

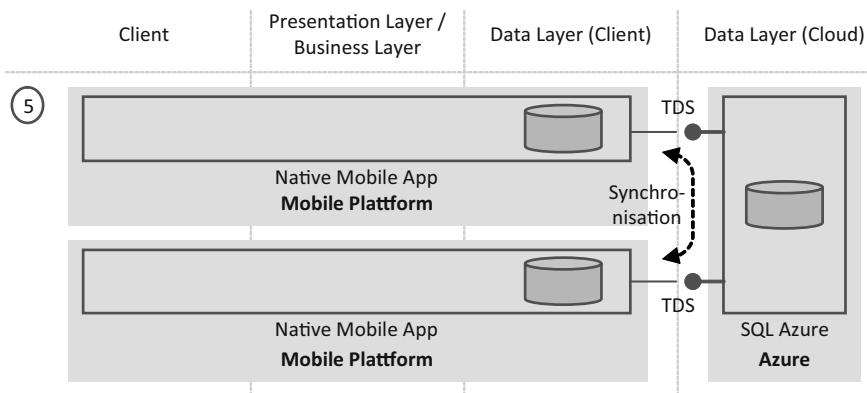


Abb. 24.6 Datensynchronisation über die Cloud

chronisieren, um sie anderen Nutzern (die die Daten dann ihrerseits auf ihre lokalen Datenbanken synchronisiert bekommen) zu teilen. Das Architekturkonzept ist in Abb. 24.6 zu sehen.

Da der Synchronisationsmechanismus mit SQL Azure auf dem Microsoft Sync Framework basiert, kann eine Synchronisationsbeziehung mit jedem Datenbanksystem, für das es entsprechende Adapter gibt, etabliert werden.

#### 24.4.4 Kommunikationsdienste in der Cloud

Die Cloud kann auch als Kommunikations- und Integrationsplattform genutzt werden. Die Windows Azure Platform stellt hierzu die Windows Azure AppFabric zur Verfügung. Mit dieser ist es unter anderem möglich, App Services, die im eigenen Rechenzentrum betrieben werden, über einen Service Bus in Smart Apps zu integrieren. Das Architekturkonzept hierzu ist in Abb. 24.7 zu sehen.

Ein solcher Aufbau einer Smart App ist immer dann sinnvoll bzw. erforderlich, wenn die zu integrierenden App Services selbst nicht in der Cloud liegen dürfen (z. B. aus Gründen der Compliance), die Integration aber über eine leistungsfähige

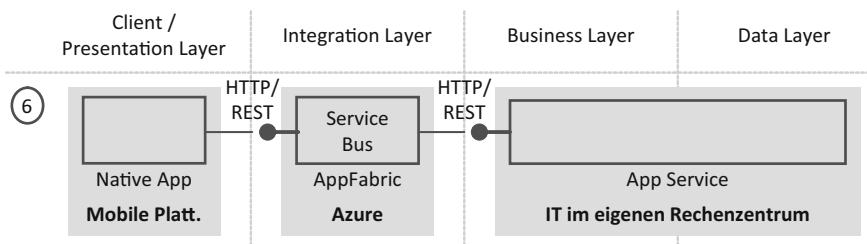


Abb. 24.7 Integration eigener Backend-Systeme

ge, skalierbare Cloud-Plattform erfolgen soll. Nachdem in diesem Szenario die Geschäftslogik und Datenspeicherung im eigenen Rechenzentrum verbleibt, können auch individuelle Wünsche hinsichtlich Sicherheit (z. B. Verschlüsselung) und Datenschutz erfüllt werden.

## 24.5 Fazit

Smart Apps, bei denen mobile Anwendungsclients Cloud Services integrieren, erfreuen sich immenser Beliebtheit. Durch Kombination der Vorteile mobiler Anwendungen mit denen der Cloud können leistungsfähige Anwendungssysteme geschaffen werden, die quasi immer und überall genutzt werden können. Microsofts Technologieplattform bietet Anwendungsentwicklern hier maximale Flexibilität bei der Entscheidung, welche Anwendungskomponenten in der Cloud und welche auf dem mobilen Endgerät betrieben werden. Dabei stehen Entwicklerwerkzeuge sowohl für .NET als auch für Java, PHP und weitere zur Verfügung. Damit sind die Voraussetzungen geschaffen, leistungsfähige Smart Apps zu implementieren und dabei einen maximalen Nutzerkreis zu adressieren.

## Literaturverzeichnis

1. Armbrust M et al. (2009) Above the Clouds: A Berkeley View of Cloud Computing, Berkeley. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>. Accessed 29 August 2010
2. Microsoft Corp. (2011) Photosynth. <http://www.photosynth.net/>. Accessed 16 June 2011
3. Microsoft Corp. (2011) Xbox Live für Windows Phone. <http://www.xbox.com/de-DE/live?mobile>. Accessed 16 June 2011
4. Tommy Palm (2003) The Birth Of The Mobile MMOG. [http://www.gamasutra.com/resource\\_guide/20030916/palm\\_01.shtml](http://www.gamasutra.com/resource_guide/20030916/palm_01.shtml). Accessed 16 June 2011
5. Wikipedia (2011) Massively Multiplayer Online Game. [http://en.wikipedia.org/wiki/Massively\\_multiplayer\\_online\\_game](http://en.wikipedia.org/wiki/Massively_multiplayer_online_game). Accessed 16 June 2011
6. Cyber Creations Inc. (2011) MMORPG Gamelist – All Listed Games. <http://www.mmorpg.com/gamelist.cfm>. Accessed 16 June 2011
7. CipSoft GmbH (2011) CipSoft Online Entertainment. <http://www.cipsoft.com/home/>. Accessed 16 June 2011
8. Vela Entertainment Studios (2011) Quizza Free. [http://www.androidzoom.com/android\\_games/brain\\_puzzle/quizza-free\\_xaub.html](http://www.androidzoom.com/android_games/brain_puzzle/quizza-free_xaub.html). Accessed 16 June 2011
9. Vela Entertainment Studios (2011) Sportmanager Fußball. <http://www.sportmanager-fussball.de/>. Accessed 16 June 2011
10. Apple iTunes (2011) Photosynth for iOS, <http://itunes.apple.com/us/app/photosynth/id430065256?mt=8>. Accessed 16 June 2011
11. Microsoft Corp. (2011) Silverlight and Windows Phone Augmented Reality Toolkit. <http://slartoolkit.codeplex.com/>. Accessed 16 June 2011
12. Baun C et al. (2010) Cloud Computing – Web-basierte dynamische IT-Services, Springer-Verlag, Berlin
13. Rhoton J (2010) Cloud Computing Explained, Recursive Press, Breinigsville, PA USA

14. IDC (2011) IDC Forecasts Worldwide Smartphone Market to Grow by Nearly 50 % in 2011.  
<http://www.idc.com/getdoc.jsp?containerId=prUS22762811>. Accessed 16 June 2011
15. Microsoft Corp. (2009) .NET Application Architecture Guide, 2. Aufl, Microsoft Press, Redmond

---

# Kapitel 25

## Business Web: Cloud-basierte Flexibilisierung und Mobilisierung von Geschäftsprozessen

Uwe Kubach, Ralf Ackermann, Michael Ameling, Volkmar Lotz, Daniel Oberle und Vasco Alexander Schmidt

**Zusammenfassung** In der jüngeren Vergangenheit haben zwei Trends die Entwicklung von Geschäftsanwendungen wesentlich beeinflusst. Zum einen hat der Wandel von monolithischen Anwendungen hin zu Service-orientierten Architekturen (Service Oriented Architectures - SOA) ein neues Maß an Flexibilität gebracht, die es ermöglicht, bestehende Dienste in neuen Geschäftsprozessen schnell und einfach wiederzuverwenden. Zum anderen haben Apps und App-Stores den Vertrieb, die Installation und Erweiterung von Software stark vereinfacht. Hier stehen bislang einfache, kleine, voneinander unabhängige Anwendungen im Vordergrund.

In diesem Beitrag stellen wir die Business-Web-Initiative der SAP-Forschung vor, die darauf zielt, die Vorteile beider Trends in einer neuen cloud-basierten Plattform, dem Business Web, zu vereinen. Dabei geht das Business Web weit über den funktionalen Umfang heutiger App-Stores hinaus, indem es die sichere, flexible und echtzeitfähige Zusammenarbeit aller Geschäftspartner entlang einer vollständigen Wertschöpfungskette ermöglicht. Ein besonderes Augenmerk liegt dabei

---

Uwe Kubach  
SAP AG, Walldorf,  
E-mail: uwe.kubach@sap.com

Ralf Ackermann  
SAP AG, Walldorf,  
E-mail: ralf.ackermann@sap.com

Michael Ameling  
SAP AG, Walldorf,  
E-mail: michael.ameling@sap.com

Volkmar Lotz  
SAP Labs France SAS, Sophia-Antipolis, Frankreich,  
E-mail: volkmar.lotz@sap.com

Daniel Oberle  
SAP AG, Walldorf,  
E-mail: d.oberle@sap.com

Vasco Alexander Schmidt  
SAP AG, Walldorf,  
E-mail: vasco.alexander.schmidt@sap.com

auf der Vereinfachung der Kooperation der Geschäftspartner, dem mobilen Zugang zur Business-Web-Plattform und zu den darauf angebotenen Diensten sowie deren einfacher Kombinierbarkeit.

## 25.1 Motivation

Um der steigenden Komplexität ihrer Produkte, Dienstleistungen und Lösungen bei gleichzeitig immer kürzer werdenden Innovationszyklen zu entgegnen, ziehen sich viele Unternehmen auf ihre Kernkompetenzen zurück. Anstatt einen großen Teil der Wertschöpfungskette selbst abzudecken, reduzieren sie ihre Fertigungstiefe und kooperieren vorzugsweise in Geschäftsnetzwerken und Partner-Eco-Systemen. Entscheidend für den Erfolg des einzelnen Unternehmens wird dabei mehr und mehr die Integrität und Funktionsfähigkeit seines Geschäftsnetzwerks oder Eco-Systems. Beispiele für die Abhängigkeit selbst renommiertester Unternehmen von ihrem Geschäftsnetzwerk sind diverse Rückrufaktionen in der Automobilindustrie, die häufig auf eine unzureichende Qualitätssicherung bei Zulieferern zurückzuführen sind. Der „Plattformkrieg“ bei Smartphones wird weniger über die technischen Fähigkeiten der Plattformen entschieden als vielmehr über die Größe des Eco-Systems, das Apps für die jeweilige Plattform anbietet.

Ganz wesentlich für die Funktionsfähigkeit dieser meist sehr dynamischen Geschäftsnetzwerke ist unter anderem eine entsprechende Unterstützung unternehmensübergreifender Geschäftsprozesse durch eine flexible, einfach zu rekonfigurierende IT-Infrastruktur. Ein entscheidender Fortschritt in dieser Richtung war der Paradigmenwechsel hin zu service-orientierten Architekturen. Sie kapseln Funktionalitäten einzelner Prozessschritte in Dienste, die über standardisierte Schnittstellen angesprochen werden und somit einfach und schnell zu neuen oder veränderten Geschäftsprozessen kombiniert werden können.

Ebenso schnell wie die Geschäftsnetzwerke verändern sich zum Teil die Unternehmen selbst. Sie müssen ggf. schnell wachsen oder sich in Krisenzeiten verschlanken können sowie Übernahmen und Zusammenschlüsse bewältigen. Ihre MitarbeiterInnen sind zunehmend mobil und müssen von unterwegs Geschäftsprozesse bearbeiten können. Eine erfolgreiche IT-Infrastruktur muss diese Mobilität und die häufigen Wechsel in den Unternehmensstrukturen effizient unterstützen. Dies erklärt den Erfolg flexibler Software-Vertriebsmodelle, wie etwa Software-as-a-Service (SaaS), bei denen Software nur noch gemietet wird. Technisch wird diese überwiegend betriebswirtschaftliche Innovation begleitet von einer deutlichen Vereinfachung in der Installation der Software-Services. Noch weiter getrieben wird diese Vereinfachung, allerdings bislang fast ausschließlich im Endkonsumentenbereich, in App-Stores. Hier werden die Apps nur noch heruntergeladen und können i.d.R. ohne jeden weiteren Installationsaufwand direkt auf der sehr häufig mobilen Zielplattform gestartet werden. Diese Einfachheit geht jedoch zu Lasten der Kombinierbarkeit und Wiederverwendbarkeit einzelner Apps. Die Apps sind meist in sich geschlossene, voreinander unabhängige, kleine Anwendungen, die in der Regel nativ programmiert und somit stark plattformabhängig sind.

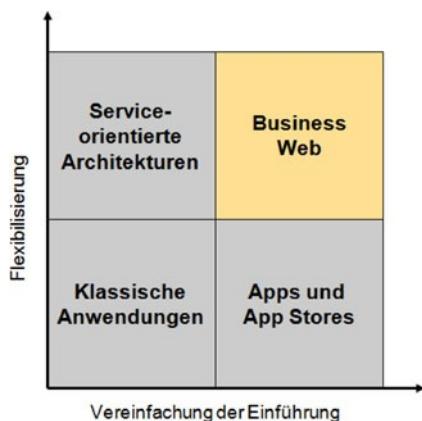


Abb. 25.1 Positionierung des Business Web aus Software-Sicht

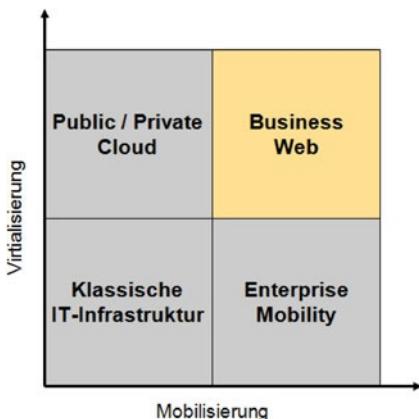


Abb. 25.2 Positionierung des Business Web aus Infrastruktur-Sicht

Ziel der Business-Web-Initiative, die die SAP-Forschung gemeinsam mit ihren Partnern Deutsche Telekom, Ericsson, GlobalLogic, HP und Intel verfolgt, ist es, eine Plattform zu schaffen, die die Einfachheit der Software-Installation von Apps und die Flexibilität von service-orientierten Architekturen vereint (s. Abb. 25.1). Zusätzlich zu dieser auf Software fokussierten Zielsetzung adressiert das Business Web Fragen der IT-Infrastruktur. Hier geht es im Wesentlichen um die effiziente Unterstützung der fortschreitenden Virtualisierung von IT-Infrastrukturen durch Cloud Computing und der zunehmenden Mobilisierung von Geschäftsprozessen (s. Abb. 25.2). Schließlich spielt die Integration unterschiedlicher Informationsquellen aus Enterprise-Systemen, dem Internet der Dinge sowie dem World Web Web, insbesondere aus Web 2.0-Anwendungen und sozialen Netzwerken, eine weitere bedeutende Rolle innerhalb der Business-Web-Initiative.

Somit wird das Business Web eine Plattform sein, die optimal auf die Bedürfnisse von Unternehmen und Geschäftsnetzwerken abgestimmt ist. Diese Bedürfnisse sind zunehmend durch das hochdynamische Geschäftsumfeld und die Mobilität der MitarbeiterInnen geprägt. Neben der technischen Lösung werden im Projekt auch betriebswirtschaftliche Fragen wie Preismodelle und der Aufbau von Wertschöpfungsnetzwerken diskutiert.

Im Folgenden gehen wir zunächst auf die betriebswirtschaftlichen Aspekte einer solchen Plattform ein und stellen danach einige Anwendungsszenarien vor. Schließlich diskutieren wir die technischen Herausforderungen bei der Umsetzung einer solchen Plattform und die entsprechenden Lösungsansätze. Abgeschlossen wird der Beitrag durch die Beschreibung der Business-Web-Architektur.

## 25.2 Umsatzpotenziale

Im Gegensatz zu App-Stores ist das Business Web eine Plattform, die über das reine Anbieten und Verkaufen von Software-Anwendungen hinausgeht. Beim Business Web geht es darum, geschäftliche Dienstleistungen Ende-zu-Ende abzudecken. Das kann neben der benötigten Software – Software-Services, Business-Apps – und Dienste zu deren Orchestrierung weitere Dienstleistungen, z. B. rechtlicher Art, umfassen. Das Business Web wird die erforderliche sichere Infrastruktur, Technologien, Anwendungen und Inhalte für eine echtzeitfähige Geschäfts- und Handelsplattform zur Verfügung stellen. Dabei liegt ein besonderes Augenmerk auf der Vereinfachung der Kooperation der Geschäftspartner, dem mobilen Zugang zur Plattform und zu den darauf angebotenen Diensten sowie auf deren einfacher Kombinierbarkeit.

Das Business Web nimmt den Trend auf, dass immer mehr Geschäftsprozesse, auch über e-Commerce hinaus, im Internet abgewickelt werden. Das Business Web stellt die Sicherheit und Zuverlässigkeit bereit, die für unternehmenskritische Anwendungen notwendig sind, aber im heutigen Internet nur mit Einschränkungen vorhanden sind. Dadurch zeigt das Business Web einen möglichen Weg zur Realisierung eines „Future Internet“ auf, das als Rückgrat für Wirtschaft und Gesellschaft dienen kann.

Somit bringt das Business Web ganz unterschiedliche Marktteilnehmer sowohl auf der Produzenten- als auch auf der Konsumentenseite zusammen:

- Unternehmen können geschäftliche Dienstleistungen anbieten und die verschiedenen Anbieter über eine sichere Cloud-Infrastruktur in Anspruch nehmen.
- Anbieter von Inhalten können Daten verkaufen, die sie bspw. über das Internet der Dinge oder Enterprise-Systeme sammeln.
- Software-Entwicklungsfirmen können sich zu Communities zusammenschließen und eine große Vielfalt von Anwendungen entwickeln, die auf spezielle Kundenanforderungen zugeschnitten sind. Dies wird insbesondere durch die Kombinierbarkeit und Wiederverwendbarkeit einzelner Software-Dienste wirtschaftlich sinnvoll.

- Geschäftliche Nutzer und private Endverbraucher können einfach und problemlos geschäftliche Dienste auf ihren mobilen Endgeräten nutzen.

Damit erschließt das Business Web die folgenden Umsatzpotenziale:

- „*One-stop Shop*“ für mobile geschäftliche Dienstleistungen: Das Business Web bietet einen Marktplatz für das Finden und die Nutzung geschäftlicher Dienstleistungen. Dies schließt ggf. die Inbetriebnahme von Software ein. Umsatzpotenziale ergeben sich in diesem Kontext bspw. aus nutzer- oder transaktionsbasierten Gebühren für die Dienstleistungen. Ferner sind zeitlich begrenzte Abonnements denkbar. Für den Marktplatzbetreiber können Werbeeinnahmen weitere Umsätze generieren.
- *Vertrauenswürdiges Netzwerk von Dienstleistern*: Unternehmen werden ihre Geschäftsaktivitäten nur in eine Cloud-Infrastruktur verlagern, wenn diese sicher, schnell und zuverlässig ist. Das Business Web wird Mechanismen zur Zertifizierung und Kontrolle aller Dienstleister und Geschäftspartner bieten. Die Privatheit aller Daten wird über verschiedene Kernmechanismen sichergestellt (s. Abschnitt 4.2). Mögliche Umsätze ergeben sich hier durch Zertifizierungs- und Qualifikationsprogramme.
- *Business Intelligence auf globalen Echtzeitdaten*: Im Business Web werden eine große Anzahl heterogener Daten- und Informationsquellen zusammengeführt. Dazu zählen unter anderem Enterprise-Systeme, die über zuverlässige und sichere Verbindungen in das Business Web integriert werden, sowie intelligente Objekte aus dem Internet der Dinge. Die Aufbereitung, Analyse und Aggregation dieser Informationen mit Hilfe moderner, echtzeitfähiger Business-Intelligence-Werkzeuge bietet für viele Unternehmen einen Mehrwert, der für entsprechende Dienstleister neue Geschäftsfelder eröffnet. Beispiele sind der direkte Verkauf dieser Daten oder das Angebot von Dienstleistungen basierend auf diesen Daten.

## 25.3 Anwendungsfälle

Neben dem Entwurf und dem Aufbau der Business-Web-Plattform werden im Rahmen der Business-Web-Initiative Anwendungsszenarien evaluiert und prototypisch realisiert. Dies erfolgt gemeinsam mit Unternehmen, die als Early Adopter frühzeitig die im Business Web verwendeten Technologien kennenlernen möchten.

Im Vordergrund stehen dabei zunächst Anwendungsfälle, die grundsätzlich durch existierende Software-Anwendungen abgebildet werden können oder zum Teil, bspw. im Metro Future Store, bereits abgebildet wurden. Sie heben jedoch einzelne Aspekte des Business Web hervor und erlauben dadurch einen schrittweisen Aufbau und Test der Technologie im Verlauf des Forschungsprojekts. Der Mehrwert des Business Web im Gegensatz zu isolierten Implementierungen der einzelnen Anwendungsfälle liegt in der Wiederverwendbarkeit und flexiblen Kombination der in den Anwendungsfällen verwendeten Teilfunktionalitäten.

Die unten beschriebenen Szenarien der ersten Projektphase verlangen gleichermaßen den Zugang zu Geschäftsprozessen über mobile Endgeräte, den Zugriff auf Daten aus Backend-Systemen und eine Echtzeitverarbeitung. Bei der Flexibilisierung und vereinfachten Einführung, der Skalierbarkeit und Sicherheit sowie bei den angesprochenen Nutzungsgruppen werden unterschiedliche Schwerpunkte gesetzt. Die organisationsübergreifende Zusammenarbeit über das Business Web wird in der zweiten Projektphase durch weitere Anwendungsszenarien abgedeckt.

### ***25.3.1 Shopping Assistant und Präzisionsmarketing***

Gemeinsam mit Groupe Casino, einem weltweit führenden Einzelhandelskonzern aus Frankreich, entwickelt die SAP-Forschung in der Business-Web-Initiative eine Marketing-Plattform, die sowohl Kunden als auch Einzelhandelsunternehmen unterstützt.

Hier können Konsumenten ihr mobiles Endgerät als persönlichen Shopping-Assistenten nutzen, der sie auf dem Weg zum und im Supermarkt über aktuelle Angebote informiert, Kaufempfehlungen gibt und zusätzliche Produktinformationen bereitstellt, bspw. zu Preisen, Inhaltsstoffen oder Bewertungen aus Online-Communities. Das Business Web dient hierbei als Plattform, auf der die mobile Anwendung bereitgestellt wird, aber auch die dahinterliegenden Prozesse orchestriert und Prozessschritte ausgeführt werden. Hierzu gehören die Identifikation des Konsumenten, die Erfassung seines Standorts sowie die gemeinsame Auswertung seines Kaufprofils und Treuestatus mit Informationen zu Produkten und Angeboten des betreffenden Supermarkts. Die Daten werden aus verschiedenen Quellen so konsolidiert und in Echtzeit ausgewertet, dass dem Konsumenten relevante Produktinformationen und -empfehlungen personalisiert zur Verfügung gestellt werden.

Der Einzelhändler kann über die Plattform wiederum Daten zu den Kaufentscheidungen seiner Kunden in Echtzeit auswerten. Käufergruppen, Produkte, deren Preise und Umsätze sowie Einkaufsort und -zeit lassen sich in Beziehung setzen, wodurch die Kaufgewohnheiten und -wünsche für eng umrissene Käufergruppen sichtbar werden. Diese Informationen lassen sich bspw. für temporäre, regionale Rabatte oder ein Cross-Selling zielgerichtet nutzen.

### ***25.3.2 Vertriebsunterstützung in der Konsumgüterindustrie***

Im zweiten Anwendungsfall wird das Business Web als Plattform für mobile Anwendungen speziell für VertriebsmitarbeiterInnen aus der Konsumgüterindustrie eingesetzt. Das Szenario, das gemeinsam mit dem niederländisch-britischen Konsumgüterkonzern Unilever entwickelt wird, umfasst zwei Anwendungen. Zum einen können VertriebsmitarbeiterInnen ihren Tablet-Computer nutzen, um unterwegs alle Daten, die für Gespräche mit Einzelhändlern relevant sind, anzuzeigen. Hierzu

gehören Produktinformationen, wie sie sich auch in traditionellen, gedruckten Produktkatalogen finden, aber auch interaktive Statistiken zu Verkaufszahlen einzelner Produkte, regionale Vergleiche und Zeitreihen zur Historie von Kaufentscheidungen. Zudem lassen sich gezielt regionale Werbekampagnen anzeigen, einschließlich Streamings von TV-Werbung. So wird das mobile Endgerät zu einem handlichen und flexiblen Werkzeug, das Gespräche mit Einkäufern und Managern von Supermärkten unterstützt.

Zum anderen stellt die Anwendung Funktionen bereit, mit denen VertriebsmitarbeiterInnen zentrale Arbeitsschritte direkt vor Ort durchführen können. Hierzu gehört die Erstellung von To-Do-Listen, die Bestellung von Produkten, aber auch die Kontrolle der Stellflächen im Supermarkt. So ist vertraglich meist festgelegt, in welcher Regalbreite ein Supermarkt die Produkte eines Konsumgüterunternehmens aufstellt. Mit dem Tablet-Computer lässt sich dank Zugriff auf Backend-Systeme und integrierter Kamera die Regalbreite kontrollieren, Abweichungen von der vertraglich vereinbarten Breite dokumentieren und vor Ort direkt ansprechen.

Bereits auf den ersten Blick weist die Anwendung auf zentrale Eigenschaften des Business Web hin, bspw. auf die flexible Einbindung von Daten aus verschiedenen Quellen und darauf aufbauend auf die Mobilisierung von Geschäftsanwendungen. Mindestens genauso wichtig ist aber, dass hier das Business Web Werkzeuge für die kosteneffiziente Entwicklung und Anpassung solcher Anwendungen bereitstellt und als Cloud-Infrastruktur deren Skalierbarkeit sicherstellt, was eine wichtige Voraussetzung für die Lokalisierung und den weltweiten Roll-out solcher Anwendungen ist.

### 25.3.3 Monitoring von Kühltruhen

Im Zentrum des dritten Anwendungsfalls steht das Internet der Dinge. Gemeinsam mit Unilever erstellt die SAP-Forschung eine Business-Web-Anwendung, die die Versorgung von Händlern mit Eiscreme und die laufende Kontrolle der Kühltruhen unterstützt. Die Kühltruhen werden durch Sensoren zu intelligenten Objekten, die ihren Ort und die Eistemperatur kontinuierlich über das Business Web dokumentieren. Dadurch lassen sich die Kühltruhen leichter auffinden und leichter mit den richtigen Eissorten versorgen. Nicht selten werden Kühltruhen an andere Händler weitergegeben oder entgegen vorheriger Absprachen mit Eissorten anderer Hersteller bestückt. Zudem lässt sich über den Temperaturverlauf aufzeichnen und damit ermitteln, welches Eis mit hoher Wahrscheinlichkeit zwischenzeitlich geschmolzen war und damit von niedriger Qualität bzw. ungenießbar ist und ausgetauscht werden muss.

Dieser für das Internet der Dinge „klassische“ Anwendungsfall ermöglicht die Evaluierung des Business Web als Plattform für Anwendungen, die auf Informationen aus dem Internet der Dinge aufsetzen. Das Business Web dient hier als Middleware, die Sensordaten aufnimmt, deren Auswertung in Echtzeit ermöglicht und darauf aufbauend die Optimierung von Geschäftsprozessen unterstützt.

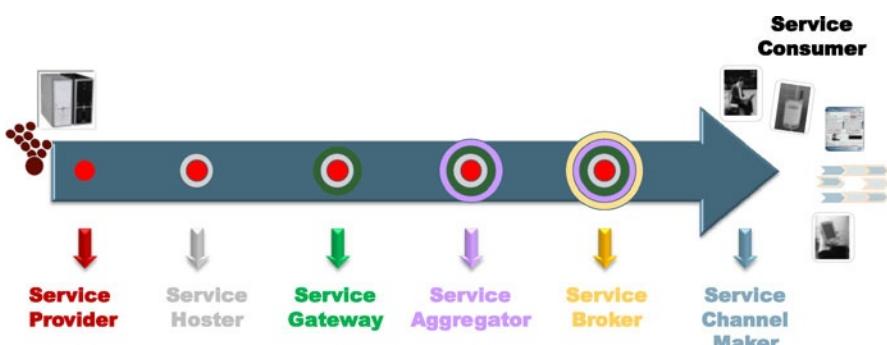
## 25.4 Technologische Herausforderungen

Die sehr umfassende Vision des Business Web bringt eine Reihe technologischer Herausforderungen mit sich. Nachfolgend gehen wir nur auf einige, sehr wesentliche Herausforderungen exemplarisch ein, da eine umfassende Abhandlung aller Herausforderungen den Rahmen dieses Beitrags sprengen würde.

### 25.4.1 Flexibilisierung

Wie bereits eingangs erwähnt, ziehen sich viele Unternehmen auf ihre Kernkompetenzen zurück. Unternehmen reduzieren ihre Fertigungstiefe und kooperieren vorzugsweise in dynamischen Geschäftsnetzwerken und Partner-Eco-Systemen. Daher benötigen sie IT-Systeme, die flexibel genug sind, diese Dynamik zu unterstützen. Von besonderer Bedeutung ist dabei die Wiederverwendung und Kombinierbarkeit von existierenden Komponenten in neuen oder neu angepassten Geschäftsprozessen. Der Paradigmenwechsel von Client-Server-Architekturen zu service-orientierten Architekturen ist dabei aus IT-Sicht ein entscheidender Schritt. Neben den von SOA ursprünglich eingeführten Rollen Dienstnutzer (service consumer), -vermittler (service broker), und -anbieter (service provider) kristallisieren sich gerade zusätzliche Rollen heraus (vgl. Abb. 25.3) [1]: Zum einen kümmert sich die Rolle des Service Hoster um den Betrieb von Diensten in Cloud-Umgebungen im Auftrag des Service Providers. Service Gateways behandeln die Interoperabilität und Mediation von Informationen zwischen Diensten. Der Service Aggregator bündelt Dienste zu Mehrwertdiensten. Schließlich kümmert sich der Service Channel Maker um die finanzielle Verwertung und Darstellung von Diensten in Form von Mashups oder Widgets u. a. auf mobilen Endgeräten.

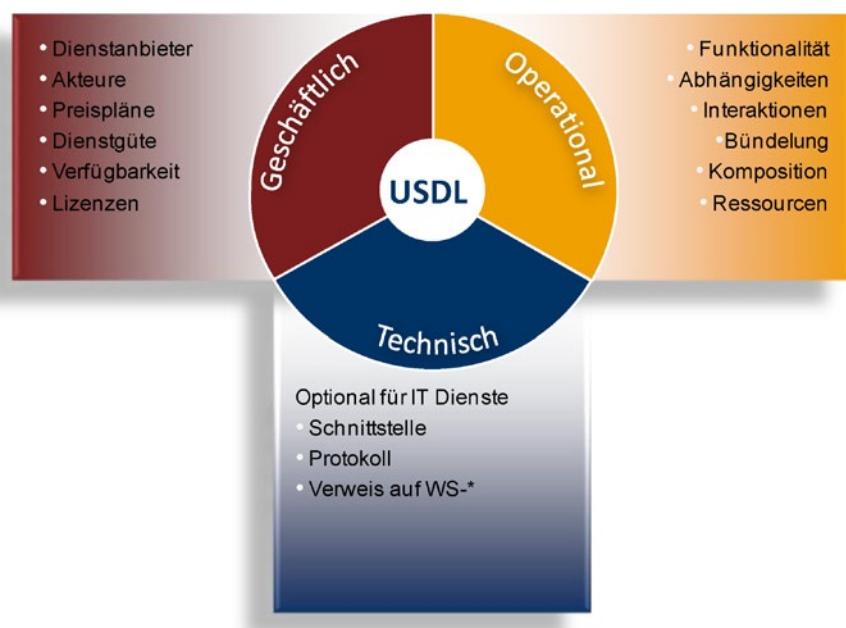
Durch die Vielzahl der Rollen steigt die Signifikanz eines gemeinsamen Standards zur Beschreibung von operationalen und betriebswirtschaftlichen Eigenschaf-



**Abb. 25.3** Die von SOA eingeführten Rollen Service Consumer, Provider und Broker werden erweitert um Service Hoster, Service Gateway, Service Aggregator und Service Channel Maker

ten eines Dienstes. Eine standardisierte Beschreibung hilft bei der Flexibilisierung, was gerade am Beispiel des Dienstaggregators, welcher Dienstbündel aus eigenständigen Diensten baut, klar zutage tritt. Ein Dienstbündel ist dabei eine zentrale Eigenschaft, um aufbauend auf existierenden Diensten (ggf. spontan) neue, innovative Dienstleistungen zu realisieren. Die existierenden Dienste können dabei von unterschiedlichen Anbietern bereitgestellt und aggregiert werden. Ein Beispiel für ein Dienstbündel wäre eine Pauschalreise mit Flughafentransfer, Flug, Hotelbuchung und anderen Diensten. Ein weiteres Beispiel wäre ein Dienstbündel, welches den Export von Gütern unterstützt durch Aggregation von Logistikdiensten, Versicherungen, Zollabfertigung, Zertifizierung, Bürgschaften u. v. m. Die Ökonomie von Dienstbündeln definiert sich als Kosten der Aggregation in Relation zum wahrgenommenen Vorteil. Naheliegenderweise sind die Kosten dominiert durch die Aggregationskosten. Eben diese Aggregationskosten können durch eine einheitliche Beschreibung minimiert werden. Idealerweise kann ein Dienstbündel von standardisiert beschriebenen Diensten auf Basis einer standardisierten Werkzeugkette durchgeführt werden, so dass die Aggregationskosten minimal werden.

Im Kontext des SOA-Paradigmas gibt es bereits eine Reihe von Standards, üblicherweise bezeichnet als WS-\*. Spezifikationen. WS-\* ist eine Sammelbezeichnung für mehrere Dutzend Beschreibungssprachen, darunter bspw. WSDL (Web Service Description Language) oder WS-BPEL (Business Process Execution Language), die auf rein technische Aspekte von IKT-Diensten fokussieren.



**Abb. 25.4** USDL vereinheitlicht die Beschreibung von betriebswirtschaftlichen, operationalen und technischen Aspekten von Diensten

Im Rahmen des Business Web wird WS-\* deshalb erweitert um betriebswirtschaftliche Aspekte, wie z. B. Preispläne, Verfügbarkeit, oder Angaben zum Anbieter. Eine entsprechende Beschreibungssprache befindet sich derzeit unter dem Namen *Unified Service Description Language (USDL)*<sup>1</sup> im Zuge der Standardisierung beim World Wide Web Consortium (W3C). Eine solche umfassende Dienstbeschreibung mit betriebswirtschaftlichen und operationalen Aspekten ist die Voraussetzung für die Bündelung und das Auffinden von Diensten. Eine Übersicht ist in Abb. 25.4 dargestellt.

### 25.4.2 Sicherheit

Die Fokussierung auf Kernkompetenzen und die daraus folgende starke Einbindung in hochgradig verteilte Geschäftsnetzwerke haben zwangsläufig ein hohes Maß an Informationsaustausch zur Folge. Wie die Anwendungsfälle in Abschn. 25.3 zeigen, beinhaltet das in der Regel auch unternehmenskritische und personenbezogene Daten wie Kundenprofile, Lagerbestände oder Verkaufszahlen. In diesem Zusammenhang wird zu Recht die Frage nach der Sicherheit und Vertrauenswürdigkeit des Business Web gestellt, umso mehr, je dynamischer und komplexer die geschäftlichen und technischen Strukturen sind. Es ist im Interesse aller Teilnehmer des Business Web, wenn der Besitzer sensitiver Informationen seine Sicherheitspräferenzen formulieren und die Kontrolle über deren Verarbeitung behalten kann, auch und insbesondere, wenn diese Informationen seinen eigenen Hoheitsbereich verlassen. Mit Hilfe eines Nachweises, dass Dienste diese Präferenzen berücksichtigen, kann blindes Vertrauen durch kalkulierbares Risiko ersetzt werden. Der Nachweis kann durch Prüfung und Zertifizierung, aber auch durch technologische Erzwingung in der Erstellung und Ausführung von Diensten erbracht werden.

Die Business-Web-Technologie unterstützt diese Anforderungen durch Sicherheitsdienste auf Plattformebene, die bedarfsgerecht genutzt werden können, um vertrauenswürdige Dienste und Anwendungen bereitzustellen. Ein solcher Sicherheitsdienst umfasst beispielsweise das Identity Management (IdM). Die Verwaltung von Benutzern, deren Authentifikation, die Auswahl der benötigten Identitätsattribute sowie deren Kapselung in einem sicheren Token wird von einem oder mehreren Identity Providern übernommen, die diese Sicherheitsfunktionen im Auftrag durchführen und ihre Ergebnisse an die aufrufenden (Anwendungs)-Dienste weitergeben. Auf diese Weise sind Anwendungsdienste von der notorisch fehleranfälligen Realisierung der IdM-Funktionen befreit. Der Ansatz ermöglicht zudem die Unterstützung föderierter Identitäten, d. h. die Zusammenführung und vertrauenswürdige Handhabung verschiedener digitaler Identitäten im Identity Provider; eine Grundvoraussetzung für die Unterstützung domänenübergreifender dynamischer Sicherheitskonzepte. Das im Business Web bereitgestellte IdM-as-a-Service (IdMaaS)

<sup>1</sup> Ausführliche Spezifikationen der Sprache finden sich unter [www.internet-of-services.com](http://www.internet-of-services.com).

unterstützt mit SAML2.0<sup>2</sup> den wichtigsten Industriestandard für die beschriebenen Konzepte.

Wie bereits erwähnt ist der weitgehende gegenseitige Austausch sensibler Daten ein charakteristisches Merkmal von Business-Web-Anwendungen. Der Shopping Assistant (siehe Abschn. 25.3.1) verwendet etwa personenbezogene Nutzerdaten (Identitäten und Kundenprofile), Lokationsinformationen, Geschäftsdaten des Supermarktes (auch einzelner Filialen) und allgemeine Marktdaten in verschiedenen Umgebungen: der lokalen Applikation auf dem mobilen Endgerät, der Backend-Anwendung des Einzelhandelsunternehmens sowie externer Dienste, etwa spezifischer Marktforschungsangebote. Der Kunde hat ein Interesse daran, dass neben der Berücksichtigung datenschutzrechtlicher Bestimmungen nur solche Daten weitergegeben werden, für die er seine explizite Zustimmung gegeben hat. So möchte er vielleicht Daten über Lebensmittel, nicht aber über Hygieneprodukte in sein Profil aufgenommen sehen, möchte ausschließen, dass sein Profil an bestimmte Dritte weitergegeben wird, oder die Löschung seines Profils nach Ablauf einer bestimmten Zeit erzwingen. Wurde dies bislang meist vertraglich gelöst, bietet das Business Web technologische Unterstützung. Während traditionelle Berechtigungsmechanismen die Definition und Durchsetzung von Sicherheitspolitiken meist nur im Kontext einer Applikation oder eines Systems unterstützen, können im Business Web mit sogenannten „Sticky Policies“ Sicherheitspräferenzen untrennbar an die Daten gekoppelt werden und bleiben so unabhängig vom Verwendungskontext erhalten. Da die Policies über die gesamte Lebenszeit der Daten verfügbar sind, erlauben sie außerdem die Formulierung von Obligationen, d. h. Verwendungseinschränkungen und Verarbeitungsvorschriften, die erst zu einem Zeitpunkt nach der Weitergabe der



**Abb. 25.5** Prinzipdarstellung der Definition individueller Sticky Policies im Shopping-Assistant-Szenario

<sup>2</sup> <http://saml.xml.org/saml-specifications>

Daten wirksam werden. Abbildung 25.5 zeigt, wie der Nutzer Sticky Policies auf einem Mobilgerät definieren könnte. Beispiele für Obligationen sind das Löschen von Daten nach einer bestimmten Zeit, Benachrichtigung bei Ereignissen oder die Anonymisierung von Daten bei der Weitergabe.

Natürlich decken die obigen Ausführungen nur einen Teil der Sicherheitsanforderungen im Kontext des Business Web ab. Weitere Sicherheitsdienste können flexibel eingebunden werden und unterstützen skalierbare Sicherheit in Abhängigkeit der Risikoeinschätzung. Somit können gerade kleine und mittlere Dienstanbieter und -konsumenten ein Sicherheitsniveau erreichen, für das sie außerhalb des Business Web hohe Investitionen tätigen müssten.

### **25.4.3 Internet der Dinge**

Die Integration von Informationen aus dem Internet der Dinge in Geschäftsprozesse ermöglicht neue Anwendungsfelder, die dadurch gekennzeichnet sind, dass sich der Ablauf eines Prozesses, häufig in Echtzeit, an Gegebenheiten in der realen Welt anpassen kann. Meist werden diese Anwendungsfelder durch die Vorstellung des Begriffs „Smart“ gekennzeichnet. Im Kontext des Business Web steht gegenwärtig von diesen Anwendungsfeldern der Bereich Smart Logistics im Vordergrund. Sehr bald werden jedoch weitere Felder wie Smart Fab, Smart Product, als Spezialfall hiervon das Smart Car, sowie Smart Cities hinzukommen.

Ganz entscheidend für den Erfolg solcher Anwendungen wird eine ausreichende Standardisierung der Schnittstellen und Kommunikationsprotokolle im Internet der Dinge sein. Dabei geht es nicht nur um den Kommunikationskanal von den Dingen/Objekten zu IT-Systemen, wie wir ihn heute beispielsweise von RFID-Anwendungen im Supply Chain Management kennen. Wichtig ist auch der Rückkanal von IT-Systemen in das Internet der Dinge und eine effiziente Kommunikation der Objekte untereinander. Die Peer-to-Peer-Kommunikation unter Objekten spielt insbesondere beim Aufbau von Cyber-physical Systems eine wichtige Rolle. Diese Systeme sind dadurch gekennzeichnet, dass die einzelnen Objekte, aus denen das System aufgebaut ist, autonom und kooperativ auf Veränderungen und Ereignisse in ihrer Umgebung reagieren können. Im Projekt CoBIs [2] wurde ein solches System zur Überwachung von Lagervorschriften bereits erfolgreich aufgebaut.

Eine weitere wichtige Herausforderung ist die effiziente Verwaltung der Objekte bzw. der zugehörigen Geräte und Software. Heutige Lösungen zur Geräteverwaltung, wie Sybase Afaria [4], konzentrieren sich in der Regel auf Smartphones. In der Business-Web-Initiative wird geklärt werden, wie die dort verwendeten Konzepte zur Verwaltung der Software auf den Geräten und zur Sicherheit auf andere Geräteklassen wie RFID-Leser, Sensoren und eingebettete Systeme übertragen werden können.

Schließlich werden Fragen zur Preisfindung und Bezahlung adressiert. Vorstellbar sind bspw. transaktions-, volumen-, oder nutzungsbasierte Lizenzmodelle. Wei-

tere Modelle könnten das Objekt aus dem Internet der Dinge als BenutzerIn ansehen. Bei der Bezahlung müssen komplexe Szenarien wie beispielsweise ein elektrisch angetriebenes Mietfahrzeug unterstützt werden, das ggf. von verschiedenen BenutzerInnen an Stromladesäulen unterschiedlicher Energieversorger aufgeladen werden soll. Neben rein technischen Lösungen, wie beispielsweise SAP Convergent Charging [3], sind hier organisatorische Lösungen, wie Clearing Houses, im Fokus.

#### **25.4.4 Mobilität**

Ein besonderes Augenmerk beim Business Web liegt auf der Möglichkeit, die angebotenen Dienste von mobilen Endgeräten aus nutzen zu können. Entsprechend werden zukünftige Dienstanbieter im Business Web in der Entwicklung mobiler Anwendungen unterstützt. Eine große Herausforderung bei der Entwicklung und dem Vertrieb von mobilen Applikationen ist die Heterogenität der mobilen Endgeräte. Sie unterscheiden sich in den verwendeten Betriebssystemen und in ihren Hardware-Eigenschaften. Zusätzlich erschwert wird die kosteneffiziente Entwicklung von Anwendungen für diese Geräte durch die kurzen Produktlebenszyklen der Geräte selbst sowie der darauf verwendeten Laufzeitumgebungen.

Das Business Web bietet daher entsprechende Werkzeuge, die die plattformunabhängige Entwicklung und Wartung von mobilen Applikationen unterstützen. Hierzu gehören auch Werkzeuge zur Unterstützung des Zugriffs auf cloud-basierte Dienste und Datenbanken sowie Dienste zur effizienten Verwaltung der mobilen Endgeräte innerhalb der IT-Infrastruktur eines Unternehmens.

Bei der Unterstützung einer plattformunabhängigen Anwendungsentwicklung geht das Business Web über einen rein browser-basierten Ansatz hinaus, da ein solcher die Applikationen auf die vom Browser unterstützte Funktionalität einschränken würde. Stattdessen wird ein sogenannter hybrider Ansatz verfolgt, bei dem gerätespezifische „Player“ soweit wie möglich von der Zielplattform abstrahieren und dennoch die jeweilige Gerätefunktionalität weitestgehend nutzen können. Ausgeführt werden auf dieser als „Player“ bezeichneten Abstraktionsschicht schließlich geräteunabhängige Applikationen.

### **25.5 Architektur**

Das Business Web stellt sowohl die Technologien und Infrastruktur als auch die entsprechenden Applikationen und Inhalte für die Flexibilisierung von Geschäftsprozessen bereit. Ziel ist es, alle Phasen, bspw. Design, Entwicklung, Inbetriebnahme und Betrieb von Systemen und den vollständigen Lebenszyklus von Business Apps, Software-Diensten und Software-Komponenten abbilden zu können. In einer cloud-basierten Laufzeitumgebung können Dienste von den Betreibern, ihren Partnern und

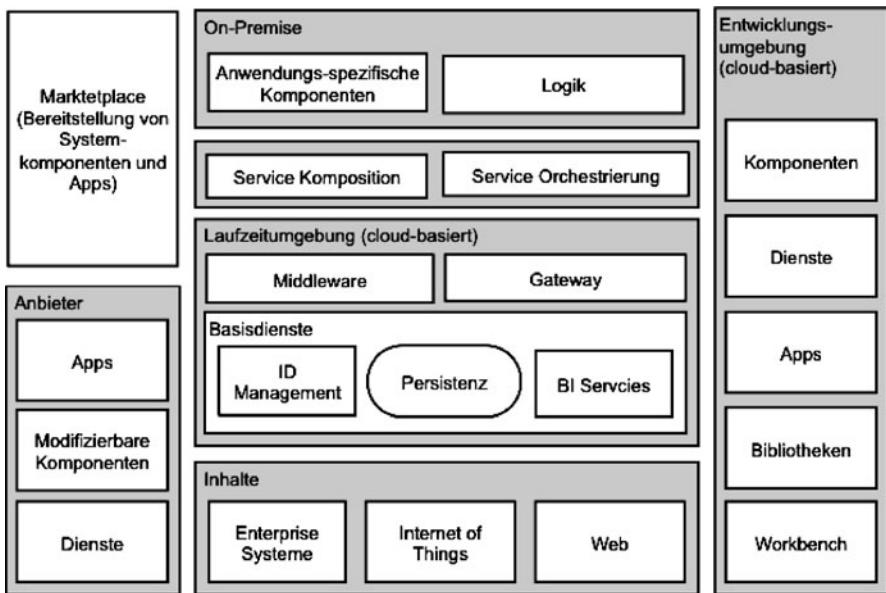


Abb. 25.6 Funktionale Komponenten des Business Web

unabhängigen Nutzern des Business Web einfach und effizient entwickelt, bereitgestellt und wiederverwendet werden.

Die angebotenen Daten, Informationen und Inhalte können aus ganz unterschiedlichen Quellen stammen, bspw. aus dem Internet der Dinge oder von integrierten Enterprise-Systemen. Durch die Nutzer können zudem Zusatzinformationen bereitgestellt werden, die von den verwendeten mobilen Endgeräten und deren Sensoren kommen, z. B. bei der Nutzung von Location-based Services.

Entsprechend ergeben sich die in Abb. 25.6 dargestellten funktionalen Einheiten des Business Web. Hierzu gehören Schnittstellen zur Integration von Enterprise-Systemen, zum Zugriff auf das Internet der Dinge oder zum Zugriff auf web-basierte Inhalte. Middleware und Gateways stellen sicher, dass auch Inhalte für komplexe, mobile Applikationen zwischen mobilen Endgeräten und Backend-Systemen repliziert und synchronisiert werden können. Anwendungsspezifische Komponenten oder Logik werden durch On-Premise-Bestandteile abgebildet. Die Laufzeit- und die Entwicklungsumgebung hingegen basieren weitestgehend auf dem On-Demand-Ansatz. Basisdienste stellen sicher, dass bspw. ein Identity Management (s. Abschn. 25.4.3), die Persistenz von Inhalten sowie Business-Intelligence-Dienste mit Unterstützung durch In-Memory-Technologie umgesetzt werden können. Durch letztere ist die in Abschnitt 2 erwähnte echtzeitfähige Analyse von Geschäftsdaten möglich.

Bereitgestellte Bibliotheken, Business Apps und eine Entwicklungsplattform erleichtern die Entwicklung und Wartung neuer mobiler Anwendungen. Hier kommen im ersten Schritt bspw. Eclipse-basierte Entwicklungskomponenten zum Ein-

satz. Die Verteilung von Inhalten über standardisierte Formate (z. B. HTML5) ermöglicht es, u. a. über hybride player-basierte Anwendungen, schnell und flexibel Geschäftsprozesse mobil abzubilden.

Die Marketplace-Komponente bildet die kommerzielle Plattform für das Anbieten, Finden und Vertrieben von Diensten, Anwendungen, Business Apps und Komponenten. On-Premise-, On-Demand-Lösungen und die Kombination beider Ansätze können entsprechend über mehrere Unternehmen hinweg angeboten werden. Eine zusätzliche Komponente ist für die Empfehlung von Lösungen verantwortlich. Zudem können auch Nachfragen nach bislang nicht existierenden Diensten oder Anwendungen von Kunden platziert werden.

Durch das Business Web lassen sich somit bestehende und neue Geschäftsprozesse flexibel und einfach abbilden und integrieren. Der vollständige wirtschaftliche Prozess für die Bereitstellung der Systemkomponenten, Dienste und Apps lässt sich über alle Partner hinweg im Business Web umsetzen.

## 25.6 Acknowledgements

Abschließend möchten wir erwähnen, dass die hier vorgestellten Konzepte und Lösungen keinesfalls ausschließlich durch die Autoren erdacht wurden. Sie sind vielmehr das Ergebnis einer großen Initiative, die die SAP-Forschung gemeinsam mit ihren Partnern durchführt. Daher gilt unser Dank selbstverständlich allen, die an dieser Initiative beteiligt sind und die geholfen haben, die Vision des Business Web zu schärfen.

## Literaturverzeichnis

1. Barros A und Kyau U (2011) Service Delivery Framework – An Architectural Strategy for Next-Generation Service Delivery in Business Network. In: P. Kellenberger (ed) Proceedings 2011 Annual SRII Global Conference SRII 2011, 30 March–2 April 2011, San Jose, California, USA. IEEE Computer Society Conference Publishing Services (CPS), pp 47–58
2. Kubach U, Decker C, Douglas K (2004) Collaborative control and coordination of hazardous chemicals, Proceedings of the 2nd international conference on Embedded networked sensor systems 2004 (SenSys 2004), Baltimore, MD, USA
3. Sybase Afaria. <http://www.sybase.com/products/mobileenterprise/afaria>, 27. Oktober 2011
4. SAP Convergent Charging.  
<http://www.sap.com/industries/telecom/businessprocesses/convergent-charging/index.epx>, 27. Oktober 2011

**Teil VIII**

**Technologische Sicht: Plattformen &**

**Betriebssysteme**

---

# Kapitel 26

## Architekturen mobiler Multiplattform-Apps

Felix Willnecker, Damir Ismailović und Wolfgang Maison

**Zusammenfassung** Die dramatisch schnelle Entwicklung von Smartphones und Tablets hatte zur Folge, dass extrem heterogene Systemlandschaften entstanden sind. Für die Entwicklung von Applikationen der Post-PC-Ära sind daher eine Reihe von Ansätzen zur Multiplattform-Entwicklung entstanden. Dieser Beitrag beleuchtet die Vor- und Nachteile verschiedener Systemarchitekturen für die Entwicklung von Multiplattform-Smart-Mobile-Apps: Angefangen bei Web Apps, über Hybrid Apps und Cross-Compiler bis hin zu Applikationsbeschreibungssprachen.

### 26.1 Plattformwildwuchs

Die Anzahl mobiler Applikationen und Geräte steigt rasant an. Für 2015 rechnet die International Data Corporation (IDC) bereits mit 1 Milliarde Smartphones und bereits jetzt sind über 500.000 Apps in Apples AppStore, dem derzeit größten Vertrieber mobiler Applikationen, verfügbar [1, 2]. Das Spektrum der Applikationen reicht von einfachen Alltagshelfern bis hin zu komplexen prozessunterstützenden Applikationen im Geschäftsbereich. Vor allem die iOS-Plattform von Apple ist Vor-

---

Felix Willnecker

Weptun GmbH München, Technische Universität München, Institut für Informatik, Lehrstuhl für Angewandte Softwaretechnik,

E-mail: f.willnecker@weptun.de

Damir Ismailović

Technische Universität München, Institut für Informatik, Lehrstuhl für Angewandte Softwaretechnik,

E-mail: damir.ismailovic@in.tum.de

Wolfgang Maison

Weptun GmbH München,

E-mail: w.maison@weptun.de

reiter im Bereich von innovativen Applikationsideen, doch andere Plattformen, allen voran Googles Android-Plattform, holen stark auf [3].

Die mobilen Plattformen zusammen mit den Vertriebs- und Distributionskanälen haben sich dabei jeweils zu eigenen Ökosystemen entwickelt. Für jedes dieser Ökosysteme wird vom Hersteller ein eigenes Entwicklungssystem vorgegeben. Dies führt dazu, dass die Entwicklungskosten mit der Anzahl angebotener Plattformen multipliziert werden muss. Diese Rechnung gilt außerdem für den kompletten Lebenszyklus einer Applikation; insbesondere für Wartung und Pflege. Um die Kosten zu senken gibt es ein starkes Bestreben, das „Develop once, run everywhere“-Prinzip auf mobile Umgebungen zu übertragen. Eine Reihe von Konzepten ermöglicht bereits jetzt die Entwicklung von Applikationen für mehrere Mobilplattformen.

Web Apps sind die bekanntesten Vertreter von plattformübergreifenden mobilen Anwendungen. Hybrid Apps für mobile Plattformen etablieren sich ebenfalls am Markt. Dazu stoßen Cross-Compiler und der relativ neue Bereich der Applikationsbeschreibungssprachen, die versuchen, Nachteile von Web Apps, Hybrid Apps und Cross-Compilern auszubalancieren.

Apples CEO, Steve Jobs, prägte 2010 den Begriff der Post-PC-Ära<sup>1</sup>. Darunter ist der Paradigmenwechsel vom Desktop-Internet zu Smartphone-Applikationen und Tabletsystemen zu verstehen. Dieses neue „Zeitalter“ bringt eine völlig andere Art von Benutzerführung und Kontext mit sich und gleichzeitig neue Herausforderungen für die Entwicklung von Applikationen. In diesem Artikel geben wir einen kurzen Überblick über die bestehenden Techniken und aktuellen Entwicklungen auf diesem Gebiet.

## 26.2 Überblick über die gängigsten mobilen Multiplattformen

Probleme und Herausforderungen der Post-PC-Ära können mit unterschiedlichen Ansätzen vereinfacht bzw. gelöst werden. Für die Beurteilung der Qualität einer Multiplattform-Lösung muß die Flexibilität von mit ihr erstellten Anwendungen den technischen Möglichkeiten gegenübergestellt werden. Bei den technischen Möglichkeiten ist eine ausgeprägte Nutzung der Gerätesensorik und des jeweiligen User Interface Design wichtig. Abbildung 26.1 stellt die verschiedenen Lösungen/Konzepte gegenüber und ordnet sie nach Flexibilität und Funktionsumfang an. Die Grafik enthält ebenfalls Applikationen, die jeweils speziell für eine Mobilplattform entwickelt wurden (Native Apps).

Den größtmöglichen Funktionsumfang erreicht man derzeit mit Applikationen, die mit den jeweiligen Entwicklungsplattformen des Smartphonebetriebssystems entwickelt werden, sogenannten nativen Applikationen. Um native Applikationen entwickeln zu können, muss der Entwickler mit der jeweiligen Umgebung vertraut sein. In Tabelle 26.1 findet sich eine kurze Auflistung der gängigsten Betriebssysteme und deren typischen Sprachen bzw. Entwicklungsumgebungen.

<sup>1</sup> [http://news.cnet.com/8301-13860\\_3-20006526-56.html](http://news.cnet.com/8301-13860_3-20006526-56.html)

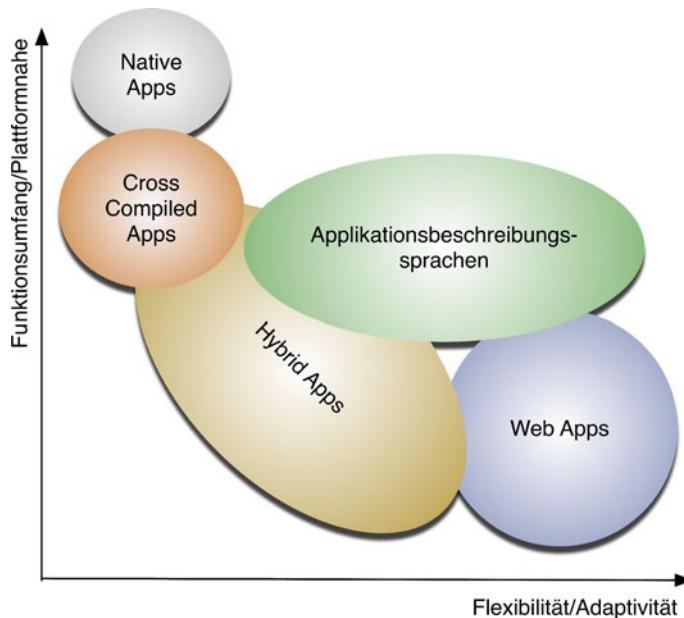


Abb. 26.1 Übersicht mobiler Multiplattformumgebungen

Tabelle 26.1 Native Plattformen und Tools

Betriebssystem	Programmiersprache	Entwicklungsumgebung
iOS	Objective-C	Xcode
Android	Java	Eclipse
Windows Phone 7	C#, Visual Basic	Visual Studio
BlackBerry	Java	Eclipse
Symbian	Java	Symbian IDE

Bereits hier zeigt sich, wie unterschiedlich die verschiedenen Smartphone-Betriebssysteme sind. Java wird bei weitem am häufigsten eingesetzt, allerdings nutzt jedes Smartphone-Betriebssystem eigene Frameworks, um die plattformspezifischen Charakteristiken der Geräte abzubilden. Für die Entwicklung von Apple iOS-Applikationen mithilfe der Entwicklungsumgebung Xcode ist ein Apple-Computer zwingende Voraussetzung. Ähnlich verhält es sich mit Windows Phone 7 Apps, hier wird Microsofts Windows-Betriebssystem vorausgesetzt. Der hohe Funktionsumfang muss hier also mit hohen Kosten bezahlt werden.

Web Apps hingegen sind zunächst plattformunabhängig und lösen auf den ersten Blick die Kosten- und Komplexitätsprobleme nativer Applikation. Man büßt dafür allerdings einiges an Funktionsumfang ein. Die Hersteller der Smartphone-Betriebssysteme stellen Toolkits bereit, die Zugriff auf die Gerätesensorik zulassen und somit den Funktionsumfang von Webapplikationen deutlich erweitern. Dafür kommt aber wiederum proprietäre Software zum Einsatz, wie z. B. Googles Web

**Tabelle 26.2** Web-App-Plattformen und Tools

Betriebssystem	Browserengine	Erweiterungen
iOS	Webkit	Safari Exentions
Android	Webkit	Android WebKit
Windows Phone 7	Trident	WebMatrix
BlackBerry	Mango/Webkit	WebWorks
Symbian	Webkit	Symbian Web App Tools
webOS	Webkit	webOS SDK

Toolkit (GWT). Die Basistechnologie ist hier allerdings immer die gleiche: Die Hypertext Markup Language (HTML), JavaScript (JS) und Cascading Style Sheets (CSS). Darüber hinaus steht für jedes Betriebssystem eine Vielzahl von unterschiedlichen Browsern zur Verfügung. Diese Browser unterscheiden sich bei der Interpretation des HTML-Standards (insbesondere im Falle von HTML5<sup>2</sup>) zum Teil sehr stark. So kann auch hier nur von sehr eingeschränkter Plattformunabhängigkeit gesprochen werden. Tabelle 26.2 zeigt sehr deutlich die Unterschiede der diversen Erweiterungen der Browserengines für den Zugriff auf die Gerätesensorik [4].

Den Ansatz der Web Apps verstärken die sog. Hybrid Apps, indem sie ein API bereitstellen, der einen direkten Zugriff auf Sensoren der Geräte ermöglicht. Entwicklungswerzeuge wie das Titanium Framework<sup>3</sup> oder PhoneGap<sup>4</sup> nutzen die aus dem Web bekannten Entwicklungs- und Beschreibungssprachen, um daraus quasi native Smartphoneapplikationen zu generieren. Dabei kommen eigene API-Implementierungen zum Einsatz. Nach eigenen Angaben unterstützt PhoneGap mit diesem Ansatz derzeit fünf verschiedene Plattformen (iOS, Android, WebOS, BlackBerry, Symbian). Titanium kann derzeit auf iOS und Android eingesetzt werden.

Einen anderen Ansatz verfolgen Cross-Compiler. Sie übersetzen den Quellcode direkt in eine Sprache, die nativ auf dem jeweiligen Gerät kompiliert werden kann. Für die Entwickler von .NET-Applikationen steht das MonoTouch Framework bereit. Mit diesem Framework ist es möglich, mit Microsofts .NET Entwicklungssprachen wie zum Beispiel C# oder VisualBasic Smartphone-Applikationen zu entwickeln und diese mithilfe eines Cross-Compilers in native iOS- oder Android- Applikationen umzuwandeln. Wie bei Titanium oder PhoneGap kommen hier proprietäre API Implementierungen zum Einsatz, um die Spezifikationen der Geräte möglichst optimal abzubilden [5].

Applikationsbeschreibungssprachen (Application Description Language, ADL) wie das Application Construction Kit (AppConKit<sup>5</sup>) verhalten sich ähnlich wie Hybrid Apps. Sie nutzen allerdings die Vorteile einer Webapplikationsarchitektur. Diese zeichnet sich vor allem durch ihren einheitlichen Ressourcen-basierten Archi-

<sup>2</sup> <http://dev.w3.org/html5/spec/Overview.html>

<sup>3</sup> <http://www.appcelerator.com/products/titanium-cross-platform-application-development/>

<sup>4</sup> <http://docs.phonegap.com/>

<sup>5</sup> <http://www.weptun.de/portfolio/appconkit>

tekurstil aus, auch Representational State Transfer (REST) genannt. Anstelle von HTML und JS wird hier allerdings auf für den mobilen Kontext ausdrucksstärkere DSLs, so genannte ADLs, zurückgegriffen. Es ist sowohl möglich, Applikationen zu generieren, als auch diese zur Laufzeit abzufragen, um ein hohes Maß an Adaptivität zu erreichen [6, 14].

## 26.3 Mobile Web Apps

Basis des Content-Publishing im Web ist die 1999 fertiggestellte Spezifikation der Hypertext Markup Language (HTML/4.01<sup>6</sup>). HTML war von Beginn an als plattformunabhängige offene Dokumentbeschreibungssprache konzipiert. Gemeinsam mit dem Übertragungsprotokoll Hypertext Transfer Protocol (HTTP/1.1<sup>7</sup>) bilden beide Technologien die Grundpfeiler des Internets und sämtlicher darauf aufbauender Plattformen. Der offene hardware- und softwareherstellerunabhängige Standardisierungs- und Spezifikationsprozess gilt hier als Erfolgsmodell [8]. Zu diesen beiden Basistechnologien gesellen sich JavaScript, eine dynamisch typisierte Skriptsprache für Manipulationen des dem HTML-Standard zu Grunde liegenden Document Object Models (DOM), und Cascading Style Sheets (CSS), eine Sprache zur Definition von Dokumentvorlagen auf Basis von HTML Tags [9, 10]. Diese beiden Erweiterungen erlauben die Implementierung von clientseitiger Logik sowie die Reduktion der Gestaltungslogik mithilfe von Vorlagen. Die Anwendung dieser Technologie für die Entwicklung von Mobile Apps, besonders unter der Berücksichtigung der Plattformunabhängigkeit, ist daher naheliegend.

Aufgrund der Beschaffenheit des Web impliziert seine Nutzung einen ganz bestimmten Architekturstil: Representational State Transfer (REST) oder auch ressourcenorientierte Architektur (ROA). Letztlich handelt es sich hier um eine einfache Client-Server-Architektur, die den Spezifikationen des HTTP-Standard entspricht und HTML-Dokumente auslieferiert, wie in Abb. 26.2 kurz erläutert [6].

Während das Desktop-Internet in erster Linie für die Anwendung auf statioären Computern und mit verlässlicher Internetverbindung konzipiert war, sind in der Smartphone-Welt ganz andere, neue Herausforderungen zu bewältigen.

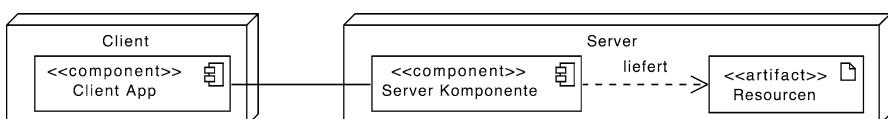


Abb. 26.2 Web-Architektur [6]

<sup>6</sup> <http://www.w3.org/TR/1999/REC-html401-19991224/>

<sup>7</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616.html>

### ***26.3.1 User Interfaces***

Web Apps bieten über Plattformgrenzen hinweg einheitliche User Interfaces. Hier stehen die Elemente der Webstandards zur Verfügung, die eine Vielzahl von unterschiedlichen Gestaltungsmöglichkeiten erlauben. Auf den verschiedenen Smartphone-Plattformen haben sich allerdings einige nicht direkt unterstützte HTML-Elemente als zielführend erwiesen. So hat sich beispielsweise eine eindimensionale Liste zur Navigation durchgesetzt. Dieses Element wird allerdings auf jeder Plattform anders dargestellt. Zwar lässt sich solch eine Liste mit einer Web App nachbauen, aber diese wird entweder unnatürlich aussehen oder muss für jede Plattform eigens entworfen werden, was dem Multiplattform-Ansatz widerspricht. Native Benutzeroberflächen wirken in der Regel hochwertiger.

### ***26.3.2 Instabile Verbindungen***

Der mobile Kontext bringt die Problematik mit sich, dass Verbindungen unterwegs aufgebaut und genutzt werden, die langsam oder aufgrund von schlechtem Empfang instabil sind. Normale Webseiten folgen dem Lazy-Loading-Prinzip. Das bedeutet, dass HTML-Dokumente immer dann geladen werden, wenn sie benötigt werden. Eine Web App besteht häufig aber aus einer Reihe von HTML-Dokumenten, die so Stück für Stück geladen werden müssen. Bei Verbindungsabbrüchen bedeutet dies, dass der Benutzer seine Arbeit mit der Applikation beenden oder unterbrechen muss, mindestens bis wieder eine Datenfunkverbindung besteht. Das unter dem Schlagwort AJAX (Asynchronous JavaScript und XML) zusammengefasste Technologie-Set sowie die HTML5-Spezifikation helfen dieses Problem zu verringern, indem sie Layout und Daten entkoppeln und clientseitige Caching-mechanismen erweitern und ausnutzen. Sie lösen aber das Grundproblem einer ressourcenorientierten Architektur nicht: die Abhängigkeit von diesen Ressourcen.

Für eine Web App bedeutet dies, dass in der Entwurfs- und Implementierungsphase stets Verbindungsabbrüche beachtet werden müssen und der Nutzer entweder darüber informiert oder ein eingeschränktes Weiterarbeiten ermöglicht werden muss. Für letztere Variante bedeutet dies, dass die gesamte Applikation im Idealfall nur aus einem einzigen Dokument, welches Daten asynchron bei Bedarf nachlädt, besteht. Dafür muss der Benutzer dann allerdings erhöhte Wartezeiten in Kauf nehmen; insbesondere bei instabilen Internetverbindungen und „vergesslichen“ Browsern kann dies schnell zu Frustrations- und Ablehnungsreaktionen führen.

### 26.3.3 Browser

Bereits im Zeitalter des Desktop-Internets kämpften Webdesigner mit der unterschiedlichen Interpretation des HTML-Standards durch einzelne Browser. Das prominenteste Beispiel ist hier der Internet Explorer 6, dessen eigenwillige Interpretation der Webstandards selbst Microsoft als Hersteller dazu bewegte, an der Ausrottung dieser veralteten Software mitzuwirken. Noch heute nutzen fast 11 % der Internetnutzer diesen 10 Jahre alten Brower [11]. Für andere Softwarehersteller gestaltete sich eine adäquate Interpretation ebenso schwierig. Erst neun Jahre nach der Veröffentlichung des HTML4.01-Standards bestand der erste Brower den ACID3-Test<sup>8</sup>, der als Referenz für die Umsetzung der Webstandards gilt.

Dieses Problem wiederholt sich nun in der Welt der Web Apps, bei der Interpretation des HTML5-Standards sowie der Unterstützung hardwarenaher Sensoren. Allerdings verschlimmert es sich, da im Gegensatz zum Desktop-Internet kein dominierendes Betriebssystem existiert. Für jedes Smartphone-Betriebssystem existiert nun eine Reihe von Browern, die zum Teil auf unterschiedliche Webengines zurückgreifen. Somit wird die Abhängigkeit der Web App vom Betriebssystem immer stärker. Musste man früher lediglich gegen drei bis vier prominente Brower testen, ist es jetzt notwendig geworden, drei bis vier Brower auf jeweils vier bis fünf Smartphone-Betriebssystemen und eventuell unterschiedlichen Geräten zu testen, um eine qualitativ hochwertige Web App zu entwickeln. Es existieren allerdings bereits erste JS Frameworks (wie zum Beispiel JQuery Mobile<sup>9</sup> oder Sencha touch<sup>10</sup>) die versuchen, die Unterschiede der mobilen Web-Brower zu abstrahieren [12].

### 26.3.4 Gerätesensorik

Die Spezifikation von HTML und JS sieht zunächst keinerlei Unterstützung für hardwarenahe Funktionen vor, ganz im Gegensatz zu Audio- und Videounterstützung [13]. Mit JS existiert aber eine vollwertige Programmiersprache, die auch mithilfe von Bibliotheken erweitert werden kann. Betriebssystemhersteller stellen hierfür eigens entwickelte Bibliotheken bereit, die Zugriff auf hardwarenahe Funktionen des Gerätes ermöglichen. Dem steht aber wieder die Abhängigkeit zu den Betriebssystemen gegenüber, was den Entwicklungsaufwand deutlich erhöht. Der Zugriff ist hier auch sehr eingeschränkt; so ist es zum Beispiel nicht möglich, mit einer Web App die Position eines Benutzers ausfindig zu machen, wenn der Brower nicht aktiv ist. Hier unterscheiden sich auch die Technologielebenszyklen von

---

<sup>8</sup> <http://www.webstandards.org/action/acid3/>

<sup>9</sup> <http://jquerymobile.com/>

<sup>10</sup> <http://www.sencha.com/products/touch/>

Smartphones (ein Jahr<sup>11,12</sup>) und die von Webstandards (ca. zehn Jahre<sup>13,14</sup>) zu stark, als dass eine optimale Unterstützung zu erwarten wäre. Letztlich handelt es sich bei den Browsern der Smartphones auch nur um native Apps, die regelmäßig gewartet und aktualisiert werden müssen. Neue Funktionen des Betriebssystems werden daher in der Regel erst von nativen Applikationen unterstützt, bis sie ihren Weg ins mobile Web finden. Die bereits oben genannten Javascript Frameworks versuchen auch hier, systemnahe Funktionen zu einem einheitlichen API zu abstrahieren [12].

## 26.4 Hybrid Apps

Einheitliche APIs für hardwarenahe Sensorikaufzüge wie das Abfangen von Touch-Gesten über die verschiedenen Plattformen hinweg bieten Hybrid Apps; aktuell prominent vertreten durch Produkte wie PhoneGap und Titanium. Hier werden Webstandards für eine hybride Lösung aus nativer und Web App genutzt, die dennoch auf mehreren Smartphone-Betriebssystemen lauffähig ist. Dabei werden HTML- und JS-Dateien als Ressourcen verwendet und in eine App eingebettet, die dann in den unterschiedlichen Stores eingereicht werden kann. Mit dieser Technik können auch Ressourcen wie Bilder und Daten mit ausgeliefert werden, so dass ein Webserver optional wird.

### 26.4.1 PhoneGap

PhoneGap verwendet einen vergleichsweise einfachen Ansatz, um aus einer Web App eine Hybrid App zu erstellen. Alle Smartphone-Betriebssysteme werden mit User-Interface-Klassen ausgeliefert, die eine Webseite eingebettet darstellen können. So können sowohl Ressourcen aus dem Web, als auch lokale verfügbare HTML-Dokumente in einer nativen App angezeigt werden. Dies kann hilfreich sein, um z. B. Text entsprechend zu formatieren.

PhoneGap nutzt diese Möglichkeit und lässt Entwickler ihre Apps komplett in HTML entwerfen. Darüber hinaus stellt PhoneGap einige native Funktionen über JS-Bibliotheken bereit. Zugriff auf die Kamera, den Kompass, Geodaten oder auch das Dateisystem werden so ermöglicht [14]. Die PhoneGap-Bibliothek registriert sich dazu als URL-Schema im System (z. B. `gap://`). Hinter diesen Aufrufen verborgen sich native Bibliotheken, die für die unterstützten Plattformen entwickelt wurden. Es besteht die Möglichkeit, auf die Eingabe des Benutzers (z. B. eine Geste) zu reagieren, indem eine JS-Routine registriert wird, die bei Ausführung der Geste

<sup>11</sup> [http://en.wikipedia.org/wiki/IOS\\_version\\_history](http://en.wikipedia.org/wiki/IOS_version_history)

<sup>12</sup> [http://en.wikipedia.org/wiki/Android\\_version\\_history](http://en.wikipedia.org/wiki/Android_version_history)

<sup>13</sup> <http://www.w3.org/TR/html401/>

<sup>14</sup> <http://dev.w3.org/html5/spec/Overview.html>

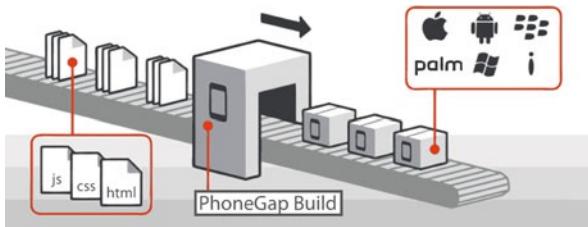


Abb. 26.3 PhoneGap Build-Process [16]

aufgerufen wird. Durch weitere Registrierungen kann auf Positionsänderungen oder Beschleunigungen reagiert werden [15].

Der Build-Process wird, wie in Abb. 26.3 zu sehen, von PhoneGap in der Cloud durchgeführt. Die fertig entwickelte App wird an die Cloud übermittelt, die für die entsprechenden Betriebssysteme Binaries zusammenbaut. Dabei werden native Bibliotheken und die übermittelten Ressourcen gebunden und für die entsprechenden Stores fertig kompiliert. Anschließend kann der Entwickler die fertig gebaute Hybrid App in den einzelnen Stores einstellen. Diese Lösung hat den Charme, dass die Entwicklungswerzeuge keinen festen Releasezyklen unterstehen, sondern jederzeit auf den neuesten Stand der Technik angepasst werden können.

## 26.4.2 Titanium

Das Titanium-Framework geht einen Schritt weiter und erzeugt nicht nur einfache webbasierte User Interfaces, sondern versucht so weit wie möglich native Objekte zur Laufzeit zu generieren. So entsteht das Gefühl, ein für die entsprechende Plattform entworfenes App-Design zu nutzen und zu erleben. Das Titanium-Framework analysiert den JS-Code und erzeugt native Symbole. Letztlich findet ein Übersetzungsprozess statt und ein Transfer von JS-Code in plattformspezifische ausführbare Elemente. JS wird hier vielmehr als eine Art Domain Specific Language eingesetzt

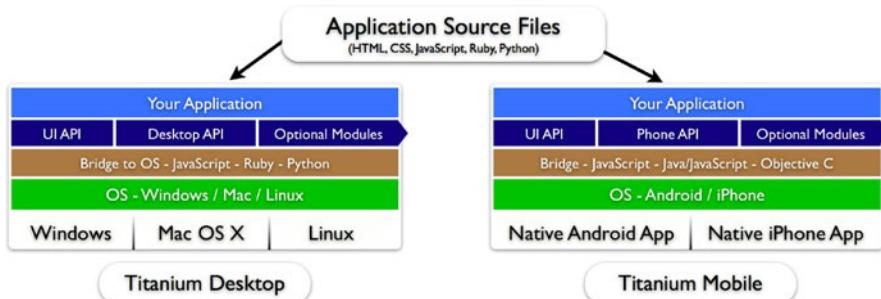


Abb. 26.4 Titanium-Architektur [17]

und weniger für die Manipulation des Document Object Models, für das es konzipiert wurde.

Titanium bietet ein deutlich größeres API für systemnahe Funktionen an als PhoneGap. Neben Geodaten, Beschleunigungssensor, Kamera, etc. sind auch analytische Funktionen sowie Schnittstellen zu Facebook bereits enthalten.<sup>15</sup> Derzeit werden allerdings nur zwei Plattformen unterstützt: iOS und Android. Zusätzlich lassen sich auch Desktop-Applikationen über diesen Weg erstellen.

Titanium kommt mit einem eigenen Eclipse-basierten SDK, mit dem das Komplizieren der Applikationen lokal möglich ist. Die fertigen Applikationen können dann über die entsprechenden Stores verkauft werden.

## 26.5 Cross-Compiler

Richten sich Hybrid-Apps vor allem an Webentwickler und versuchen deren Fähigkeiten optimal in der Smart-Apps-Welt zu nutzen, ist es mit Hilfe von Cross-Compilern möglich, in Programmierumgebungen wie Microsoft .NET für App Plattformen zu entwickeln. Statt JS-Code zu interpretieren wird die Programmierumgebung an die Laufzeitumgebung der Zielplattform gebunden. Anstatt für beispielsweise .NET lauffähige Applikationen zu erzeugen, werden iOS oder Android-Applikationen kompiliert.

### 26.5.1 MonoTouch

MonoTouch richtet sich in erster Linie an .NET-Entwickler und ermöglichte ursprünglich nur die Entwicklung von iOS-Apps. Die Entwicklung von Android-Applikationen ist damit mittlerweile auch möglich. Als Programmiersprache kommt C# zum Einsatz. Da Objective-C einige Funktionen der Sprache C# nicht unterstützt, ist der Entwickler gezwungen auf diese zu verzichten. So werden Generics nur eingeschränkt unterstützt, Invoke und Reflection überhaupt nicht [18]. MonoTouch stellt APIs für die Nutzung der systemnahen Funktionen bereit, ähnlich wie bei Hybrid Apps. Es ist allerdings auch möglich, native Bibliotheken der Zielplattform zu linken und diese in einer MonoTouch-App zu nutzen. Die Entwicklung von MonoTouch-Apps unterliegt damit weniger den Releasezyklen der MonoTouch-Entwickler, da neue Frameworkfunktionen direkt eingebunden werden können [5]. Es steht sowohl eine spezielle IDE namens MonoDevelop zur Verfügung, als auch eine Projektvorlage für Apples Xcode. Das Bauen der Applikation erfolgt lokal und diese kann seit der Lockerung von Apples Entwicklerbestimmung Mitte 2010 auch im AppStore veröffentlicht werden [19].

---

<sup>15</sup> <http://developer.appcelerator.com/apidoc/mobile/latest>

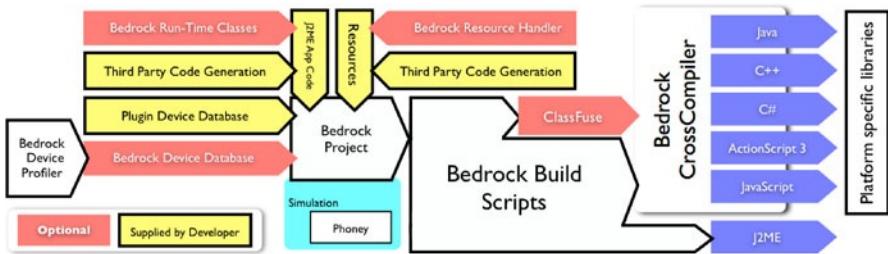


Abb. 26.5 Bedrock-Architektur [21]

### 26.5.2 Bedrock

Bedrock ist eine auf Mobilgeräte optimierte Crossplattform-Middleware und basiert auf Java. Sie setzt J2ME (Java Micro Edition) als Programmiersprache ein. J2ME ist für den Einsatz in eingebetteten und mobilen Geräten konzipiert [21]. Der Bedrock Cross-Compiler übersetzt den J2ME-Code in den Quellcode der jeweiligen Zielsprache (C++, C#, ActionScript oder JS).

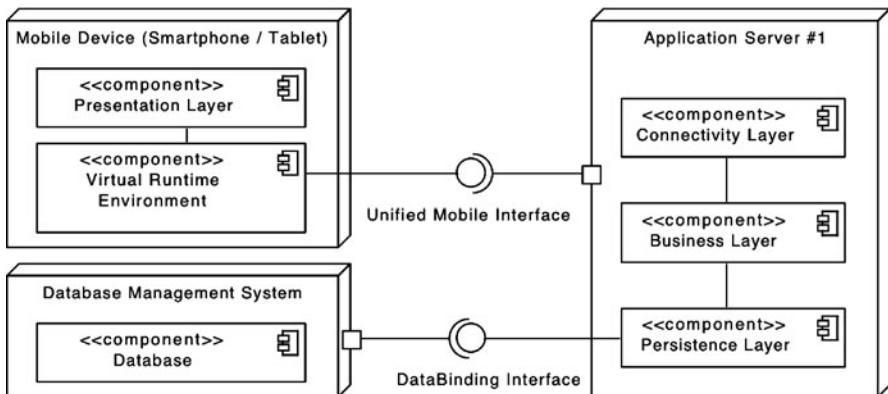
Anschließend kann der übersetzte Code für die entsprechende Zielplattform kompiliert werden. Abbildung 26.5 verdeutlicht den Aufbau der Bedrock-Architektur. Bedrock konzentriert sich auf die Übersetzung in die jeweilige Zielsprache, kann aber plattformspezifische Aspekte wie Frameworks für die Erzeugung des User-Interfaces nicht abbilden. Dieses System ist für 14 Systeme verfügbar, unterstützt aber nur sehr eingeschränkte Funktionalität. Bedrock ist daher prädestiniert für die Entwicklung von Spielen, die auf OpenGL basieren, da hier bereits ein plattformübergreifender Standard existiert.

## 26.6 Applikationsbeschreibungssprachen

Eine Alternative zu HTML-basierten Applikationen bieten Applikationsbeschreibungssprachen (ApplicationDescriptionLanguages-ADL), wie das App Construction Kit (AppConKit<sup>16</sup>). Diese nutzen anstelle von Webtechnologien eigens für die Smartphone-Welt konzipierte Sprachen. Anders als bei HTML ist man hier nicht von Spezifikationsgremien oder Browserherstellern abhängig, sondern nur von den Technologieanbietern. ADLs basieren im Grunde auf dem MVC-Pattern und trennen sehr strikt zwischen Daten, Aktionen und dem Layout [7]. Anders als bei Hybrid Apps werden hier nicht die Sprachen des Web explizit genutzt, sondern die Architektur in Form von REST-Services, wie in Abb. 26.6 zu sehen ist.

ADLs beschreiben ein User Interface möglichst genau, aber abstrakt über unterschiedliche Smartphone-Betriebssysteme hinweg. So muss eine mobile App lediglich einmal entwickelt werden und wirkt dennoch völlig nativ auf der jeweiligen

<sup>16</sup> <http://www.weptun.de/portfolio/appconkit>



**Abb. 26.6** Architektur von ADL-basierten Applikationen

Zielplattform. Das User Interface kann in XML- oder JSON-Dateien zerlegt werden und als Ressource der App eingebunden oder über die Netzverbindung nachträglich bezogen werden. Die Übersetzung in die Zielumgebung kann hier sowohl beim Kompilieren, als auch zur Laufzeit der App erfolgen. Dadurch kann die Flexibilität einer Web-Anwendung mit der vollen Lauffähigkeit bei instabilen Netzwerkverbindungen kombiniert werden. Das Design erfolgt hier häufig toolgestützt, mit Hilfe von Interface Buildern, so dass wenige Programmierkenntnisse notwendig sind, um eine App zu erstellen. Die Funktionalität beschränkt sich hier, wie auch bei Hybrid-Apps, auf die verfügbaren APIs für hardwarenahe Aktionen.

## 26.7 Anwendungsgebiete

Für alle oben genannten Technologien gibt es bestimmte Anwendungsgebiete, deren Anforderungen sie besonders gut erfüllen. Plattformunabhängige Architekturen werden mehr und mehr notwendig, um die Heterogenität des Marktes mit überschaubaren Kosten zu durchdringen.

Für Anwendungen mit höchsten Anforderungen an Leistung oder Design bieten sich natürlich native Applikationen an. Allerdings sind hier die Entwicklungskosten zum Unterstützen mehrerer Mobile-Plattformen enorm.

Mobile Web Apps hingegen eignen sich vor allem zur kostengünstigen Bereitstellung schon bestehender Web-Anwendungen. Mit wenig Aufwand und bekannter Technologie kann schnell Content bereitgestellt werden, der sinnvoll mobil genutzt werden kann. So lassen sich die Reviewprozesse der einzelnen Stores umgehen und die Bindung an ein vorgegebenes Geschäftsmodell ist nicht notwendig. Auf die Funktionalitäten der neuen Technologien muss allerdings größtenteils verzichtet werden.

Hybrid-Apps stellen eine Alternative für Apps dar, die gezielt für den Smartphone- oder Tablet-Markt konzipiert sind. Wie gut die jeweilige Plattform unterstützt wird, ist allerdings stark vom jeweiligen Technologieanbieter abhängig.

Cross-Compiler eignen sich besonders für Umsteiger aus der .NET- oder Java-Welt. Die gewohnte Sprache und plattformspezifische vorhandene Bibliotheken können hier wiederverwendet werden. Cross-Compiler bieten sich vor allem für Umsteiger und Spieleentwickler an.

Mit Applikationssprachen entsteht so etwas wie der goldene Mittelweg. Die Flexibilität ist nicht so groß wie im Web-App-Markt, dafür erhält man native Anwendungen, die auf die jeweiligen Plattformen optimiert sind. Besonders für Entwickler mit wenig Programmiererfahrung eignet sich diese Technologie, da sie die architektonischen und programmatischen Probleme der Zielplattformen abstrahiert. Aufgrund der Möglichkeit, bestehende Services mit Hilfe von REST anzudocken, ist dieser Weg bestens geeignet für plattformübergreifende Business Apps.

## 26.8 Fazit

Für einige Experten sind Modelle wie Apples AppStore bereits jetzt ein Auslaufmodell [22]. Allerdings wurde auch das Internet noch Anfang der 90er Jahre nur als „Modeerscheinung“ bezeichnet [23]. Dennoch hat sich diese Technologie durchgesetzt und vieles spricht dafür, dass dies auch für Apps gelten wird [24].

Die enormen Kosten, die bei der Entwicklung von Apps entstehen, wenn diese auf verschiedenen mobilen Betriebssystemen laufen sollen, zwingen die Softwareentwickler früher oder später Multiplattformtechniken einzusetzen. Drittanbieter ermöglichen dank intelligentem Architekturdesign bereits jetzt plattformübergreifende App-Entwicklung. Gezielte Technologieauswahl ist hier der Schlüssel zum Erfolg, getrieben von den fachlichen Anforderungen und den technischen Fertigkeiten des Projektteams. Web-Entwickler sind sicher besser beraten mit Hybrid Apps zu arbeiten, anstatt sich für Cross-Compiler zu entscheiden.

Anders als bei klassischen Anwendungen ist im Bereich der Spieleentwicklung eine homogenere Systemlandschaft entstanden. Mit der Basistechnologie OpenGL ist hier bereits eine sehr ausdrucksstarke plattformübergreifende Sprache verfügbar, die von Framework-Anbietern wie Corona<sup>17</sup>, Unity<sup>18</sup> oder ShiVa<sup>19</sup> genutzt wird. Für klassische Anwendungen hat sich bislang keine Sprache durchgesetzt und Webtechnologien erweisen sich als zu ausdrucksschwach, wenn sie nicht durch umfangreiche Hybrid-Frameworks erweitert werden. Applikationsbeschreibungssprachen oder aus dem Desktop-Bereich bekannte Technologien wie QT<sup>20</sup> könnten diese Lücke füllen und eine übergreifende native Entwicklungslandschaft schaffen.

---

<sup>17</sup> <http://www.anscamobile.com/corona/>

<sup>18</sup> <http://unity3d.com/unity/>

<sup>19</sup> <http://www.stonetrip.com/>

<sup>20</sup> <http://qt.nokia.com/products/>

## Literaturverzeichnis

1. Restivo K, Llamas R, Shirer M (2011) Worldwide Smartphone Market Expected to Grow 55 % in 2011 and Approach Shipments of One Billion in 2015 *IDC – Press Release*, <http://www.idc.com/getdoc.jsp?containerId=prUS22871611> Zugriff am 11. Juni 2011.
2. Caolo D (2011) Apple unofficially reaches 500K approved iOS apps in the App Store. *TUAW*. <http://www.tuaw.com/2011/05/24/apple-unofficially-reaches-500k-approved-ios-apps-in-the-app-sto/> Zugriff am 25. Mai 2011
3. Perez S (2011) Android Market Share Numbers Questioned. *ReadWriteWeb*. [http://www.readwriteweb.com/archives/android\\_market\\_share\\_numbers\\_questioned.php](http://www.readwriteweb.com/archives/android_market_share_numbers_questioned.php) Zugriff am 11. Juni 2011
4. Ableson F (2009) Android and iPhone browser wars. *developerWorks*. <http://www.ibm.com/developerworksopensource/library/os-androidiphone1/> Zugriff am 11. Juni 2011
5. B. Costanich (2009) MonoTouch: .NET Development for the iPhone. *InfoQ*. <http://www.infoq.com/articles/monotouch-introduction> Zugriff am 11. Juni 2011
6. Fielding R (2000) Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine.
7. Gamma E et al (1994) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional
8. Berners-Lee T and Jaffe J (1999) The World Wide Web Consortium (W3C). W3C <http://www.w3.org/> Zugriff am 13. Juni 2011
9. ECMA International (2009) ECMA-262 ECMAScript Language Specification
10. Bos B et al. (2011) Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification W3C <http://www.w3.org/TR/2002/WD-CSS21-20020802/> Zugriff am 13. Juni 2011
11. Microsoft Corporation (2011) The Internet Explorer 6 Countdown. [www.theie6countdown.com](http://www.theie6countdown.com/) Zugriff am 15. Juni 2011
12. Mühlwitz G (2010) JavaScript goes Mobile: Überblick zu den wichtigsten WebApp-Frameworks. *t3n*. <http://t3n.de/news/javascript-mobile-ueberblick-wichtigsten-282544/> Zugriff am 15. Juni 2011
13. Way J (2010) 28 HTML5 Features, Tips, and Techniques you Must Know. *Nettuts+*. <http://nettutsplus.com/tutorials/html-css-techniques/25-html5-features-tips-and-techniques-you-must-know/> Zugriff am 14. Juni 2011
14. Nitobi Inc. (2011) PhoneGap Documentation. *PhoneGap Documentation*. <http://docs.phonegap.com/> Zugriff am 15. Juni 2011
15. Macfadyen J (2009) PhoneGap for iPhone exposed. *Jesse Macfadyen's Blog*. <http://blogs.nitobi.com/jesse/2009/11/04/phonegap-for-iphone-exposed/> Zugriff am 15. Juni 2011
16. Nitobi, Y.S. of (2011) Phonegap Artwork. <http://www.phonegap.com/artwork/> Zugriff am 15. Juni 2011
17. Appcelerator Inc. (2011) Titanium Platform. <http://www.appcelerator.com/products/titanium-cross-platform-application-development/> Zugriff am 15. Juni 2011
18. Novell Inc. (2011) MonoTouch Documentation <http://monotouch.net/> Zugriff am 15. Juni 2011
19. Grannemann K (2010) Apple erlaubt Fremdcode in iPhone-Apps: Kein Freifahrtschein für Packager und MonoTouch. *Macnotes*. <http://www.macnotes.de/2010/06/14/apple-erlaubt-fremdcode-in-iphone-apps-kein-freifahrtsschein-fuer-packager-und-monotouch/> Zugriff am 15. Juni 2011
20. Oracle and Sun (2011) Java ME and Java Card Technology, *Oracle Technology Network*. <http://www.oracle.com/technetwork/java/javame/index.html> Zugriff am 16. Juni 2011
21. Metismo Ltd (2011) Bedrock Architecture. *Metismo*. <http://www.metismo.com/architecture.html> Zugriff am 11. Juni 2011
22. Mehl W (2011) Experte erwartet das Ende des App Store. *iPhoneWelt*. [http://www.macwelt.de/artikel/\\_News/376345/experte\\_erwartet\\_das\\_ende\\_des\\_app\\_store/1](http://www.macwelt.de/artikel/_News/376345/experte_erwartet_das_ende_des_app_store/1) Zugriff am 16. Juni 2011

23. Dariani M (2009) Online-Werbung wächst im Krisenjahr 2009 um 10%. *Plista*. <http://blog.plista.com/2009/12/online-werbung-wachst-im-krisenjahr-2009-um-10/> Zugriff am 16. Juni 2011
24. Meeker et al (2009) The Mobile Internet Report Setup. Morgan Stanley Mobile Internet Report 9:104

---

# Kapitel 27

## Programmierung von Smart Mobile Apps

### Ein Spagat zwischen Stil, Performanz und Benutzerfreundlichkeit

Thomas Strang und Michael Lichtenstern

**Zusammenfassung** Die Programmierung von Smart Mobile Apps unterscheidet sich in einigen Punkten deutlich von der PC- und Webbasierten Anwendungen. Die Größe der Anzeige, die Performance der Geräte und die typischen Entwicklungs- und Debuggingumgebungen mit virtuellen Endgeräten stellen dabei nur eine Seite der besonderen Herausforderung dar. So mancher gute Programmierstil und die Wartungsfreundlichkeit der Software sind in der Praxis schon Performancegewinnen geopfert worden. Auch die Nutzung der Geräte gerade in jeglicher Alltagssituation sollte stets bedacht werden, da nicht überall die 100 %ige mentale Kapazität des Nutzers zur Verfügung steht. Hier liefert der stets wachsende Umfang von neuen Sensoren auf den Endgeräten zusammen mit der Vernetzung zur Backend-Infrastruktur ein immenses Potential, die Benutzerfreundlichkeit (Usability) der Smart Mobile Apps durch Abstimmung auf den gerade vorliegenden Nutzungskontext zu maximieren. Dabei ist die Verwendung von Kontextinformation wie dem aktuellen Aufenthaltsort als Informationsfilter nur ein allererster Schritt. Dieser Artikel widmet sich daher dem Spagat zwischen Programmierstil, Performanz sowie Benutzerfreundlichkeit von Smart Apps für iPhone und Co. und versucht Ansätze für einen gelungenen Kompromiss aufzuzeigen.

### 27.1 Einleitung

Kaum ein modernes Unternehmen am Markt kommt an diesem Trend vorbei – wer etwas auf sich hält, stellt eine Software für das Mobiltelefon seiner Kunden bereit.

---

Thomas Strang  
Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Kommunikation und Navigation,  
Wessling/Oberpfaffenhofen,  
E-mail: thomas.strang@dlr.de

Michael Lichtenstern  
Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Kommunikation und Navigation,  
Wessling/Oberpfaffenhofen,  
E-mail: m.lichtenstern@dlr.de

Diese Software wird *Smart Mobile App* oder kurz *App* genannt. Der funktionale Umfang der Apps ist beinahe so vielfältig wie die Anzahl der verfügbaren Apps in den einschlägigen Datenbanken. So können aktuell (Juni 2011) beispielsweise etwa 400.000 Apps im Apple AppStore für iPhone und Co. heruntergeladen werden [1]. Neben den eigentlichen Diensten ist die Kundenbindung über regelmäßige Kommunikation eine der Motivationen der Anbieter der Apps.

Die Namensbestandteile „Smart“ und „Mobile“ geben einen Hinweis, worin der enorme Erfolg dieser Apps in den letzten Jahren begründet ist: Durch die Verlagerung eines großen Teils der Applikationslogik auf das mobile Endgerät kann eine flüssigere und integrierte, notfalls auch von Netzwerkverbindungen unabhängige App gestaltet werden, was vom Nutzer unter anderem als Performanzgewinn gegenüber webbasierten mobilen Anwendungen wahrgenommen wird. Dabei ist die Nutzung von mobilen Endgeräten als Applikationsplattform nicht unbedingt eine Erfindung der letzten Jahre – schon mit Aufkommen der ersten *SmartPhones* nach der Jahrtausendwende wurde das Konzept von Smart Mobile Apps zusammen mit entsprechenden Portalen zum Herunterladen individueller Zusammenstellungen von Apps gezeigt [2]. Zu dieser Zeit war die Leistungsfähigkeit der mobilen Endgeräte allerdings noch sehr limitiert. Insbesondere die Rechenleistung und die Speicherkapazität waren die einschränkenden Faktoren für den Erfolg der Apps dieser Zeit.

Eine Dekade später stellt sich dies mittlerweile ganz anders dar. Die Rechenleistung heutiger Mobiltelefone ist höher als die von PCs noch vor wenigen Jahren. Ähnlich verhält es sich mit dem flüchtigen wie nicht-flüchtigen Speicher. In der Spezifikation der Java2 Micro Edition (J2ME) für Mobiltelefone von 2002 lag die Anforderung für das mobile Endgerät bzgl. Speicher bei „256 kilobytes of non-volatile memory for the MIDP implementation, 8 kilobytes of non-volatile memory for application-created persistent data, 128 kilobytes of volatile memory for the Java runtime“ [3]. Zum Vergleich: iPhones der Generation 4 sind mit bis zu 32 Gigabyte nicht-flüchtigem Flash-Speicher und zusätzlichen 512 Megabytes Hauptspeicher ausgestattet. Rechenleistung und Speicher stellen somit heute keine echte Limitierung mehr dar. Die verfügbare Bandbreite der Kommunikationsverbindungen (und deren Kosten) sowie die Batteriekapazität werden derzeit eher als beschränkende Faktoren wahrgenommen. Der Wegfall dieser früheren Leistungsbeschränkungen hat immens zum jüngsten Erfolg der Smart Mobile Apps beigetragen.

Gleichermaßen zuträglich für den Erfolg waren sowohl die quantitative Zunahme wie auch die qualitative Ausprägung der Ausstattung der mobilen Endgeräte. Heutige Mobiltelefone haben eine Vielzahl an Sensoren wie beispielsweise GPS, Kompass, Beschleunigungs-, Temperatur-, Helligkeits- oder Drucksensoren, die sehr gewinnbringend für Smart Mobile Apps eingesetzt werden können. Die Entwickler vieler Apps machen zunehmend Gebrauch von diesen Möglichkeiten, wie das Beispiel des Pairing zwischen Bluetooth-Headset und Telefon durch gemeinsames „Schütteln“ beider Geräte [4] oder die „Sleep Cycle“ App [5] zeigen. Letztere wertet die Bewegungen eines schlafenden Menschen mittels der Beschleunigungssensoren aus und ist dadurch in der Lage, einen besonders günstigen Zeitpunkt zum Wecken

der Person zu ermitteln. Auch die Ein- und Ausgabefunktionen heutiger Mobiltelefone bieten Möglichkeiten, die auch professionellen Ansprüchen z. B. hinsichtlich Auflösung, Ästhetik und Benutzerfreundlichkeit genügen.

Smart Mobile Apps werden heute in allen möglichen Alltagssituationen genutzt. Dies stellt eine besondere Herausforderung an die Entwicklung mobiler Apps dar, da nicht zu jeder Zeit die 100 %ige mentale und kognitive Kapazität der Benutzer vorausgesetzt werden kann. Es ist besonders zu beachten, dass Apps auch (und vielleicht gerade) in kritischen, möglicherweise für den Nutzer gefährlichen Umgebungsbedingungen wie im Straßenverkehr genutzt werden. Deshalb ist der Entwickler bereits zur Entwurfszeit gefordert, möglichst optimale Interaktionsmethoden für typische Nutzungsszenarien vorzusehen. Dies beinhaltet z. B. die zusätzliche Verwendung von akustischen oder haptischen Hinweisen statt lediglich Bildschirmausgaben von Apps.

Auch Eingaben sind nicht nur auf die typischen Einhandeingaben von Mobiltelefonen reduziert. An dieser Stelle setzen viele Entwickler an und nutzen die Möglichkeiten, die sich durch die Vielfalt der Sensoren auf modernen mobilen Endgeräten ergeben. Statt beispielsweise bei einer Busfahrplananwendung Start- und Ziel-Haltestelle über die Tastatur vom Benutzer zu erfragen, wird über den GPS-Empfänger des Geräts der aktuelle Aufenthaltsort bestimmt, aus welchem sich die nächste Haltestelle ableiten lässt, die dann wiederum als Start-Haltestelle voreingestellt wird. Gleichermaßen kann auch die Personalisierung von Apps dazu beitragen, Anwenderspezifika wie immer wiederkehrende Nutzungsmuster gewinnbringend in der Applikationslogik mit einzubeziehen. Umfangreiche Forschungsarbeiten zur optimalen Gestaltung der Schnittstelle zwischen Mensch und Gerät (*Human Machine Interface, HMI*) sowie situationsangepasster Dienstnutzung (*Context Awareness*) wurden in den letzten 10 Jahren in der Forschungslandschaft dazu erarbeitet [6, 7].

## 27.2 Auswirkungen auf die Programmierung

Die Entwicklung von Smart Mobile Apps erfolgt in der Regel auf der Ebene einer höheren Programmiersprache wie Java oder C++. Es wird jedoch eine reduzierte Variante der Programmiersprache verwendet, die sich in einigen spezifischen Details von der Standardversion der Programmiersprache und ihren Bibliotheken unterscheidet. Insbesondere werden die Besonderheiten der Zielplattformen berücksichtigt, wie etwa die Beschränkungen in der Displaygröße oder die Tatsache, dass Mobiltelefone oftmals keine Festplatte mit sich bringen. Dafür werden für den Zugriff auf besondere Hardwareausstattungen wie Beschleunigungssensoren oder Kameras entsprechende Programmbibliotheken zur Verfügung gestellt, auf welche in der Standardversion der Programmierumgebungen nicht zugegriffen werden kann.

Als Beispiel sei an dieser Stelle Java angeführt. Diese Programmiersprache gibt es in mehreren Editionen, wobei die *Java2 Micro Edition* für Mobiltelefone kon-

ziert wurde. Da auf Mobiltelefonen der Umfang des nicht-flüchtigen Speichers für Java-Programme zunächst sehr begrenzt war, stellt J2ME, genauer gesagt die Kombination des *Mobile Information Device Profiles (MIDP)* zusammen mit der *Connected Limited Device Configuration* kein vollständiges Dateisystem zur Verfügung. Stattdessen kann eine Art Mini-Datenbank, das sogenannte *Record Management System*, zur persistenten Speicherung von Daten verwendet werden. Das hat unmittelbar Auswirkungen auf die Programmierung von entsprechender Software, da hier nicht in gewohnter Weise auf Dateien in einer Ordnerstruktur zugegriffen werden kann. Des Weiteren ist zu beachten, dass aufgrund der Verwendung von Flash-Speicher zur Bereitstellung von persistentem Speicher Schreibzugriffe ungleich langsamer sind als Lesezugriffe. Schreibender Zugriff auf persistenten Speicher ist daher nach Möglichkeit auf ein Minimum zu reduzieren, was sich je nach Applikationslogik mehr oder weniger gut umsetzen lässt.

Auch die Gestaltung einer Benutzeroberfläche für eine App stellt den Entwickler vor neue Herausforderungen. Im Vergleich zu einer Desktop-Applikation ist bei einer Smart Mobile App die verfügbare Auflösung des Bildschirms meist limitiert. Um die Bildschirmgröße optimal zu nutzen, kann der Benutzer bei einigen mobilen Endgeräten mit Beschleunigungssensor beispielsweise durch Drehen des Geräts um 90° die Breite der Anzeige zu Lasten der angezeigten Höhe nahezu verdoppeln. Ein solches dynamisches Verhalten der Benutzeroberfläche zur Laufzeit der App sollte bei der Planung, wie auch der Programmierung immer beachtet werden. Ähnliche Aspekte gilt es bei der Benutzereingabe zu beachten: Bei Desktop-Anwendungen werden Eingaben in die Software üblicherweise über Maus und Tastatur getätig, wohingegen bei Android und Co. die Steuerung und Dateneingabe über eine Tastensteuerung oder über eine berührungssensitive Anzeigoberfläche durchgeführt wird. Das Drücken eines zu klein geratenen Buttons mit dem Finger kann sich folglich als ungleich schwieriger herausstellen. Entsprechend sollte bei der Gestaltung von Apps auf kleine Schaltflächen und Menüelemente zu Gunsten der Bedienbarkeit verzichtet werden.

Die notwendige Anpassung der Programmierung für die Gestaltung der Benutzeroberfläche ist oftmals mit erheblichem Mehraufwand verbunden. Hier sehen viele Programmierumgebungen zwei mögliche Paradigmen vor: einerseits eine sogenannte Highlevel-Programmierung und andererseits als Gegenstück die Lowlevel-Programmierung. Bei der Highlevel-Programmierung wird dem Programmierer soweit es geht die Detailarbeit abgenommen, was aber auch die Kontrolle über Design und Interaktionsmöglichkeiten reduziert. Demgegenüber bietet die Lowlevel-Programmierung mehr Freiheitsgrade in der Gestaltung der Benutzeroberfläche, wobei diese umfangreicheren Programmieraufwand mit sich bringt. Diese Unterscheidung wird bereits in der Java Standard Edition (*J2SE*) getroffen und ist beispielsweise der grundlegende Unterschied in den Grafikbibliotheken *Swing* und *AWT*. Bei der Programmierung der Benutzeroberflächen für Smart Mobile Apps sind aber noch weitere Aspekte zu beachten. Unter Android ist zum Beispiel die Benutzerschnittstelle in einer sogenannten *Activity* gekapselt. Eine *Activity* kann in ihrem Lebenszyklus unterschiedliche Zustände aufweisen, wie beispielsweise aktiv (*active*), wartend (*paused*), angehalten (*stopped*) oder abgeschlossen (*killed*). Alle

*Activities* werden in einem *activity stack* verwaltet, wodurch jede *Activity* über jeden Zustandswechsel informiert werden kann, zum Beispiel wenn eine App den Fokus verliert.

Durch das Sandbox-Konzept von Java wird sichergestellt, dass die entwickelte Software keinen Zugriff auf Komponenten des Hostsystems erhalten, für die sie nicht autorisiert sind. Dieses Sandbox-Konzept haben die Entwickler von J2ME an einigen entscheidenden Stellen ausgebaut und das Applikationsmanagement entsprechend angepasst. Apps können nur in Form von Archiven – sogenannten MIDlet Suiten – bestehend aus einer oder mehreren Apps auf den Mobiltelefonen installiert werden, von welchen zur Laufzeit immer nur eine aktiv sein darf. Viel entscheidender ist jedoch, dass während der Laufzeit der Apps nur auf Bibliotheken der jeweiligen MIDlet Suite zugegriffen werden kann. Bei oberflächlicher Betrachtung scheint dies vielleicht nicht besonders bemerkenswert zu sein, ist jedoch de facto eine große Einschränkung: Das Konzept des *late binding*, d. h. das Binden eines Unterprogrammaufrufs an eine zur Laufzeit (*at runtime*) zu bestimmende Implementierung stellt einen wesentlichen Bestandteil moderner Programmiersprachen dar. J2ME setzt hier jedoch das Konzept des *closed late binding* um, also die Möglichkeit des Bindens eines Unterprogrammaufrufs ausschließlich an Implementierungen, die zum Zeitpunkt der Installation der Suite (*at installtime*) bestimmt wurden. Dieser Umstand verhindert beispielsweise, dass nach der Installation einer App diese um weitere Komponenten ausgebaut werden kann. Stattdessen muss sie immer als Ganzes neu installiert werden, was auch mit dem Verlust etwaiger persistenter Daten einhergeht. Dies bedeutet, dass obwohl auch J2ME grundsätzlich ein *late binding* implementiert, die Nebenbedingungen von großer Bedeutung sind und eine erhebliche Anpassung der Programmierung und Einschränkung im Vergleich zur Standardprogrammierung darstellen.

Die Programmierung von Smart Mobile Apps erfordert also, an einigen Stellen von der etablierten Vorgehensweise in der Standardprogrammierung [8] abzuweichen. Dies erfolgt insbesondere im Spannungsfeld zwischen Programmierstil und Effizienz der Smart Mobile App auf dem jeweiligen Endgerät.

## 27.3 Stil vs. Effizienz

Für ein bestimmtes Problem gibt es meist mehrere Lösungswege, was auch bei Smart Mobile Apps nicht anders ist. Welcher Programmieransatz gewählt wird, sollte wohl bedacht sein, wobei im Vordergrund immer die Performanz stehen sollte. Gibt es zur Lösung einer Problemstellung mehrere Algorithmen in unterschiedlichen Laufzeit- oder Speicherkomplexitätsklassen, ist zunächst anhand der Anforderungen eine Vorauswahl hinsichtlich der optimalen Komplexitätsklasse zu treffen. Dabei gilt es die zu verarbeitende Datenmenge möglichst genau abzuschätzen, um beispielsweise für zu erwartende geringere Datenmengen einen Algorithmus auszuwählen, der eine höhere Speicherkomplexität, aber eine vergleichsweise bessere Laufzeitkomplexität hat. Erst nach dieser Vorauswahl ist es in der Regel sinnvoll,

sich um die Optimierung der Implementierung zu kümmern. Auch hier können signifikante Unterschiede erreicht werden. Grundsätzlich sollten, soweit es geht, die Designaspekte moderner Softwareentwicklung (*Use intention revealing names, small functions, concurrency defense* etc. [8]) Beachtung finden. Doch auch hier lassen sich Optimierungsmöglichkeiten hinsichtlich des Spagats zwischen Stil und Effizienz identifizieren.

Als Beispiel soll das Abfangen von Benutzereingaben mittels Eingabetasten in einer Android App dienen. In den folgenden Codefragmenten soll durch das Drücken des Buttons „Speichern“ oder „Abbruch“ der Benutzer die vorherigen Eingaben abspeichern oder den Vorgang abbrechen können. Die Verwendung der entsprechenden Objektklasse erfolgt nach folgendem Muster:

```
Button speichern = (Button) findViewById(R.id.speichernButton);
speichern.setText("speichern");
Button abbruch = (Button) findViewById(R.id.abbrechenButton);
abbruch.setText("abbrechen");
```

Zuerst müssen die entsprechenden Instanzen der Klasse Button einer Activity mit Hilfe der Methode findViewById(int id) geladen und anschließend ein Listener registriert werden. In der Praxis findet man nun häufig Beispiele, bei denen die Möglichkeiten der objektorientierten Programmierung umgangen werden:

```
// listener registrieren
ClickListener clickListener = new ClickListener();
speichern.setOnClickListener(clickListener);
abbruch.setOnClickListener(clickListener);

// Getrennte Klasse für beide Buttons
public class ClickListener implements View.OnClickListener {
    public void onClick(View view) {
        if (((Button) view).getText().toString().equals("speichern"))
            // speichern der Benutzereingaben
        else
            if (((Button) view).getText().toString().equals("abbrechen"))
                // abbrechen des Vorgangs
    }
}
```

Während diese Vorgehensweise grundsätzlich zum Ziel führt und für den Entwickler möglicherweise besonders simpel und wartungsfreundlich erscheint, gibt es erhebliche Probleme damit. So ist das hierbei notwendige String parsen vergleichsweise rechenaufwendig. Darüber hinaus gibt es auch eine Abhängigkeit zur Sprache, mit der das Button-Objekt betitelt wurde, was bei einer zukünftigen Internationalisierung der App zusätzlichen Programmieraufwand mit sich bringt. Wesentlich besser und effizienter ist die Verwendung eigener Listener für jedes Button-Objekt:

```
// listener registieren
speichern.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        // speichern der Benutzereingaben
    }
});
abbruch.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        // abbrechen des Vorgangs
    }
});
```

Ein weiteres Beispiel liegt in der geschickten Verwendung der verfügbaren Basiskomponenten. Nehmen wir beispielsweise an, die Aufgabe besteht in der Programmierung einer Smart Mobile App, die eine analoge Uhr darstellt. Eine mögliche Vorgehensweise wäre, die anzugezeigenden Elemente einer analogen Uhr mit Zeitskala und Zeigern mit Methoden der Lowlevel-Bibliotheken zu zeichnen und jede Sekunde alle Elemente neu zu zeichnen. Ein laufzeiteffizienterer Ansatz wäre es allerdings, die statischen Elemente einer Uhr (Zeitskala) mit einer Highlevel-Komponente wie einem Image einzulesen und als Hintergrund zu verwenden. Auf diesem Hintergrund brauchen dann nur noch die dynamischen Elemente (Stunden-, Minuten- und Sekundenzeiger) mit Lowlevel-Methoden gezeichnet zu werden. Durch diese relativ einfachen Tricks lassen sich in der Praxis enorme Effizienzsteigerungen erzielen.

Wie bereits erwähnt sind Rechenleistung und Speicherplatz heute nicht mehr die primär einschränkenden Faktoren auf mobilen Endgeräten. Dementsprechend sollten Smart Mobile Apps den verfügbaren Speicherplatz zur Optimierung der Programmfunctionalität ausnutzen. Dies kann zum Beispiel anhand der Serialisierung von Objekten im *Record Management System* dargestellt werden: J2ME sieht im Gegensatz zu Standard-Java grundsätzlich keine Methoden zur Serialisierung von Objekten vor. Dennoch ist die Serialisierung von Objekten zur Übertragung über das Netzwerk an andere Geräte oder zur Speicherung im nicht-flüchtigen Speicher z. B. bis zum nächsten Programmstart für viele moderne Apps sehr sinnvoll. Durch die heute relativ großzügige Bemessung des vorhandenen nicht-flüchtigen Speichers sollten dabei nicht nur die reinen Objektdaten hinterlegt werden, sondern zusätzlich auch Informationen über die Struktur bzw. den Inhalt der Objekte. Bei der sogenannten *selbstbeschreibenden Serialisierung* werden Informationen über die Klasse, von der das Objekt eine Instanz ist, die Version, die Datentypen der Variablen etc. mit in der serialisierten Version des Objekts hinterlegt:

```
// Serialisierung einer Instanz eines Adressen-Objekts
// in ein ByteArray
public byte[] toByteArray() {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    DataOutputStream dos = new DataOutputStream( baos );
```

```
try {
    dos.writeUTF( this.getClass().getName() );
    dos.writeUTF( this.getPLZ() );
    dos.writeUTF( this.getCity() );
    // ... fortfahren mit weiteren Attributen
}
}
```

Durch selbstbeschreibende Serialisierung wird die Deserialisierung in die Lage versetzt, ohne weitere Informationen oder gar Annahmen aus den serialisierten Daten wieder Objekte zu rekonstruieren. Dies verbessert nicht nur das Fehlermanagement, der Code wird auch erheblich flexibler einsetzbar. So können sich ganze Klassen über einen entsprechenden Aufruf im Konstruktor „selbst“ aus der Serialisierung reinstanziiieren, wie das folgende Beispiel zeigt.

```
// statische De-Serialisierungs-Funktion
static public Entry fromByteArray(byte[] ba) {
    ByteArrayInputStream bais = new ByteArrayInputStream(ba);
    DataInputStream dis = new DataInputStream(bais);
    return new Address(dis);
}

// Konstruktor
protected Address(DataInputStream dis) {
    try {
        String classname = dis.readUTF();
        if (!classname.equals(this.getClass().getName())) // check!
            throw new RuntimeException("Cannot_deserialize" +
                this.getClass().getName() + "from" + classname);
        setPLZ(dis.readUTF());
        setCity(dis.readUTF());
        // ... fortfahren mit weiteren Attributen
    }
}
```

Wie man sieht wird durch das Hinzufügen der Strukturinformationen (hier: der Klassename, von dem das Objekt instanziert wurde) eine effiziente Möglichkeit geschaffen, komplexe serialisierte Objekte z. B. im *Record Store Management* von J2ME als byte array abzulegen, um z. B. nach einem Neustart der App an der Stelle den Programmablauf fortzusetzen, an der die Serialisierung zum letzten Mal angestoßen wurde. Auf dieses Stilmittel sollte nicht verzichtet werden.

## 27.4 Steigerung der Benutzerfreundlichkeit

Die Benutzerfreundlichkeit von Smart Mobile Apps im Sinne der Usability definiert sich nach Nielsen [9] als „Qualitätsmaß, welches es erlaubt zu bemessen, wie leicht ein Nutzerinterface zu bedienen ist. Dies bezieht sich auch auf die Methoden zur

Steigerung der Benutzerfreundlichkeit während der Designphase“. Im Allgemeinen umfasst dieses Qualitätsmaß die Unterkriterien

- Erlernbarkeit – Wie leicht ist es für einen Nutzer bei der erstmaligen Nutzung der App, mit der App einfache Aufgaben zu erfüllen?
- Effizienz – Wenn der Umgang mit der App grundsätzlich klar ist, wie schnell kann der Anwender bestimmte Aufgaben damit erfüllen?
- Einprägsamkeit – Wenn der Nutzer nach einer gewissen Zeit ohne Nutzung die App erstmals wieder nutzt, wie schnell ist er wieder mit der App und ihren Funktionen vertraut?
- Fehler – Wie viele Fehler macht ein Nutzer, wie schwer sind diese Fehler, und wie leicht kann er eine Lösung zur Behebung des Problems finden?
- Zufriedenheit – Wie zufriedenstellend ist die Nutzung der App?

Bereits mit dieser Auflistung wird klar, dass viele Entwickler die Komplexität der Benutzerfreundlichkeit ihrer Smart Mobile Apps nicht in all ihren Facetten hinreichend beachten. Dabei lässt sich die Benutzerfreundlichkeit gerade durch die Auswertung der Situation, in der eine Smart Mobile App gerade genutzt wird, positiv beeinflussen. Einerseits kann der Nutzungskontext zur Steigerung der Effizienz, wie in oben genanntem Beispiel der automatischen Suche der Start-Haltestelle in Abhängigkeit vom Ort der Nutzung dargestellt, als intelligenter Informationsfilter verwendet werden. Andererseits lässt sich die Nutzung von Kontextinformationen auch zur Fehlervermeidung in Smart Mobile Apps verwenden. Beispielsweise sollte eine Apothekenfinder-App bei Nutzung um 23:00 Uhr nur Apotheken mit Notdienstbereitschaft anzeigen, auch wenn es sich nicht um die nächstgelegene Apotheke handelt. Es ist offensichtlich, dass andernfalls das Kriterium der Zufriedenheit mit der App unerfüllt ist, wenn der Nutzer die ersten beiden von der App empfohlenen Apotheken anfahren muss nur um festzustellen, dass diese bereits geschlossen sind, und erst die dritte der Liste aktuell Notdienst und daher geöffnet hat.

Die Nutzung von vielen „intelligenten“ Funktionen und Kontextinformationen in der App hat aber auch eine Kehrseite hinsichtlich der Nachvollziehbarkeit. Insbesondere wenn die gleiche App in unterschiedlichen Kontexten verschiedene Ergebnisse liefert oder Aktionen auslöst, ist dies weder der Erlernbarkeit noch der Einprägsamkeit im Sinne der Benutzerfreundlichkeit zuträglich. Es kann sogar dazu führen, dass der Nutzer das Vertrauen in den deterministischen Ablauf der Anwendung verliert [10]. Es sind daher auf jeden Fall Mechanismen vorzusehen, die es dem Nutzer einer Smart Mobile App auf Wunsch erlauben nachzuvollziehen, welche internen Schlussfolgerungen die App gezogen hat, die zu einem bestimmten Ergebnis in der Ausführung geführt haben. Ein Beispiel hierfür wird in [11] aufgezeigt.

Ähnlich negativ hinsichtlich der Benutzerfreundlichkeit wirkt sich die falsche Verwendung der auf mobilen Endgeräten möglicherweise vorhandenen Sensoren aus. Das oben genannte Beispiel zum Pairing von Telefon und Headset zeigt, wie Beschleunigungssensoren sinnvoll verwendet und zu einer Steigerung der Benutzerfreundlichkeit führen können. Die gleichen Beschleunigungssensoren können auf einem iPhone oder iPod auch dazu verwendet werden, durch Schütteln am Gerät

zwischen Musikstücken zu wechseln, die gerade abgespielt werden. Diese Funktion ist jedoch hinsichtlich der Benutzerfreundlichkeit kontraproduktiv, wenn das Gerät auf einem Fahrrad oder beim Jogging verwendet wird. Um die gewünschte hohe Benutzerfreundlichkeit zu erzielen, ist es unabdingbar, den Endanwender und seinen spezifischen Anwendungskontext zu berücksichtigen.

Auch der besondere Entwicklungszyklus von Smart Mobile Apps stellt eine Herausforderung an die Entwickler und die Benutzerfreundlichkeit ihrer Smart Mobile Apps dar. Schließlich werden Apps üblicherweise nicht auf den Endgeräten selbst, sondern auf PC- oder MAC-basierten Entwicklungsumgebungen geschrieben (*cross platform development*). Für die Entwicklung von Android Apps werden zum Beispiel entsprechende Emulatoren und Fehlersuchewerkzeuge bereitgestellt, die sich einfach in bestehende Entwicklungsumgebungen wie etwa Eclipse integrieren lassen. Apple Inc. stellt für Entwickler von iPhone Apps das umfangreiche Entwicklungspaket Xcode bereit, welches allerdings nur auf Apple-Computern lauffähig ist. Mit Hilfe dieser Entwicklungswerkzeuge können und sollten die Apps getestet werden. Dazu gehört auch die Verwendung von Profilern, mit denen sich versteckte Speicher- oder andere Laufzeitprobleme identifizieren lassen. Emulator-basierte Tests decken aber nur einen Teil der Vielfalt der Endgeräte ab und können die Fehlersuche auf der wirklichen Hardware nicht ersetzen. Es ist daher unbedingt erforderlich, eine ausreichend große Anzahl von Endgeräten mit ihren unterschiedlichen Betriebssystemen und -versionen, den unterschiedlichen Sensoren, Laufzeitverhalten etc. heranzuziehen und die Apps auf diesen Endgeräten umfangreichen Tests zu unterziehen.

## 27.5 Zusammenfassung und Ausblick

In diesem Beitrag wurden verschiedene Elemente aus dem Spannungsfeld zwischen Programmierstil, Performanz und Benutzerfreundlichkeit für Smart Mobile Apps aufgezeigt. Die dabei herausgearbeiteten Fakten sind aber nicht als abschließend zu betrachten. Vielmehr soll der Beitrag anregen, sich bei der Entwicklung von Smart Mobile Apps stets dieses Spannungsfeld vor Augen zu halten. Ziel sollte es sein, Apps zu entwickeln, die ein Optimum an Benutzerfreundlichkeit und Performanz auf den vorgesehenen Zielgeräten erreichen. Wenn die Programmierung dann noch stilistisch sauber ist, kann auch der Wartung der Software einigermaßen gelassen entgegengesehen werden.

## Literaturverzeichnis

1. App Store Metrics. <http://148apps.biz/app-store-metrics/?mpage=appcount>, 19. Juni 2011
2. Wendlandt K, Robertson P, Angermann M, Strang T, Kammann J, Wasel C, Dorsch T (2004) Experiences from Ramping Up an Environment for Mobile Information Access. Mobile and Ubiquitous Information Access (MUIA) workshop at MobileHCI04, Glasgow, Scotland

3. Mobile Information Device Profile for Java 2 Micro Edition. <http://jcp.org/aboutJava/communityprocess/mrel/jsr118/index.html>, 19. Juni 2011
4. Mayrhofer R, Gellersen H (2007) Shake well before use: two implementations for implicit context authentication. UbiComp 2007, Innsbruck, Austria
5. Die Sleep Cycle App für das iPhone im Test. <http://www.wuestenigel.com/2010/01/13/sleep-cycle-app-fuer-das-iphone-ein-erfahrungsbericht/>, 19. Juni 2011
6. Moviele HCI Conference Series. <http://www.mobilehci2011.org/>, 19. Juni 2011
7. International Symposium series on Location- and Context-Awareness (LoCA). <http://www.context-aware.org>, 19. Juni 2011
8. Martin RC (2009) Clean Code: A Handbook of Agile Software Craftsmanship. Person Education, Boston
9. Nielsen J (2005) Usability 101: Introduction to Usability. <http://www.useit.com/alertbox/20030825.html>, 19. Juni 2011
10. Vermeulen J (2010) Improving Intelligibility and Control in Ubicomp. Adjunct Proceedings, UbiComp 2010, Copenhagen, Denmark
11. Assad M, Carmichael DJ, Kay J, Kummerfeld B (2007) PersonisAD: Distributed, Active, Scrutable Model Framework for Context-Aware Services. Pervasive 2007, Toronto, Ontario, Canada

---

# Kapitel 28

## Eine Dienstplattform zur Entwicklung mobiler sozialer Anwendungen auf Basis von XMPP

Robert Lübke, Daniel Schuster, Thomas Springer, Nikolas Jansen  
und Alexander Schill

**Zusammenfassung** Die Entwicklung mobiler sozialer Apps mit direkter Interaktion zwischen Teilnehmern ist bislang sehr aufwändig und wird von den bestehenden mobilen Plattformen nur unzureichend unterstützt. Die Mobilis-Plattform stellt diese fehlenden kollaborativen Funktionen als wiederverwendbare Dienste auf Basis des eXtensible Messaging and Presence Protocol (XMPP) zur Verfügung. Damit wird die Erstellung mobiler sozialer Apps für den privaten oder geschäftlichen Bereich deutlich vereinfacht. Das Kapitel beschreibt die Architektur und die einzelnen Dienste der Plattform und zeigt anhand implementierter Beispiel-Anwendungen den Nutzwert der Mobilis-Dienstumgebung für den App-Entwickler.

---

Robert Lübke  
Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: robert.luebke@tu-dresden.de

Daniel Schuster  
Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: daniel.schuster@tu-dresden.de

Thomas Springer  
Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: thomas.springer@tu-dresden.de

Nikolas Jansen  
Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: nikojansen@gmail.com

Alexander Schill  
Technische Universität Dresden, Fakultät Informatik, Professur Rechnernetze,  
E-mail: alexander.schill@tu-dresden.de

## 28.1 Einleitung

Mit der starken Verbreitung sozialer Netzwerke werden auch Mobile Apps zunehmend „sozial“ und ermöglichen direkte Interaktion zwischen den Personen, die die jeweilige App benutzen. Das Spektrum reicht hier von der Anbindung mobiler Apps an soziale Spiele-Plattformen wie Apple Game Center<sup>1</sup> oder OpenFeint<sup>2</sup>, über mobile soziale Netzwerk-Apps wie Facebook for iPhone<sup>3</sup> oder WhatsApp Messenger<sup>4</sup>, bis hin zu strukturierter, ortsbezogener Echtzeitkollaboration wie spontaner Vermittlung von Mitfahrangelegenheiten (OpenRide<sup>5</sup>). All diesen Anwendungen ist gemeinsam, dass sie auf den Consumer-Markt ausgerichtet sind und dort schnelle (virale) Verbreitung finden. Die Konzepte lassen sich jedoch häufig auch auf Business Apps übertragen, wie das Beispiel von Unsocial<sup>6</sup> zeigt. Hier wird das bekannte Buddy-Finder-Konzept, das interessante Bekanntschaften mit ähnlichen Interessen in der Umgebung anhand ihrer hinterlegten Profile und Ortsinformationen findet, auf das Finden von vielversprechenden Business-Kontakten (z. B. auf Messen oder Konferenzen) angewendet. Ähnlich bergen auch andere mobile soziale Anwendungen großes Potential im Business-Umfeld.

Ein wesentliches Problem bei der Entwicklung solcher Anwendungen ist die fehlende Unterstützung der wichtigsten mobilen Plattformen (Android, iOS, BlackBerry, ...) für direkte Interaktion zwischen Nutzern. So müssen wiederkehrende Funktionen wie Presence und Awareness, Instant Messaging, Austausch und Matching von Ortsinformationen und anderem Kontext oder synchronisierte Manipulation von Zustandsobjekten jeweils neu implementiert werden. Dieser Prozess ist aufgrund von praktischen Schwierigkeiten direkter Kommunikation im mobilen Umfeld sowie der Komplexität der Konsistenzalgorithmen für Echtzeitkollaboration sehr aufwändig und fehleranfällig. Eine Wiederverwendung implementierter und getester Komponenten kann deshalb den Entwicklungsaufwand erheblich reduzieren.

Wir stellen in diesem Beitrag mit dem Mobilis-Projekt eine Dienstplattform vor, die sich als Ergänzung zu existierenden mobilen Plattformen versteht und diese um wichtige Aspekte zur Entwicklung mobiler sozialer Apps erweitert. Die Grundidee von Mobilis ist die Nutzung des eXtensible Messaging and Presence Protocol (XMPP) als Infrastruktur für mobile interaktive Anwendungen. Dabei setzen wir auf bestehende XMPP-Erweiterungen (XEPs) der sehr aktiven XMPP-Community auf und ergänzen diese um einige zusätzliche Mechanismen zur Realisierung einer mobilen sozialen Dienstumgebung.

Durch die offene XML-basierte Kommunikation in XMPP ist die Einbindung beliebiger Clients möglich, wobei wir uns bislang auf die Umsetzung von Android-Clients sowie webbasierten Zugang zur Plattform beschränkt haben. Es lassen sich

<sup>1</sup> <http://www.apple.com/de/game-center/>

<sup>2</sup> <http://openfeint.com/>

<sup>3</sup> <http://www.facebook.com/iphone>

<sup>4</sup> <http://www.whatsapp.com/>

<sup>5</sup> <http://www.open-ride.com/>

<sup>6</sup> <http://unsocial.mobi/>

jedoch mit vertretbarem Aufwand zum Beispiel auch native iOS-Clients in die Plattform einbinden. Eine solche offene Plattform für mobile Interaktion existiert bislang weder in der Forschung noch im kommerziellen Bereich.

Im Mittelpunkt dieses Beitrags steht die Beschreibung der XMPP-basierten Architektur der Mobilis-Plattform. Die Plattform steht als Open-Source-Lösung frei zur Verfügung und kann so im Rahmen anderer Projekte (insbesondere auch zur Umsetzung von sozialen Business Apps) genutzt werden. Im Evaluierungsteil werden einige Beispieldaten beschrieben, die auf Basis der Plattform realisiert werden und so einen Überblick über die Möglichkeiten der Plattform geben. Im Folgenden werden zunächst bestehende Plattformen analysiert und anschließend die Anforderungen an eine mobile soziale Plattform abgeleitet.

## 28.2 Mobile Social Software, Anforderungen und Frameworks

Um die Anforderungen der Entwickler mobiler sozialer Apps richtig zu verstehen, ist es wichtig, zunächst einen Überblick über diesen sich sehr schnell entwickelnden Bereich zu geben. Dabei soll gleichzeitig eine Eingrenzung des Fokus der Mobilis-Plattform erfolgen.

### 28.2.1 Mobile Social Software

Der Begriff *Mobile Social Software* bezeichnet eine Klasse von Anwendungen für mobile Endgeräte mit dem Hauptziel, die soziale Interaktion zwischen den Nutzern zu ermöglichen und zu unterstützen. Dabei stehen Kommunikation, Koordination und Kooperation zwischen den Nutzern im Vordergrund.

Zur Systematisierung der Fülle von existierenden Applikationen und Systemen wurde bereits eine Reihe von Klassifikationen und Taxonomien entwickelt.

In [1] werden drei Modelle für Social Software vorgestellt: *intimate*, *crowd* und *hybrid*. Intimate Social Networks sind solche Systeme, deren Nützlichkeit sich erst zeigt, wenn andere persönliche Kontakte aus der realen Welt ebenfalls das System nutzen (z. B. Facebook). Andererseits gibt es Crowd-Systeme, bei denen die soziale Beziehung der Teilnehmer untereinander keine Rolle spielt (z. B. eBay). Hybrid Networks verbinden Aspekte von beiden vorher genannten Modellen (z. B. Flickr).

Die Mobilis-Plattform richtet sich vor allem an Entwickler von Intimate Social Software, speziell sollen Echtzeit-Interaktionen zwischen sich gegenseitig bekannten Personen unterstützt werden (siehe dazu die Beispieldaten MobilisMapDraw, MobilisBuddy sowie die Spiele MobilisXHunt und MobilisLocPairs). Die Unterstützung von hybrider Social Software ist damit ebenfalls möglich (siehe MobilisMedia und MobilisGroups).

In [2] wird Mobile Social Software danach klassifiziert, inwieweit der soziale Kontext ausgeprägt ist. Die Autoren betrachten vier Dimensionen des sozialen

Kontextes: *spatial*, *temporal*, *inference* und *people* (STIP). Die räumliche Dimension (*spatial*) gibt Auskunft darüber, wie relevant die geografische Entfernung der Nutzer untereinander ist. Die Dimension *temporal* bestimmt den zeitlichen Aspekt bei gemeinsamer Interaktion der Nutzer. Weiterhin wird unterschieden, auf welche Art die Systeme den sozialen Kontext des Nutzers ableiten (*inference*). Letztlich wird die Art der Zielgruppe untersucht, mit der der Nutzer im sozialen Netzwerk interagiert (*people*). Dabei kann es sich um einzelne Personen, aber auch um Gruppen von Personen oder einen anonymen Nutzerkreis handeln. Die einzelnen Systeme werden anhand der Granularität der entsprechenden Dimensionen eingeordnet und letztlich entsteht für jedes System ein charakteristischer STIP-Vektor.

Nach dieser Einteilung zielt Mobilis vor allem auf zeitlich unmittelbare (T1) Interaktion im direkten (S1) oder weiteren (S2) geographischen Umfeld einer Person. Diese findet in erster Linie zwischen Personen (P1) oder Gruppen (P2) statt.

Schließlich kann man mobile soziale Anwendungen auch anhand ihrer Hauptfunktionalität klassifizieren, auch wenn diese Art der Klassifikation aufgrund der sehr schnellen Entwicklung Änderungen unterworfen ist. Die aus Sicht der Autoren momentan wichtigsten Kategorien sind:

- K1 **Messaging / Media Sharing** – Austausch von Textnachrichten und multimedialen Inhalten (WhatsApp, Twitter, Flickr).
- K2 **Social Network Service (SNS)** – „klassisches“ soziales Netz mit Pflege von Profilen, Freundeslisten, Gruppen, Status-Updates, etc. (Facebook, Google+).
- K3 **Proximity Matching** – Benachrichtigung, wenn sich ein Freund oder ein Nutzer mit ähnlichen Interessen in der Nähe befindet, z. B. VENETA [3], Whoz-That [4], und PeopleTones [5]. Darunter fallen außerdem Apps zur ortsbasierten Gruppenbildung wie Urbiflock [6] und Socialaware [7].
- K4 **Shared Editing** – gemeinsames Bearbeiten von Texten, Zeichnungen oder anderen strukturierten Objekten (Google Wave & Docs, Share Board).
- K5 **Social Tagging / Crowdsourcing** – z. B. Google Buzz.
- K6 **Mobile Social Games** – z. B. Mister X Mobile<sup>7</sup>, auch Social Networks mit Spielcharakter wie Foursquare und Gowalla.

Betrachtet man eine Auswahl von Anwendungen aus diesen Kategorien, so stellt man fest, dass viele davon ähnliche oder sogar gleiche Features aufweisen, z. B. Kommunikation, Ortsbezug, Media Sharing, Gruppenbildung und Kollaboration zwischen den Nutzern. Wir werden im Folgenden daraus Anforderungen für eine Plattform mit wiederverwendbaren Diensten ableiten.

## 28.2.2 Anforderungen

Aus der Klasse K1 ergibt sich, dass die Plattform zunächst eine einfache Möglichkeit für **direkte Kommunikation** (Textnachrichten und Protokollnachrichten) zwi-

---

<sup>7</sup> <http://www.gamesload.de/misterxmobile/>

schen Teilnehmern bieten muss. Eine damit verbundene **Presence**-Funktionalität erleichtert die Kontaktaufnahme. Darüber hinaus sollte der Austausch von Medienobjekten (**Media Sharing**) unterstützt werden.

Dagegen erscheint es wenig sinnvoll, einen Social Network Service (K2) in der Plattform anzubieten, da solche Dienste bereits existieren und häufig über geeignete Schnittstellen zur Einbindung in mobile Apps verfügen. Diese **Einbindung von existierenden SNS** soll die Mobilis-Plattform vereinfachen.

Für die derzeit sehr populäre Funktionalität des Proximity Matching (K3) spielt der Austausch von Ortsinformationen bzw. Kontextinformationen (**Context Management**) eine große Rolle. Die Plattform muss den Kontext der Nutzer ermitteln, verwalten und an autorisierte Stellen weitergeben können. Zum Kontext zählen in erster Linie Informationen zum Aufenthaltsort, aber auch z. B. Freundesangaben oder Profilinformationen des Nutzers. Weitere denkbare Dienste in diesem Bereich sind die eigentliche **Proximity Detection** sowie Mechanismen zur **ortsbezogenen Gruppenbildung**.

Ein generischer Dienst für das gemeinsame Editieren von XML-strukturierten Informationen (K4 – **Shared Editing**) beinhaltet ein großes Potential, da sich damit sowohl gemeinsames Editieren von Text, gemeinsames Zeichnen als auch andere gemeinsame Manipulation von XML-Objekten (z. B. gemeinsamer Anwendungszustand in Spielen) realisieren lassen.

Für die Anwendungsklassen K5 (Social Tagging/Crowdsourcing) und K6 (Mobile Social Games) werden keine expliziten Dienste bereitgestellt. Hier gibt es bereits entsprechende Plattformen (z. B. Game Center, OpenFeint bei Spielen), die genutzt werden können. Trotzdem profitieren auch Mobile Social Games von der Nutzung der Mobilis-Plattform, wie im Evaluierungsteil noch gezeigt wird.

Darüber hinaus werden zur Entwicklung kollaborativer Apps häufig **Server-Dienste** benötigt, die bei komplexeren Anwendungen wichtige Koordinationsaufgaben übernehmen. Die Plattform muss deshalb die Möglichkeit bieten, solche Dienste leicht hinzuzufügen.

All die bisher genannten Funktionen sollen mithilfe des **Service-Paradigmas** realisiert werden. Wie in einer serviceorientierten Architektur (SOA) sollen die Funktionalitäten gekapselt und über wohldefinierte Schnittstellen angeboten werden. Die Nutzung der angebotenen Dienste soll transparent zur zugrunde liegenden Implementation erfolgen. Es wird zusätzlich ein Mechanismus zum Auffinden der Dienste benötigt. Die Umsetzung der Funktionalität innerhalb einer Dienstumgebung führt zu einer besseren Wiederverwendbarkeit und Interoperabilität.

### 28.2.3 Existierende Frameworks

Es gibt im Forschungsbereich bereits einige Frameworks und Middleware Layer wie z. B. SAMOA [8], MobiSoc [9] und MobiClique [10], die die Entwicklung von neuen sozialen Anwendungen erleichtern. Diese Systeme verfolgen allesamt das Ziel, die Nutzergemeinde in einzelne soziale Netzwerke einzuteilen. Dafür setzen die drei

vorgestellten Systeme Nutzer- und Ortsprofile ein, die miteinander abgeglichen werden, um für die Nutzer interessante Personen und Orte ausfindig zu machen. Diese Systeme decken also überwiegend Grundfunktionen für soziale Netzwerke (K2) ab und die Entwickler können darauf aufbauend zusätzliche Anwendungslogik umsetzen. Weiterführende Funktionen zur Interaktion und Kollaboration im mobilen Umfeld werden von den existierenden Frameworks aber nicht oder nur in geringem Umfang unterstützt. Gerade die Wiederverwendung von Funktionen für die Kollaboration kann aber den Entwicklungsaufwand für mobile soziale Anwendungen erheblich reduzieren. Im Folgenden wird gezeigt, wie die Mobilis-Plattform diese Lücke füllen kann.

## 28.3 Die Mobilis-Plattform

Dieser Abschnitt gibt einen umfassenden Überblick über die Konzepte der Mobilis-Plattform. Zuerst wird kurz XMPP als Technologie vorgestellt, die für die Bereitstellung der Basiskommunikation sowie der Nutzung der Plattform-Dienste eingesetzt wird. Nach der Beschreibung der allgemeinen Architektur der Plattform sollen die Hauptkomponenten näher beleuchtet werden.

### 28.3.1 XMPP als Technologie für die Dienstumgebung

Das *Extensible Messaging and Presence Protocol* (XMPP) ist eine Protokollsammlung auf XML-Basis für den Austausch von Nachrichten und Verfügbarkeitsinformationen. Jeder XMPP-Nutzer unterhält eine permanente TCP-Verbindung zu einem zugeordneten XMPP-Server und kann über seine XMPP-Adresse (JID) identifiziert und von anderen Nutzern kontaktiert werden. Ähnlich wie bei E-Mail können Nutzer, die bei unterschiedlichen XMPP-Servern angemeldet sind, durch DNS-Unterstützung weltweit miteinander kommunizieren.

Neben Presence-Nachrichten und den Instant Messages unterstützt XMPP mit den Info/Query-Nachrichten (IQ) aber auch einen leistungsfähigen RPC-Mechanismus, mit dem sich höherwertige Dienste realisieren lassen. Es können get- und set-Requests an andere Teilnehmer geschickt werden, die dann über entsprechende Responses darauf antworten. Mittels Namespaces lassen sich beliebige XML-strukturierte Protokollnachrichten transportieren. Dieser Mechanismus wird bereits durch mehr als 200 XMPP Extension Protocols (XEP), die von der XMPP Standards Foundation angeboten werden, erfolgreich genutzt. So wird beispielsweise Google Talk über mehrere XEPs realisiert. Google Wave nutzt ebenfalls IQs zur Synchronisation von Änderungen einer Wave zwischen Servern.

Es liegt deshalb nahe, XMPP als Basistechnologie für die Mobilis-Plattform zu nutzen. Neben den genannten Möglichkeiten bietet XMPP bereits Authentisierung und sichere Kommunikation mit SASL und TLS sowie zuverlässige Zustellung

und Reihenfolgegarantie der Nachrichten durch die Nutzung von TCP. Weiterhin ist XMPP durch die Kodierung der Daten im XML-Format plattformunabhängig. Dem damit verbundenen Kommunikations-Overhead kann mit Ansätzen wie XML Stream Compression entgegengewirkt werden.

### 28.3.2 Architektur

Die allgemeine Architektur der Mobilis-Plattform zeigt Abb. 28.1. Der *Mobilis Server* bietet den Client-Anwendungen verschiedene Dienste an. Die Kommunikation basiert auf der XMPP-Bibliothek *Smack*<sup>8</sup>. Als XMPP-Server wird *Openfire*<sup>9</sup> eingesetzt, aber es können auch andere XMPP-Server verwendet werden.

Die Server-Dienste (*MobilisServices*) verfügen über eine eigene XMPP-Adresse, über die sie mittels XMPP-Nachrichten angesprochen werden können.

Auf Client-Seite wird die XMPP-Kommunikation über die MXA-Komponente (Mobilis XMPP on Android) abgewickelt, die eine angepasste Version der Smack-Bibliothek kapselt. Daneben bietet MXA noch weitere XMPP Services an, die jeweils eine XMPP-Erweiterung implementieren. Andere Anwendungen greifen über Interprozesskommunikation auf MXA zu, der als Android-Dienst für alle Anwendungen auf einem Gerät verfügbar ist. Durch die Verwendung von unterschiedlichen Namespaces können empfangene Nachrichten auch bei paralleler Nutzung mehrerer Mobilis-Apps zuverlässig zugeordnet werden. Die Komponente *Mobilis*

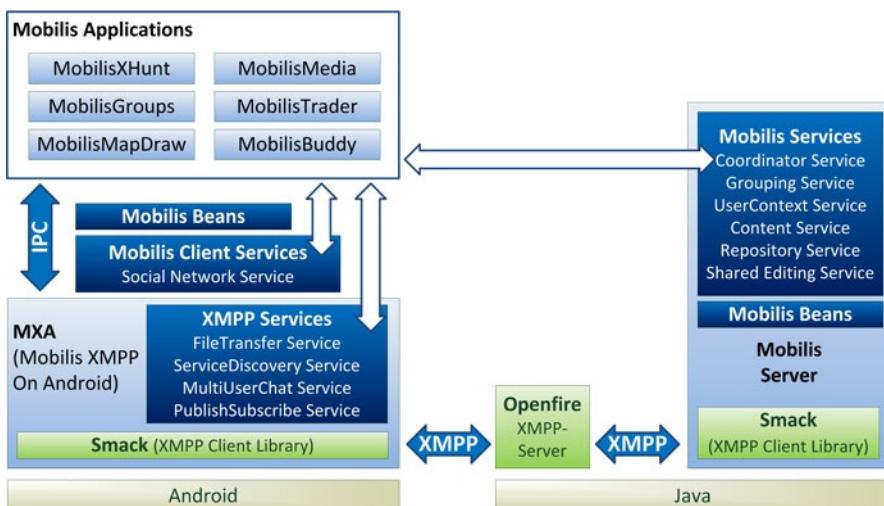


Abb. 28.1 Allgemeine Architektur der Mobilis-Plattform

<sup>8</sup> <http://www.igniterealtime.org/projects/smack/>

<sup>9</sup> <http://www.igniterealtime.org/projects/openfire/>

*Beans* ist eine Sammlung von Strukturen, mithilfe derer die verschiedenen XMPP-Nachrichten erstellt werden können, die zwischen Client und Server ausgetauscht werden. Hier sind also die einzelnen Bestandteile der Protokolle zu finden, in denen die Nutzung der Plattformdienste festgelegt ist. Daher nutzen Entwickler die *Mobilis Beans*, um konkrete Anfragen an die Dienste durchzuführen. Die Nutzung in Client und Server stellt sicher, dass beide Seiten die gleiche Nachrichtensyntax verwenden. In der Bibliothek *Mobilis Client Services* finden Entwickler von Android-Applikationen Dienste, die sie leicht in ihre Anwendungen integrieren können.

### 28.3.3 *Client Services*

Die *Mobilis Client Services* bieten Anwendungsentwicklern typische Funktionalitäten mobiler sozialer Anwendungen an. Diese Dienste sind in mehreren Anwendungen einsetzbar und leicht integrierbar. Die Integration externer sozialer Netzwerke ist dabei die Hauptaufgabe der *Mobilis Client Services*.

Umgesehen wird diese Funktionalität innerhalb des *Social Network Service*. Hier werden die APIs verschiedener sozialer Netzwerke gekapselt und dem Entwickler an einer zentralen Stelle angeboten. Der *Social Network Service* ist als Android Service umgesetzt, der seine Funktionalität über eine AIDL-Schnittstelle (Android Interface Definition Language) zur Verfügung stellt. Er verwaltet mehrere Social-Network-Manager-Komponenten, die die jeweiligen API-Funktionen der unterstützten sozialen Netzwerke kapseln.

Zu den unterstützten Systemen gehören momentan Foursquare, Gowalla und Qype, da diese in Beispieldiensten bereits genutzt wurden. Über den generischen Mechanismus ist es aber möglich, den *Social Network Service* um andere Netzwerke wie Facebook zu erweitern.

### 28.3.4 *XMPP Services*

Neben der eigentlichen XMPP-Kommunikation gehören zur MXA-Anwendung auch speziellere Dienste, die jeweils eine bestimmte XMPP-Erweiterung umsetzen. Eine dieser Erweiterungen ist der in [11] spezifizierte *Multi-User Chat* (MUC). Mithilfe des MUC-Dienstes kann man einem solchen Mehrbenutzerchat beitreten und ihn verlassen, auf Einladungen reagieren, Informationen zum Chat-Raum abrufen und natürlich auch Nachrichten an alle MUC-Teilnehmer zusenden. Weiterhin unterstützt der MUC-Dienst auch diverse administrative Funktionen. Der *FileTransfer Service* ermöglicht den Austausch von Dateien innerhalb des XMPP-Netzwerks, basierend auf der Erweiterung SI FileTransfer [12]. Nutzer können Dateien senden und eingehende Übertragungen verwalten. Der *ServiceDiscovery Service* setzt die Erweiterung XEP-0030 [13] um. Dieser Dienst ermöglicht es, Anfragen an XMPP-Entitäten zu senden, um in Erfahrung zu bringen, welche Protokolle, Erweiterun-

gen und Features in welcher Version unterstützt werden. Innerhalb der Mobilis-Plattform wird dieser Dienst im Zusammenhang mit dem *Coordinator Service* (siehe Abschn. 28.3.5) benutzt, damit die Client-Anwendungen alle auf dem Mobilis Server aktiven Dienste auffinden können. Der *PublishSubscribe Service* setzt Teile der XMPP-Erweiterung XEP-0060 um. Diese spezifiziert die Grundfunktionalität eines generischen Publish-Subscribe-Dienstes [14]. Der Service unterstützt sowohl das Abonnieren von Informationsupdates als auch deren Kündigung.

### 28.3.5 Mobilis Services

Die Mobilis Services sind die serverseitigen Dienste des Frameworks und stellen die Hauptkomponente der Dienstumgebung dar. Abbildung 28.2 veranschaulicht die verschiedenen Schichten, in die die Dienste eingeteilt werden können.

Client-Applikationen haben die Möglichkeit, über den *Coordinator Service* Informationen über alle aktiven und aktivierbaren Dienste des Mobilis Server zu erlangen und benötigte Dienste zu starten. Die generischen und wiederverwendbaren Dienste sind die Grundlage des Mobilis Server. Dazu gehören z. B. Dienste zur Verwaltung und Speicherung von Mediendaten (*Media Repository* und *Media Content Service*), zur Speicherung von Nutzerkontextinformationen (*User Context Service*), zur ortsbasierten Gruppenbildung (*Grouping Service*) und zum gleichzeitigen, kolaborativen Bearbeiten von XML-Dokumenten (*Shared Editing Service*). Ihre Daten speichern die Dienste persistent in einer Datenbank.

In einer darüberliegenden Schicht sind anwendungsspezifische Dienste (engl. App-specific Services) zu finden, die allerdings auch Zugriff auf die generischen

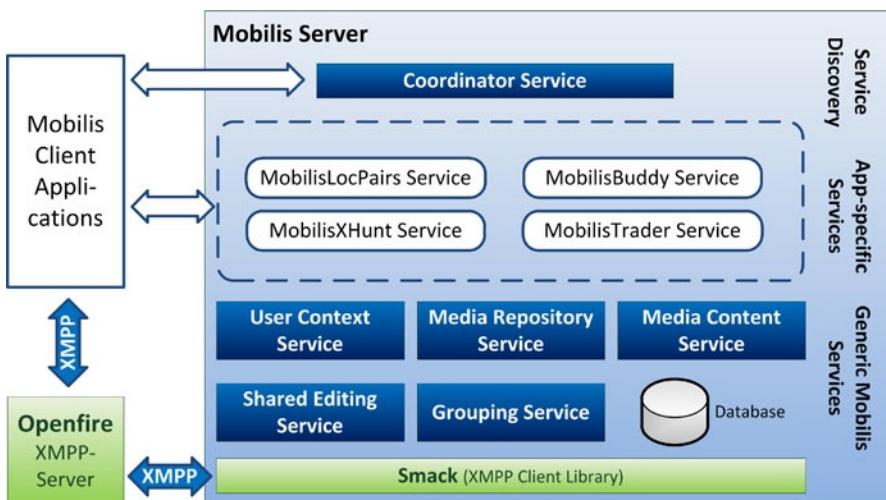


Abb. 28.2 Dienste auf dem Mobilis Server

Dienste haben. Sie vermitteln zwischen den *Mobilis Client Applications* und den *Generic Mobilis Services*. Die Client-Applikationen können entweder direkt die generischen Mobilis Services verwenden oder dafür einen eigenen App-specific Service benutzen, in dem zusätzliche Anwendungslogik zu finden ist.

Beispiele für anwendungsspezifische Dienste sind *XHunt* und *LocPairs Service*. Dabei handelt es sich um die Serverkomponenten von zwei ortsbasierten Spielen. Der *MobilisBuddy Service* setzt Proximity Detection um.

Alle Dienste auf dem Mobilis Server verfügen über eine eigene XMPP-Adresse, über die sie durch Client-Applikationen ansprechbar sind. Daher besitzt jeder Service auch eine eigene Verbindung zum XMPP-Server. Die beschriebene Aufsplitzung der Funktionalitäten auf mehrere Dienste ermöglicht auch den Einsatz in einem verteilten Szenario. Jeder Service kann so ohne weitere Anpassungen auf einem eigenen Rechner laufen. Um eine Auflistung aller unterstützten Dienste eines Mobilis Server zu erhalten, muss ein entsprechender Request an den Coordinator Service gesendet werden:

```
<iq from="client@xmpp/MXA" to='mobilis@xmpp / Coordinator'
    id='mobilis-disco-001' type='get'>
  <serviceDiscovery xmlns='...# services / CoordinatorService' />
</iq>
```

Die Antwort auf diese Anfrage könnte wie folgt aussehen:

```
<iq from='mobilis@xmpp / Coordinator' to='client@xmpp / MXA'
    id='mobilis-disco-001' type='result'>
  <serviceDiscovery xmlns='...# services / CoordinatorService'>
    <mobilisService namespace='...# services / UserContextService'
        version='1.0' mode='single'
        jid='mobilis@xmpp / Context'/>
    <mobilisService namespace='...# services / RepositoryService'
        version='1.0' mode='single'
        jid='mobilis@xmpp / Repository'/>
    <mobilisService namespace='...# services / CoordinatorService'
        version='1.0' mode='single'
        jid='mobilis@xmpp / Coordinator'/>
    <mobilisService namespace='...# services / ContentService'
        version='1.0' mode='single'
        jid='mobilis@xmpp / Content'/>
    <mobilisService namespace='...# services / CollabEditingService'
        version='1.0' mode='single'
        jid='mobilis@xmpp / CollabEditing'/>
    <mobilisService namespace='...# services / GroupingService'
        version='1.1' mode='single'
        jid='mobilis@xmpp / Grouping'/>
    <mobilisService namespace='...# services / XHuntService'
        mode='multi' instances='3' />
  </serviceDiscovery >
</iq>
```

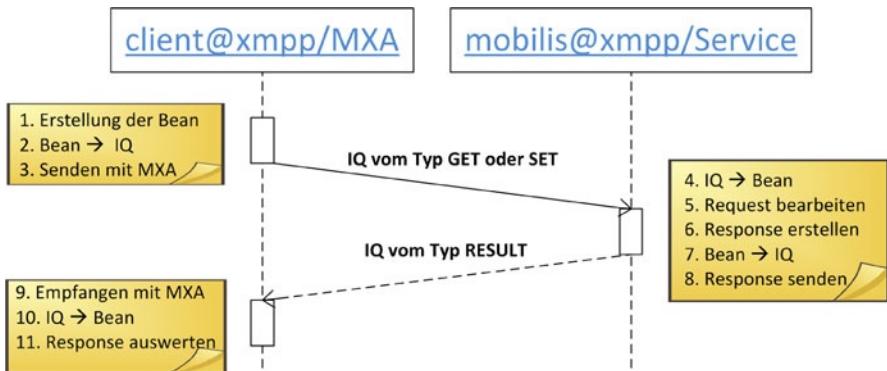


Abb. 28.3 Ablauf bei Nutzung eines einfachen Mobilis Service

Im Payload des Result IQ sind alle aktiven und aktivierbaren Dienste dieses Mobilis Server mit ihrem Namespace (obige Angaben gekürzt) und ihrer Versionsnummer zu finden. Außerdem wird im mode-Attribut angegeben, ob immer nur eine oder auch mehrere Instanzen des Dienstes existieren können. Die Client-Applikation kann den gewünschten Dienst nun über die jeweils angegebene JID ansprechen und nutzen. Dieses Service-Discovery-Verfahren unterstützt auch das Suchen nach bestimmten Diensten, indem einfach bereits im Request Namespace und Version des gesuchten Dienstes angegeben werden.

Nach erfolgtem Auffinden der Dienste kann die eigentliche Service-Nutzung durch den Austausch von IQ-Paketen erfolgen. Der detaillierte Ablauf bei einem einfachen Dienst, der nach dem Request/Response-Prinzip arbeitet, ist in Abb. 28.3 angegeben. Bei einigen Diensten sind die Abläufe auch komplexer und gleichen z. B. dem Publish/Subscribe-Prinzip.

### 28.3.6 Webbasierter Zugang

Neben nativen Client-Komponenten in Android können auch Webanwendungen über ein Web-Gateway auf die Mobilis-Plattform zugreifen. Damit können weitere mobile Plattformen wie iOS, Palm WebOS, Windows Phone, Symbian und BlackBerry für Mobilis erschlossen werden. Die stark gestiegene Leistungsfähigkeit von Javascript-Interprettern ermöglicht eindrucksvolle Benutzerschnittstellen im Webbrowser – selbst auf mobilen Endgeräten.

Die Wahl von XMPP als Kommunikationstechnologie für die Dienstumgebung bringt zwei grundsätzliche Probleme beim Entwurf eines webbasierten Zugangs mit sich: erstens sind hier die Unzulänglichkeiten des *Hypertext Transfer Protocol* (Abk. HTTP) für bidirektionale Echtzeitkommunikation zwischen Client und Server zu nennen und zweitens fehlen bisher APIs für native XMPP-Kommunikation im Webbrowser.

Für das erste Problem wird die WebSocket API des HTML5-Standards in Zukunft Abhilfe schaffen. Auf Grund der unzureichenden Unterstützung einiger Webbrowser und ungeklärten Sicherheitsbedenken ist es jedoch zu diesem Zeitpunkt nicht ratsam, einen Entwurf auf dieser Technologie zu stützen. Daher bleibt nur die Wiederverwendung von bereits bestehenden Technologien übrig, um die Semantik einer langlebigen bidirektionalen Socketverbindung (wie bei TCP) zwischen Webserver und Webbrower nachzubilden. Unter dem Namen *Bidirectional-streams Over Synchronous HTTP (BOSH)* wird in [15] eine Technologie veröffentlicht, die mit Hilfe einzelner HTTP-Request/Response-Zyklen und einem mit speziellen Attributen angereicherten <body>-Element die gewünschte bidirektionale Verbindung emuliert. Eine Komponente auf dem Webserver mit der Bezeichnung *Connection Manager* übernimmt die Auswertung der Attribute und somit die administrativen Aufgaben für das Auf- und Abbauen der langlebigen Socketverbindung.

Für das zweite Problem bietet das in [16] spezifizierte *XMPP Over BOSH* eine passende Lösung. XMPP-Nachrichten können als Kindelemente des <body>-Elements via BOSH übertragen werden. Im Webbrower kann ein Javascript BOSH Client die Informationen extrahieren und verarbeiten. Jegliche Nutzereingaben oder Kontextänderungen können über Javascript als XMPP-Nachricht verpackt und über BOSH mit der Mobilis-Service-Umgebung kommuniziert werden. Zusammen mit dem Long-Polling-Paradigma ist es so möglich, bidirektionale XMPP-Echtzeitkommunikation über HTTP zu realisieren.

Die vorgestellten Mobilis Services können also clientseitig auf Basis von *XMPP over BOSH* in Javascript implementiert werden und bieten Entwicklern von Webanwendungen für die Mobilis-Plattform ein komfortables Abstraktionsniveau. Details des Verbindungsauf- und -abbaus sowie der Erstellung von XMPP-Nachrichten werden dem Entwickler von Mobilis-Webanwendungen vorenthalten. Die Javascript-Bibliothek Mobilis Client Services ist über eine Plug-in-Schnittstelle jederzeit um weitere Services erweiterbar.

## 28.4 Evaluierung durch Mobilis Apps

Um die Funktionen der Mobilis-Plattform zu demonstrieren, wurden durch verschiedene (meist studentische) Entwickler-Teams Mobile Apps auf Basis von Mobilis umgesetzt, von denen in diesem Abschnitt einige vorgestellt werden sollen.

*MobilisMapDraw* demonstriert den serverseitigen Collaborative Editing Service, indem es kollaboratives Zeichnen ermöglicht. Die Hauptansicht der App zeigt einen Zeichenbereich, der optional mit einer Kartensicht überlagert werden kann. Der Nutzer kann nun einer Collaborative Drawing Session beitreten oder eine neue anlegen. In einer solchen Session können alle Teilnehmer gleichzeitig auf dem Shared Whiteboard oder der Karte zeichnen, wie es in Abb. 4 zu sehen ist. Collaborative Drawing kann in verschiedenen Szenarien Anwendung finden, z. B. in einer Tourismus-App, um Reiserouten zu diskutieren.

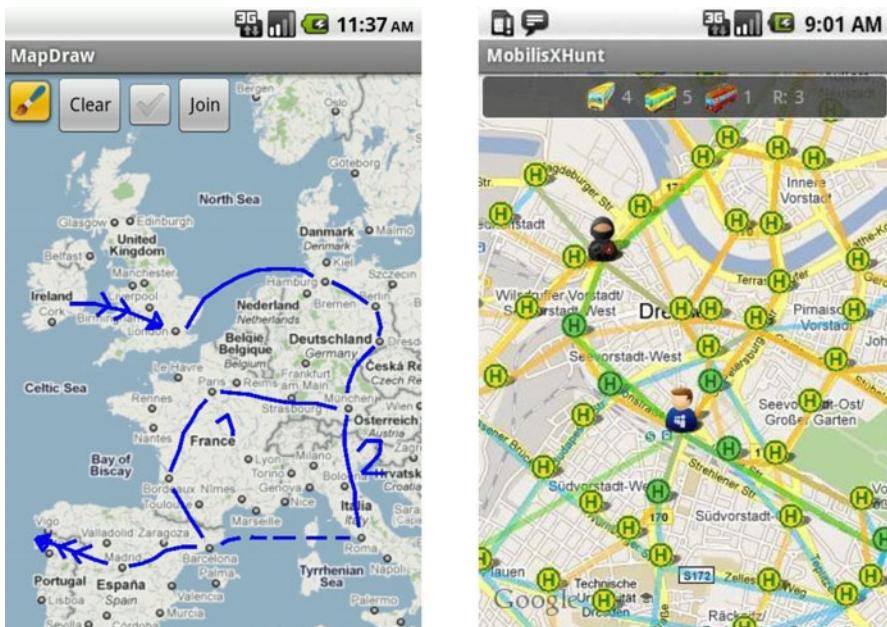


Abb. 28.4 MobilisMapDraw (links) und MobilisXHunt (rechts)

Mit *MobilisMedia* lassen sich multimediale Inhalte (insbesondere Bilder) in Nutzergruppen austauschen. Man kann damit georeferenzierte Fotos in Repositories speichern. Der Inhalt dieser Repositories kann dann durch andere Nutzer nach Autor, Aufnahmedatum und -ort durchsucht werden. Die Fotos werden in einer Kartendarstellung dargestellt. Zusätzlich ist es möglich, Bilder direkt und ohne Nutzung der Repositories an die eigenen Freunde zu verschicken.

Grundidee von *MobilisXHunt* ist die Umsetzung des bekannten Brettspiels Scotland Yard von Ravensburger in die Realität. Einer der Spieler ist der gesuchte Mister X, die restlichen Mitspieler sind Agenten, die Mister X suchen. Der Aufenthaltsort von Mister X wird nur in gewissen Abständen bekannt gegeben. In *MobilisXHunt* (siehe Abb. 28.4) gehen die mit Smartphones ausgestatteten Spieler in einer echten Stadt auf die Jagd nach Mister X und nutzen dabei echte Verkehrsmittel, um von einer Haltestelle zur nächsten zu gelangen.

*MobilisLocPairs* ist eine Adaption des bekannten Kinderspiels Memory. Es wird innerhalb von Gebäuden gespielt, in denen bestimmte Barcodes verteilt wurden. Hinter jedem Barcode versteckt sich ein Bild, das durch Scannen des Codes aufgedeckt werden kann. LocPairs ist wie das Originalspiel rundenbasiert, aber es wird in Teams zu je zwei Spielern gespielt. In jeder Runde deckt zuerst ein Spieler eine Karte auf, und der andere Spieler versucht dann im Gebäude den anderen dazu passenden Barcode zu finden. LocPairs hat den Zusatznutzen, dass beim Scannen jedes Barcodes eine Signalstärkemessung zu allen verfügbaren WLAN Access Points durchgeführt wird. Die Ergebnisse werden an einen Server gesendet und später dafür benutzt, die Indoorortung zu verbessern.

*MobilisGroups* ist eine App für ortsbasierte Gruppenbildung und wird in [17] ausführlich diskutiert. Am Beispiel dieser App soll im Folgenden gezeigt werden, wie verschiedene Plattformdienste in Kombination verwendet werden können, um kollaborative Mehrwertdienste umzusetzen. Die allgemeine Architektur, wie sie in Abb. 28.1 dargestellt ist, bleibt natürlich auch bei *MobilisGroups* erhalten. Serverseitig wird für die Verwaltung der ortsbasierten Gruppen der *Grouping Service* benutzt. Die aktuelle Position wird über den *User Context Service* verwaltet. Außerdem werden mittels des clientseitigen *Social Network Integration Service* die externen Systeme Foursquare und Qype eingebunden, damit die Nutzer ortsbasierte Gruppen schneller aus Vorlagen heraus erstellen können.

Weitere *Mobilis*-Anwendungen wie zum Beispiel *MobilisBuddy* sind auf der Webseite des *Mobilis*-Projektes<sup>10</sup> zu finden. Die Bandbreite der Anwendungen zeigt die umfassende Unterstützung der Plattform für Entwickler von mobilen Apps mit direkter Interaktion zwischen Teilnehmern.

## 28.5 Zusammenfassung, Ausblick und Diskussion

Der Hauptbeitrag dieser Arbeit ist die Vorstellung der *Mobilis*-Plattform, die wiederverwendbare, flexibel komponierbare Dienste zur Umsetzung mobiler sozialer Anwendungen bereitstellt. Die Wiederverwendbarkeit der Dienste in Kombination mit der Verwendung offener Standards wie XMPP ermöglicht eine effizientere Umsetzung mobiler sozialer Anwendungen als bei proprietärer Entwicklung. Durch die flexible Kombination verschiedener Dienste lässt sich die Entwicklungszeit und der Entwicklungsaufwand für mobile soziale Anwendungen wesentlich reduzieren, da häufig verwendete Funktionen nicht jedes Mal erneut konzipiert und implementiert werden müssen.

Die Vielfalt der mit *Mobilis* umgesetzten Anwendungsszenarien beweist, dass die von der Plattform angebotenen kollaborativen Dienste in den verschiedensten Anwendungsfällen Einsatz finden. Dies gilt sowohl für Anwendungen im privaten als auch im kommerziellen Bereich. Die gezeigten Beispiel-Apps stammen zwar nicht aus dem Business-Bereich, aber mit *Mobilis* lassen sich auch vielfältige Geschäftsszenarien umsetzen. Genannt seien etwa Orts- bzw. kontextbasierte Werbung, Friendfinder-Anwendungen, Touristenführer, Anbieter-Nachfrager-Matching mit Ortsbezug oder gruppenbasierte Kommunikation bzw. Bearbeitung von Dokumenten. Ebenso können bestehende Business-Anwendungen durch kollaborative Features, die Einbeziehung bestehender Communities für Feedback oder Ansätze des Crowd-Sourcing erweitert werden.

Im Business-Bereich spielt Sicherheit eine sehr wichtige Rolle. Durch die integrierten Sicherheitsmechanismen von XMPP kann *Mobilis* diese Anforderung erfüllen. Sämtliche Kommunikation läuft verschlüsselt ab und für unternehmensinter-

<sup>10</sup> <http://mobilisplatform.sourceforge.net>

ne Dienste lassen sich leicht eigene XMPP- und Mobilis Server aufsetzen, sodass sensible Geschäftsdaten das Unternehmensnetzwerk nicht verlassen.

Technisch wird durch die Nutzung von XMPP als Kommunikationsprotokoll der Großteil der Mobilis-Komponenten plattformunabhängig nutzbar. Außerdem können durch den webbasierten Zugriff neben Android auch weitere mobile Plattformen unterstützt werden. Es wird lediglich ein Webbrower benötigt. Darüber hinaus ist jedoch auch die Umsetzung von nativen Anwendungen innerhalb anderer Plattformen wie iOS oder Windows Phone denkbar. Dafür müssten jedoch die clientseitigen Mobilis-Komponenten (Beans, MXA, Client Services) für die jeweilige Plattform einmalig implementiert werden.

Für die Zukunft ist eine Erweiterung des Dienstkonzeptes der Mobilis-Plattform geplant. So sollen Dienstschnittstellen, aber auch nicht-funktionale Eigenschaften mithilfe einer Dienstbeschreibungssprache formal spezifiziert werden. Auf Basis der Beschreibung könnten während der Dienstentwicklung erste Codebausteine generiert werden. Ein bedarfsgerechtes Deployment würde für mehr Flexibilität sorgen. Moderne Dienste können außerdem Funktions- bzw. Qualitätsgarantien geben und Kontrollmöglichkeiten anbieten.

Die Mobilis-Plattform wird stetig weiterentwickelt und erweitert. Die verwendeten Protokolle, alle umgesetzten Anwendungen und natürlich auch die Plattform-Komponenten sind frei auf Sourceforge verfügbar, sodass die Ergebnisse auch in anderen Forschungsprojekten verwendet werden können.

## Literaturverzeichnis

1. Heyer C (2008) Mobile Social Software. PhD thesis, The University of Queensland
2. Endler M, Skyrme A, Schuster D, Springer T (2011) Defining Situated Social Context for Pervasive Social Computing. In: Second IEEE Workshop on Pervasive Collaboration and Social Networking (PerCol), Seattle, USA
3. Arb M, Bader M, Kuhn M, Wattenhofer R (2008) VENETA: Serverless Friend-of-Friend Detection in Mobile Social Networking, in 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Avignon, France
4. Beach A, Gartrell M, Akkala S, Elston J, Kelley J, Nishimoto K, Ray B, Razgulin S, Sundaresan K, Surendar B, Terada M, Han R (2008) WhozThat? Evolving an ecosystem for context-aware mobile social networks. IEEE Network 22(4):50–55
5. Li KA, Sohn TY, Huang S, Griswold WG (2008) Peopletones: a system for the detection and notification of buddy proximity on mobile phones. In: MobiSys '08: Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services, Breckenridge, CO, USA
6. Carreton AL, Harnie D, Gonzalez Boix E, Scholliers C, v Cutsem T, De Meuter W (2010) Urbiflock: An experiment in Dynamic Group Management in Pervasive Social Applications. In: First International Workshop on Communication, Collaboration and Social Networking in Pervasive Computing Environments (PerCol), Mannheim, Germany
7. Gartrell CM (2008) SocialAware: Context-Aware Multimedia Presentation via Mobile Social Networks, Master Thesis, University of Colorado, Department of Computer Science
8. Bottazzi D, Montanari R, Tominelli A (2007) Context-Aware Middleware for Anytime, Anywhere Social Networks. IEEE Int Sys 22:23–32

9. Gupta A, Kalra A, Boston D, Borcea C (2009) MobiSoC: a middleware for mobile social computing applications. *Mobile Networks Appl* 14:35–52
10. Pietiläinen A-K, Oliver E, LeBrun J, Varghese G, Diot C (2009) MobiClique: Middleware for Mobile Social Networking. In: WOSN '09: Proceedings of the 2nd ACM Workshop on Online Social Networks, Barcelona, Spain
11. Saint Andre P (2008) XEP-0045: Multi-User Chat. XMPP Standards Foundation. Tech Rep. Online: <http://xmpp.org/extensions/xep-0045.html>, 15. Juli 2011
12. Muldowney T, Miller M, Eatmon R (2004) XEP-0096: SI File Transfer. XMPP Standards Foundation. Tech Rep. Online: <http://xmpp.org/extensions/xep-0096.html>, 15. Juli 2011
13. Hildebrand J, Millard P, Eatmon R, Saint Andre P (2008) XEP-0030: Service Discovery. XMPP Standards Foundation. Tech Rep Online: <http://xmpp.org/extensions/xep-0030.html>, 15. Juli 2011
14. Millard P, Saint-Andre P, Meijer R (2010) XEP-0060: Publish-Subscribe. XMPP Standards Foundation. Tech Rep. Online: <http://xmpp.org/extensions/xep-0060.html>, 15. Juli 2011
15. Paterson I, Smith D, Saint Andre P, Moffitt J (2010) XEP-0124: Bidirectional-streams Over Synchronous HTTP (BOSH). XMPP Standards Foundation. Tech Rep. Online: <http://xmpp.org/extensions/xep-0124.html>, 15. Juli 2011
16. Paterson I, Saint Andre P XEP-0206: XMPP Over BOSH XMPP Standards Foundation, Tech. Rep., July 2010. Online: <http://xmpp.org/extensions/xep-0206.html>, 15. Juli 2011
17. Lübke R, Schuster D, Schill A (2011) MobilisGroups: Location-based Group Formation in Mobile Social Networks. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp 502–507

---

# Kapitel 29

## Mobile kollaborative Apps

Jan Gäbler, Ronny Klauck, Ralf Kopsch, Fuwen Liu, Mario Pink,  
Sebastian Schöpke und Hartmut König

**Zusammenfassung** Die steigende Leistungsfähigkeit mobiler Geräte erlaubt zunehmend die Nutzung kollaborativer Anwendungen. Die Entwicklung mobiler kollaborativer Anwendungen (Apps) ist aufwendig, da sie vielfach Anpassungen an unter-

---

Jan Gäbler

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze  
und Kommunikationssysteme

E-mail: jgaebler@informatik.tu-cottbus.de

Ronny Klauck

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze  
und Kommunikationssysteme

E-mail: rk@informatik.tu-cottbus.de

Ralf Kopsch

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze  
und Kommunikationssysteme

E-mail: rkopsch@informatik.tu-cottbus.de

Fuwen Liu

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze  
und Kommunikationssysteme

E-mail: lfw@informatik.tu-cottbus.de

Mario Pink

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze  
und Kommunikationssysteme

E-mail: pink@informatik.tu-cottbus.de

Sebastian Schöpke

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze  
und Kommunikationssysteme

E-mail: seb@informatik.tu-cottbus.de

Hartmut König

Brandenburgische Technische Universität Cottbus, Lehrstuhl Rechnernetze  
und Kommunikationssysteme

E-mail: koenig@informatik.tu-cottbus.de

schiedliche Ablaufumgebungen erfordert. Im Beitrag wird die Entwicklung mobiler kollaborativer Apps auf der Basis der *uBeeme*-Plattform beschrieben, die wiederwendbare Basiskomponenten für solche Apps bereitstellt. Die Struktur der Plattform sowie wichtige Komponenten (Mobilitätsmanagement, Gruppenkommunikation, Lokalisierung) werden vorgestellt und diskutiert. Abschließend wird die Entwicklung einer App mit Hilfe der Plattform skizziert.

## 29.1 Motivation

Mobilität und Kooperation werden zu den prägenden Elementen des Internets der Zukunft gehören. Schon heute spielen beide Punkte eine wesentliche Rolle bei der Gestaltung von Internetanwendungen. Im Internet werden kollaborative Anwendungen bereits intensiv genutzt, z. B. über Plattformen wie Sharepoint Workspace von Microsoft oder in Form von kooperativen Werkzeugen wie Whiteboards, Application Sharing und Filetransfer, die in Audio- oder Videokonferenzsoftware eingebunden werden. Die zunehmende Mobilität der Internetnutzer erfordert, dass kollaborative Anwendungen nicht mehr nur vom Festnetz zugänglich sind, sondern auch Partner einbeziehen, die unterwegs sind und nur über ein mobiles Endgerät zu erreichen sind. Die steigende Leistungsfähigkeit mobiler Geräte, insbesondere von Netbooks, Tablets und Smartphones, erlaubt dies. Dadurch kann gewährleistet werden, dass mobile Nutzer in relevante Entscheidungsprozesse einbezogen werden und sie die dafür erforderlichen Informationen bereitgestellt bekommen, unabhängig davon, ob sie sich in einem Fest- oder Mobilnetz befinden.

Es ist ein breites Spektrum mobiler kollaborativer Anwendungen denkbar, z. B. Telediagnose, eGaming, Überwachung/Tracking, mobile Groupware, mobile Business-to-Business (B2B)-Anwendungen u. v. m. Die Entwicklung jedes einzelnen dieser Dienste ist aufwendig, nicht nur bzgl. des softwaretechnischen Entwicklungsprozesses, sondern weil für kollaborative Dienste auch vielfach Anpassungen an unterschiedliche Ablaufumgebungen (Hardwareschnittstellen, Betriebssystemsoftware) erforderlich sind. Häufig wechseln die Schnittstellenanforderungen mit dem Erscheinen neuer Systeme, wie auch die Anforderungen an die Gestaltung der Dienste. Daneben ist aber auch die Integration innovativer Konzepte wie der nahtlose Übergang zwischen verschiedenen Netzen, die Integration von Webdiensten und die Gewährleistung relevanter Schutzziele wie Vertraulichkeit, Authentizität und Privatheit gefordert.

Um trotz dieser Vielfalt von Gestaltungsanforderungen eine effiziente Anwendungsentwicklung zu ermöglichen, bietet sich die Entwicklung einer Plattform an, die wichtige Funktionen kollaborativer Anwendungen (Apps) bereitstellt. Das erlaubt es dem Anwendungsentwickler, sich auf sein zu lösendes Problem zu konzentrieren und befreit ihn von der wiederholten Entwicklung gleichartiger Komponenten. Eine solche Plattform wurde mit dem *uBeemE*-Konzept im Rahmen des BMBF-ForMaT-Programms am Lehrstuhl Rechnernetze und Kommunikationssysteme der BTU Cottbus entwickelt.

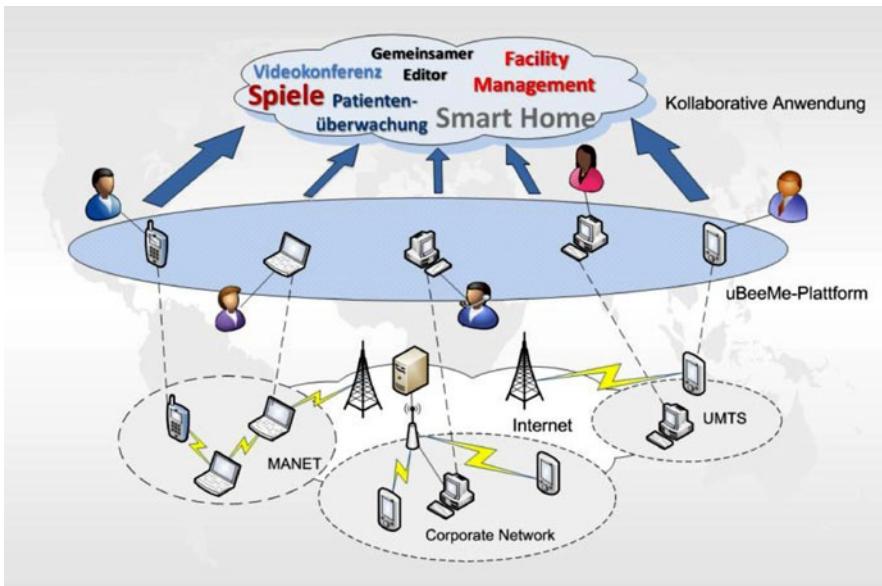


Abb. 29.1 *uBeeMe-Plattform*

Die *uBeeMe-Plattform* (siehe Abb. 29.1) stellt wiederverwendbare Basiskomponenten für die Entwicklung mobiler kollaborativer Anwendungen wie Mehr-Teilnehmer-Spiele, mobile Audio-/Videokonferenzen oder Patientenüberwachung bereit. Sie soll den Anwendungsentwickler beim Lösen immer wiederkehrender Probleme wie Netzwechsel, Gruppenkommunikation, Lokalisierung oder die Traversierung von Firewalls und NAT (Network Address Translation)-Routern entlasten. Jedoch ist die *uBeeMe-Plattform* keine Middleware, da über klassische Protokolle auch mit Nicht-*uBeeMe*-Partnern kommuniziert werden kann. Beispielsweise können diese mit Plattform-Anwendern über Chat-Programme, wie z. B. *Pidgin*, Nachrichten austauschen.

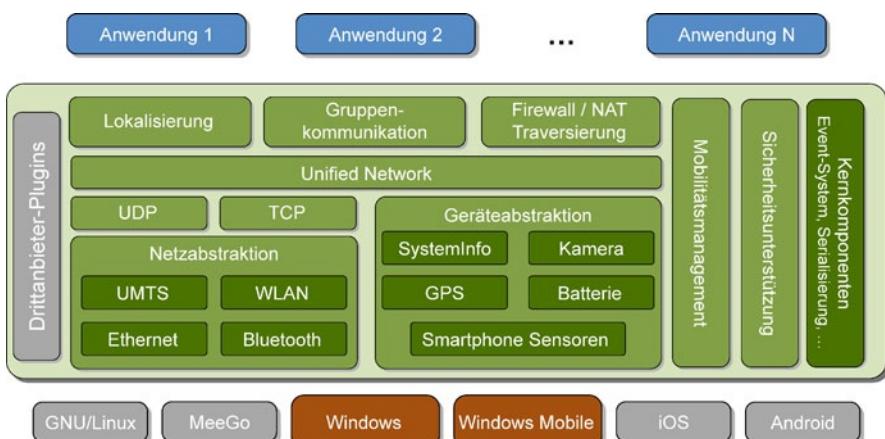
Um die unterschiedlichen Anforderungen an die Plattform zu erfassen, wurden verschiedene Anwendungsszenarien untersucht. Als Beispiel betrachten wir in diesem Beitrag ein Mehrteilnehmerspiel (siehe Abschn. 29.7), bei dem sich Teilnehmer aus verschiedenen Netzumgebungen ad hoc zu einem Spiel zusammenfinden. Dazu benötigt der mobile Nutzer eine ubiquitäre Netzanbindung, die es ihm bei einem Netzwechsel ermöglicht, ohne Unterbrechung zu kommunizieren und zu kooperieren. Für die Gestaltung der Plattform ergeben sich daraus die folgenden vier Hauptanforderungen. (1) *Kollaborationsunterstützung*: Die Plattform soll eine spontane Kommunikation und Kooperation über verschiedene Netze mit wechselnden Partnern unterstützen, ohne auf eine komplexe Server-Infrastruktur zurückgreifen zu müssen. Deshalb wurde ein Peer-to-Peer (P2P)-Ansatz gewählt mit dem Ziel, eine schlanke Overlay-Struktur aufzubauen, die anwendungsbezogen konfiguriert werden kann. (2) *Mobilitätsunterstützung*: Die Anwendungen sollen netzunabhängig sein und gleichzeitig von verschiedenen Netzen (Fest- und Mobilnetzen) aus genutzt

werden können. Ein Netzwechsel eines Nutzers soll die Anwendung möglichst nicht unterbrechen. (3) *Abstraktion, Kapselung und Erweiterbarkeit*: Um die Kooperation zwischen Nutzern zu erlauben, die unterschiedliche Endgeräte nutzen oder in unterschiedlichen Umgebungen aktiv sind, muss die Plattform ein großes Spektrum an Betriebssystemen und Netzchnittstellen unterstützen. Neue Basiskomponenten und Umgebungen sollen einfach integrierbar sein. (4) *Sichere Kommunikation*: Für Anwendungen, die dies erfordern, z. B. Beratungen oder die gemeinsame Bearbeitung von Dokumenten, soll eine vertrauliche und sichere Gruppenkommunikation im mobilen Umfeld unterstützt werden. Da keine zentrale Sicherheitsinstanz genutzt wird, ist ein dezentrales Schlüsselmanagement für die sichere Bereitstellung und Verteilung von Schlüsseln erforderlich.

## 29.2 Die *uBeeMe*-Plattform

Die *uBeeMe*-Plattform (siehe Abb. 29.2) besitzt eine modulare Struktur, welche es dem Anwendungsentwickler erlaubt, nur die für die jeweilige Anwendung benötigten Plattformfunktionen auszuwählen. Dies spart vor allem auf mobilen Geräten Ressourcen ein. Durch einen Plug-in-Mechanismus ist es weiterhin möglich, Teile der Plattform bei Bedarf dynamisch zur Laufzeit nachzuladen sowie nach Benutzung wieder zu entladen.

Die *uBeeMe*-Plattform ist in ihrer jetzigen Entwicklungsstufe als Bibliothek realisiert. Das bietet den Vorteil, dass sie einfach zu nutzen ist und im selben Anwendungskontext läuft. Dies hat jedoch den Nachteil, dass für die Plattform nicht ohne weiteres zentrale Entscheidungen getroffen werden können, wenn mehrere auf der Plattform basierende Anwendungen auf einem Endgerät laufen. Deshalb wird die Realisierung als lokaler Betriebssystem-Dienst angestrebt, wodurch für alle ange-



**Abb. 29.2** *uBeeMe*-Plattformarchitektur

meldeten Anwendungen einheitliche Entscheidungen auf einfache Art und Weise getroffen werden können.

Anwendungsentwicklungen für mobile Endsysteme sind derzeit noch stark von der jeweiligen Betriebssystemumgebung abhängig. *uBeeMe* soll eine betriebssystemübergreifende Anwendungsentwicklung unterstützen. Es soll schrittweise für alle wichtigen Mobil- und Festnetzbetriebssystemumgebungen bereitgestellt werden. Für die Prototyp-Entwicklung wurde Windows sowohl für Desktoprechner als auch für mobile Geräte ausgewählt, wobei die Plattform auf zukünftige Portierungen bereits jetzt durch Kapselung systemspezifischer Komponenten vorbereitet ist. Gegenwärtig wird eine Portierung auf *Android* durchgeführt.

Die *uBeeMe*-Plattform gliedert sich in *Kernkomponenten* sowie in Module zur Unterstützung der Anwendung auf (siehe Abb. 29.2). Als Kernkomponenten werden Komponenten bezeichnet, die grundlegende Funktionen bereitstellen. Dazu zählen u. a. ein *Event-System* zur Verteilung von plattforminternen Ereignissen, eine Serialisierung, welche Nachrichten zu Datenpaketen verpackt, und das *Modulmanagement*, das für das Laden und Entladen von Modulen zuständig ist.

Die *uBeeMe*-Plattform soll Anwendungen unterstützen, die in unterschiedlichen Netzumgebungen ablaufen. Dafür ist die Komponente *Netzabstraktion* zuständig. Sie stellt eine einheitliche Schnittstelle für den Zugriff auf UMTS, WLAN, Ethernet und Bluetooth bereit und ermöglicht den Auf-/Abbau von Verbindungen sowie die Abfrage von Verbindungsparametern. Analog können über die *Geräteabstraktion* weitere Betriebssystem-Komponenten und Geräte, z. B. GPS, Lage-/Beschleunigungssensoren, Batteriezustände und andere Systeminformationen, über einheitliche Schnittstellen angesprochen werden.

Das Modul *Unified Network* fasst alle verfügbaren Netze zu einem virtuellen Netz zusammen und übernimmt die Wegewahl (Routing) für die zu versendenden Daten. Die Anwendung wird dadurch in die Lage versetzt, in einer einheitlichen Art und Weise Punkt-zu-Punkt-Verbindungen mit Teilnehmern in verschiedenen Netzen (mit potentiell unterschiedlichen Netztechnologien) aufzubauen. In Abb. 29.3 ist beispielhaft eine Netzkonfiguration bestehend aus drei Rechnern dargestellt. Dabei besitzt der Knoten X keine direkte Verbindung zum Knoten Z oder zum Internet, sondern nur eine Ad-hoc-WLAN-Verbindung zum Knoten Y. Das *Unified Network*-Modul sorgt dafür, dass alle Konten miteinander kommunizieren können, indem es ein Routing der Daten über Netzgrenzen hinweg durchführt. Für Ad-hoc-Netze übernimmt das *Unified Network* das Routing vollständig, in Infrastruktur-Netzen wird das eigentliche Routing durch das Betriebssystem übernommen. Knoten, die in mehreren Netzen aktiv sind, stellen anderen Knoten Zugriff auf ihre Netze zur Ver-

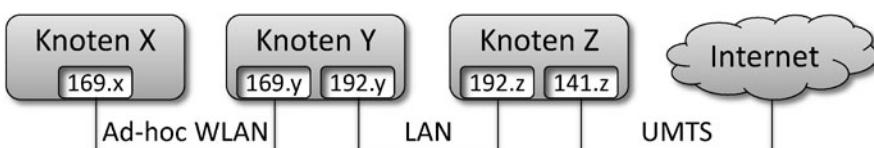


Abb. 29.3 Kommunikation im *Unified Network*

fügung, indem sie als Gateway agieren. Das *Unified Network* ist für die Anwendung transparent und ermöglicht es, auch mit Nicht-*uBeeMe*-Partnern zu kommunizieren.

Aufbauend auf diesen Komponenten enthält die *uBeeMe*-Plattform Module für das Mobilitätsmanagement, die Gruppenkommunikation, die Lokalisierung von Teilnehmern, die Firewall/NAT-Traversierung sowie eine sichere Kommunikation. Diese Module werden in den nachfolgenden Abschnitten näher erläutert.

## 29.3 Mobilitätsmanagement

Die *uBeeMe*-Plattform unterstützt eine *always on*-Nutzung. Dazu wird durch das *Mobilitätsmanagement* in enger Wechselwirkung mit der *Geräte- und Netzabstraktion* sowie dem *Unified Network* ein möglichst nahtloser Netzwechsel gewährleistet. Ein derartiger Netzwechsel (Handover) wird sowohl bei Verlust des Netzzugangs als auch bei Unterschreitung einer definierten Dienstqualität des Netzes bzw. Änderungen der Anwendungsanforderungen eingeleitet. Da heterogene Netze sich in ihren Anforderungen und Eigenschaften unterscheiden, werden dafür rekonfigurierbare Handover-Mechanismen und -Protokolle eingesetzt. Diese Mechanismen erlauben es Nutzern und Anwendungen, ihre Handover-Strategien zur Laufzeit anzupassen. Somit lassen sich Handover-Entscheidungen sowohl für eine Verbindungssicherung als auch für eine Verbindungsoptimierung auf dem mobilen Gerät realisieren.

Das rekonfigurierbare Mobilitätsmanagement (siehe Abb. 29.4) interagiert mit dem mobilen Gerät, der Netzschnittstelle und den unterschiedlichen Protokollen. Jede dieser Interaktionen wird über Parameter gesteuert, die von einem *Systemmonitor* mit Hilfe kontinuierlicher Messungen auf der *Netz- und Geräteabstraktion* ermittelt und aggregiert werden. Aufbauend auf diesen Informationen werden graduell abstuftbare Handover-Algorithmen verwendet, die innerhalb des *Handover-Entscheiders* über die Ausführung eines Handover entscheiden. Die verwendeten Algorithmen werden dafür anhand von Nutzer-, Anwendungs- und Umgebungskriterien ausgewählt und in ihrer Berechnungskomplexität sowie Ausführungs frequenz an die unterschiedliche Parameterbasis der Betriebssysteme sowie an die vom Nutzer geforderte Genauigkeit angepasst. Abschließend wird während der *Handover-*

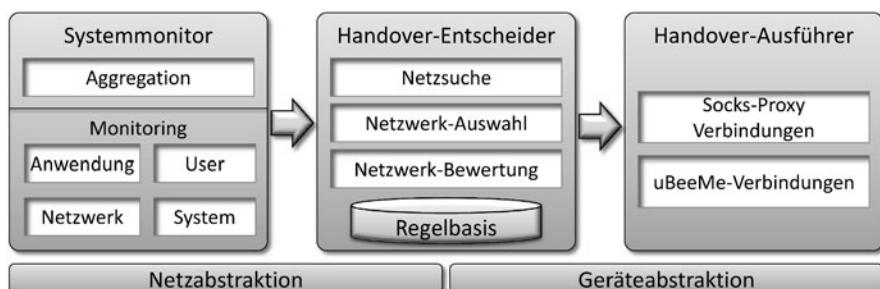


Abb. 29.4 *uBeeMe*-Mobilitätsmanagement

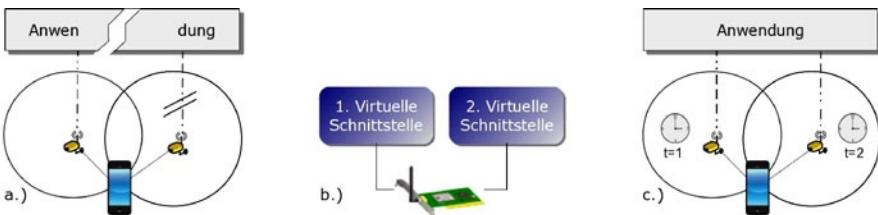
Ausführung jede Verbindung nahtlos in ein anderes Netz übernommen. Zur Optimierung dieser Teilprozesse werden unterschiedliche Strategien verwendet:

(1) Eine Herausforderung bei jedem Netzwechsel besteht in der Suche nach verfügbaren Netzen und Netzzugangspunkten, für die eine Zugangsberechtigung vorliegt. Ein inhärentes Problem dabei ist, dass jede Netzsuche eine Aktivierung der Ziel-Netzschnittstelle erfordert. Abhängig von der Suchfrequenz erhöht sich somit der Energieverbrauch. Konfigurierbare Funkschnittstellen, die alle Frequenzen überwachen können, sind eine mögliche Lösung, werden aus Kostengründen gegenwärtig aber selten eingesetzt. Um die Netzsuche dennoch optimieren zu können, werden zentrale Lokalisierungsdienste wie z. B. OpenBMAP [1] und Skyhook [2] verwendet, die Informationen über verfügbare Netze entsprechend bestimmter Positionsindikatoren, wie GPS-Koordinaten und Zellen-Identifikator, bereitstellen. Somit kann die Existenz von Netzen abgeschätzt und auf eine unnötige Aktivierung von Netzschnittstellen verzichtet werden.

(2) Aufbauend auf diesen Netzinformationen und der Analyse weiterer Verbindungs-, System- und Anwendungsgüteparameter wird über die Notwendigkeit eines Handover entschieden. Die Entscheidung für oder gegen einen Handover erfordert dabei i. d. R. Berechnungen, die in Abhängigkeit von der geforderten Genauigkeit der Entscheidung den Energieverbrauch des Gerätes beeinflussen [3].

Das Mobilitätsmanagement verwendet daher einen mehrstufigen Prozess aus Handover-Initiierung und -Entscheidung, der komplexe Berechnungen auf ein Minimum reduziert. Die Initiierung überwacht mit leichtgewichtigen Schwellwertalgorithmen den Zustand des mobilen Systems im Ruhe- und im normalen Bewegungszustand. Bei einer Zustandsänderung, die prognostiziert zu kritischem Verhalten führt, wird die eigentliche Handover-Entscheidung gestartet und mit Hilfe eines adaptiven Fuzzy-Logik-basierten Entscheidungsalgorithmus über den Netzwechsel entschieden.

(3) Im Anschluss an die Netzwechselentscheidung wird das optimale Zielnetz ausgewählt und der Netzwechsel vollzogen. Zur Auswahl des optimalen Netzes werden potentielle Netze passiv auf der Basis gesammelter Informationen und aktiv durch Abfrage von Netzparametern der Vermittlungsschicht und höherer Protokollschichten bewertet. Kritisch ist, dass nicht jedes Netz eine Dienstnutzung gewährleisten muss, da eine fehlende Internetverbindung, Firewalls oder NAT-Router die Kommunikation unterbinden können. Existierende Handover-Lösungen werden daher überwiegend auf idealisierten Netzen evaluiert und getestet. Um derartige Randbedingungen besser einzuschätzen, nutzt das *Mobilitätsmanagement* Handover-Algorithmen, die für einen Einsatz von NAT-Traversierungsmechanismen optimiert sind. Das ermöglicht es, den zeitlichen Umfang der Traversierung in die Handover-Entscheidung und ihre Durchführbarkeit in die Zielnetzbewertung einzubeziehen. Eine weitere Optimierung der Netzauswahl wird zudem durch die Vermessung der Leistungsfähigkeit höherer Protokollschichten erzielt. Abhängig von der Anzahl der Netze entsteht somit ein hoher zeitlicher Aufwand. Dabei besteht das bekannte Problem, dass lange Messperioden gute Qualitätsanalysen für wenige Netze ermöglichen, wohingegen kurze Messperioden ungenaue Qualitätsanalysen für viele Netze ermöglichen.



**Abb. 29.5** Virtualisierung von 802.11-Schnittstellen

(4) Ein grundlegendes Problem der Vermessung besteht darin, dass mobile Geräte in der Regel nur mit einer WLAN-Schnittstelle ausgestattet sind. Jeder Verbindungsauflauf zur Vermessung eines WLAN-Netzes unterbricht somit die Kommunikation der Anwendung (siehe Abb. 29.5a). Zur Lösung dieses Problems wird daher eine Virtualisierung für IEEE 802.11-Schnittstellen verwendet [4]. Dabei werden auf der physischen Netzschmittstelle mehrere logische Netzschmittstellen erzeugt (siehe Abb. 29.5b), zwischen denen die physische Netzschmittstelle im Zeitmultiplexverfahren wechselt (siehe Abb. 29.5c). Jede dieser Netzschmittstellen kann dementsprechend mit einem WLAN-Netz verbunden sein. Damit können mehrere Netze annähernd „parallel“ auf ihre approximierte Dienstgüte hin analysiert werden.

(5) Um die Ausführung mehrerer kollaborativer Anwendungen auf einem System zu optimieren, sollte die Handover-Funktionalität nicht direkt in die Anwendung integriert, sondern im angestrebten Betriebssystem-Dienst realisiert werden. Somit bietet sich die Möglichkeit, mehrere Anwendungen bei einem Netzwechsel zu steuern.

## 29.4 Mobile Gruppenkommunikation

Mobile kollaborative Anwendungen, wie z. B. Auktionen mit mobilen Teilnehmern, mobile Multiplayer-Spiele oder die Koordination und Überwachung von Einsatzkräften sowie Infrastrukturelementen während eines Katastropheneinsatzes bedingen zumeist die Abstraktion einer geschlossenen Gruppe, die zuverlässig verschiedene Dienste zur Verfügung stellt. Typischerweise bestehen diese Gruppen aus mobilen und stationären Teilnehmern, wobei die Eigenschaften der mobilen Teilnehmer innerhalb dieser Gruppe bezüglich ihrer Fähigkeiten, Ressourcen, Verbindungs- und Mobilitätscharakteristiken variieren. Etablierte Gruppenkommunikationssysteme wie ISIS [5], Totem [6], GCP [7] oder NICE [8] basieren auf LAN- bzw. WAN-zentrierten Ansätzen und unterstützen Mobilität nur unzureichend. Häufig schwankende Teilnehmerzahlen (Churns), unerwartete Verbindungsabbrüche sowie schwankende QoS-Parameter werden in diesen Protokollen nur ungenügend bzw. überhaupt nicht adressiert.

Für die *uBeeMe*-Plattform wurde mit *Moversight* ein Gruppenkommunikationsprotokoll entwickelt, das die mobilitätsbezogenen Probleme durch einen leichtge-

wichtigen P2P-Ansatz adressiert. *Moversight* stellt einen virtuell synchronen Gruppenkommunikationsdienst bereit, der es den Teilnehmern gestattet, Gruppenentscheidungen lokal zu treffen. Die Teilnehmer bilden eine geschlossene Gruppe, welcher Peers nur nach expliziter Einladung beitreten können. Es kommt dabei im gesamten Sitzungsverlauf ohne zentrale Infrastruktur-Elemente, wie Bootstrap- oder Gruppenzugangsserver, Sequenzer oder Rechtebroker, aus. Dadurch wird eine flexible, skalierbare und selbst-organisierende Struktur (vgl. [9]) definiert, welche durch die Zusammenarbeit mit dem *Unified Network* (vgl. Abschn. 29.2) auch für Ad-hoc-Umgebungen geeignet ist.

*Moversight* verbindet die Gruppenmitglieder über ein Cluster-basiertes P2P-Overlay, das die tatsächlichen Netzstrukturen abstrahiert. Durch die Cluster-Struktur wird eine Vereinfachung der Teilnehmerverwaltung und Reduzierung der Sendevorgänge pro Peer erreicht. Jedem Teilnehmer werden beim Eintritt in die Gruppe ein Cluster und seine Rolle (Slave oder Master) darin zugewiesen. Als Slave werden Peers der Gruppensitzung bezeichnet, welche Gruppennachrichten ausschließlich von ihrem Cluster-Master empfangen bzw. an diesen senden. Der Cluster-Master ist für die Nachrichtenvermittlung und Sicherung der virtuellen Synchroneität zuständig. Dazu kommuniziert er mit allen Slaves seines Clusters sowie den Master-Peers der anderen Cluster. Der Wissenstand über die Gruppenmitglieder unterscheidet sich zwischen Master und Slave jedoch nicht, sodass prinzipiell jeder Peer innerhalb der Gruppe als Master oder Slave agieren kann. Grundsätzlich wird durch die Cluster-Struktur des Overlays eine deutliche Reduzierung des Signalaufwandes erreicht, was allgemein die Ressourcen der mobilen Teilnehmer schont.

Die konkreten Anforderungen an die Struktur des Overlays unterscheiden sich dabei jedoch von Anwendung zu Anwendung. Mal kann die Reduzierung der maximalen Verbindungen pro Teilnehmer oder die durchschnittliche Latenz innerhalb des Cluster die bestimmende Eigenschaft für die Overlay-Struktur sein, mal deren geographische Position. Damit *Moversight* für verschiedene Situationen eine optimale Anpassung der Overlay-Struktur an die Erfordernisse mobiler, ressourcenschwacher Peers bieten kann, werden verschiedene *Peer-Platzierungsstrategien* eingesetzt. Diese bestimmen die Overlay-Parameter, wie Anzahl der Cluster, Peers pro Cluster, und sind verantwortlich für die Verteilung der einzelnen Peers auf die Cluster. Die Verteilung der Peers erfolgt dabei immer nach der durch die verwendeten Peer-Platzierungsstrategien definierte Metrik, wie Verbindungseigenschaften, geographische Positionen oder lokale Ressourcenausstattung. In Abb. 29.6 ist beispielhaft eine einfache Platzierungsstrategie schematisch dargestellt, in der zunächst drei Cluster innerhalb der Gruppe angelegt werden und alle weiteren Sitzungsteilnehmer auf diese Cluster verteilt werden. Dabei startet im Schritt 1) der erste Peer zunächst eine Gruppensitzung und lädt einen zweiten Peer explizit zur Teilnahme ein. Da beide Peers die gleiche Peer-Platzierungsstrategie verwenden (hier: *Erzeuge Master zuerst*), können beide Peers die Position des beigetretenen Peers lokal berechnen. Alle dazu notwendigen Daten werden bei der Aushandlung des Gruppenbeitritts ausgetauscht. Nach dem Beitritt des dritten Peers als Master sei die maximale Cluster-Anzahl für diese Gruppensitzung erreicht und der vierte Peer wird

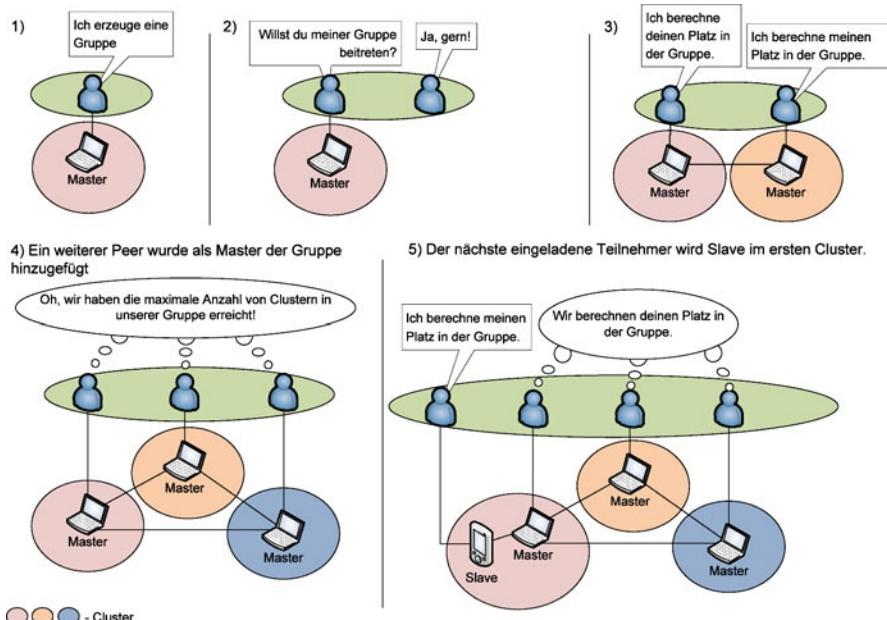


Abb. 29.6 Schematische Darstellung einer einfachen Peer-Platzierungsstrategie

als Slave dem ersten Cluster hinzugefügt. Die Position des neuen Peers wird wiederum durch alle Peers lokal bestimmt.

Abbildung 29.7 veranschaulicht eine mögliche, resultierende Overlay-Struktur für eine Strategie *Erzeuge Master zuerst* mit maximal zwei Clustern. Die geschlossene Gruppe wird auf das erzeugte Overlay abgebildet, welches wiederum die tatsächlichen Netzstrukturen abstrahiert. Der individuelle Netzzugang wird unter Nutzung des *Unified Networks* realisiert.

Der Nachrichtenaustausch innerhalb von *Moversight* erfolgt über ein virtuell synchrones und zuverlässiges Multicast mit vollständig sortierter Auslieferung auf Basis der erweiterten Lamport-Zeit [10]. Statt in einem logischen Ring mit Tokenumlauf, wie in GCP oder TOTEM, über welchen die virtuelle Synchronität geregelt wird, ist bei *Moversight*, wie bereits erwähnt, der Master eines Clusters für die Sicherung der Synchronität verantwortlich. Für diese strenge Synchronität hat man sich entschieden, da dies die Erstellung von kollaborativen Anwendungen mit einem gemeinsamen Zustandsraum erlaubt, wie es beispielsweise in Auktionen mit mobilen Teilnehmern wünschenswert ist. Dadurch werden zusätzliche komplexe Transaktionsverfahren oder Zustandstransfers vermieden.

Durch das *Unified Network* ist *Moversight* in der Lage, Gruppenkommunikationsitzungen mit stationären und mobilen Teilnehmern zu bilden, welche spontan oder infrastrukturbasiert kommunizieren. Im Verlaufe der Gruppensitzung kann es passieren, dass die Verbindung zu einzelnen Teilnehmern der Gruppe abreißt oder

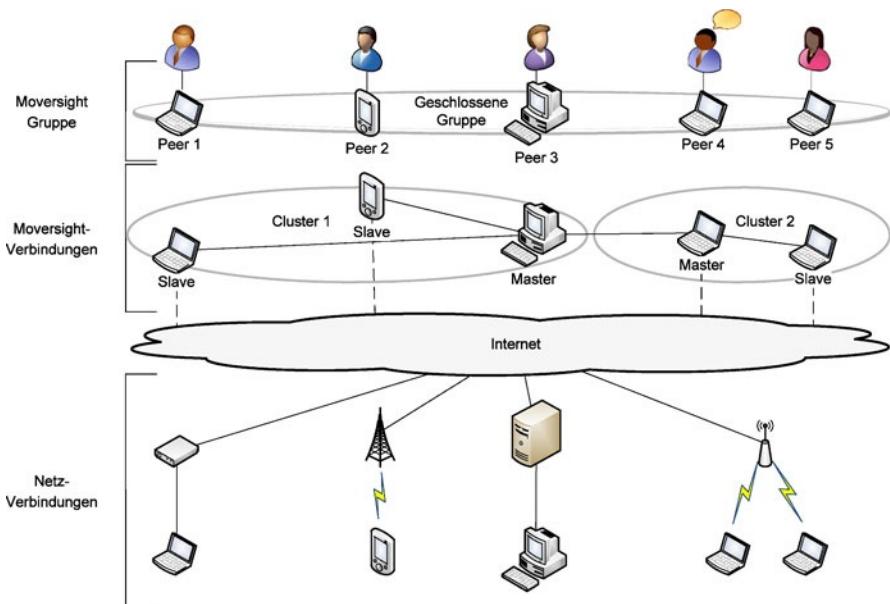


Abb. 29.7 Overlay-Struktur in *Moversight*

sich die Struktur der Gruppe in eine rein mobile bzw. stationäre wandelt. Diesen Situationen begegnet *Moversight* mit drei Konzepten.

(1) Die Struktur der Gruppe wird regelmäßig analysiert und gegebenenfalls neu aufgebaut. Dies geschieht unter Nutzung der Peer-Platzierungsstrategien. Die Häufigkeit der Analysen liegt im Sekundenbereich und ist konfigurierbar.

(2) *Moversight* besitzt einen Network-Failure-Detection-Dienst, der auf Basis einer hybriden Heartbeat-/Pingstrategie und unter Nutzung des aktuellen Datenverkehrs innerhalb der Gruppe den Verbindungsstatus der einzelnen Peers überwacht. Ein Peer, der die Verbindung zur Gruppe verloren hat, wird in der Gruppe markiert und kann sich innerhalb einer bestimmten Zeit zurückmelden. Sollte dies nicht geschehen, wird er aus der Gruppe entfernt.

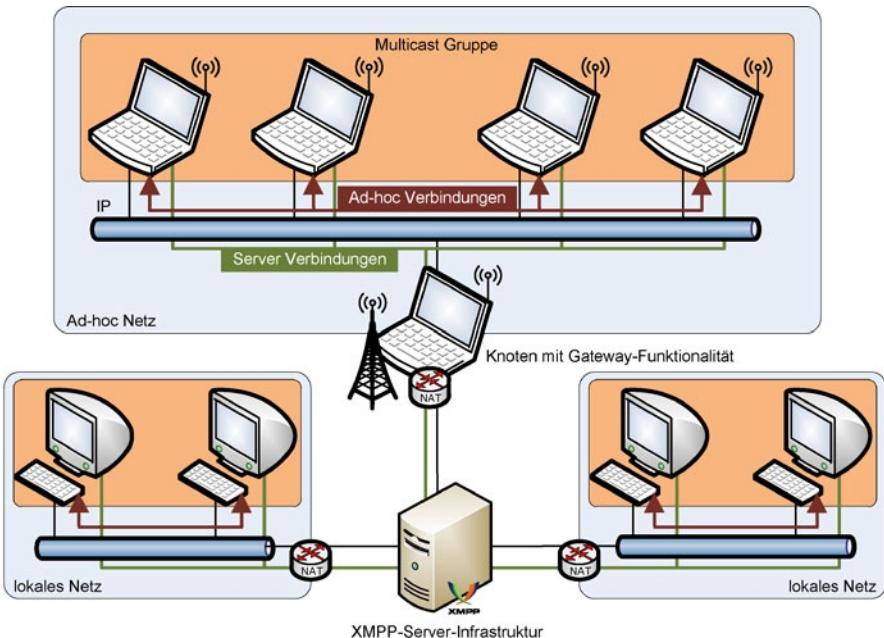
(3) Außerdem definiert *Moversight* für jeden Peer einen virtuellen *Location Descriptor* [11], bestehend aus der Netzadresse und einer Peer-ID. Dieser wird genutzt, um bei einem Peer, der das Netz wechselt, zunächst nur die Netzadresse zu ändern, ohne dabei seine Position innerhalb der Gruppe anzupassen. Damit kann aus Nutzersicht die Topologie des Overlays über längere Zeitperioden stabil gehalten werden. Eine daraus möglicherweise resultierende ineffiziente Struktur wird durch die unter (1) beschriebene Struktur-Analyse behoben.

Durch die Kombination aus diesen drei Konzepten ist *Moversight* in der Lage, Handover zwischen Netzen zu unterstützen (vgl. Abschn. 29.3) und gleichzeitig mit hohen, variablen Churn-Raten umgehen zu können. Gruppenteilnehmer können dadurch bei Abbruch oder Wechsel der Funkverbindung wieder in die Gruppe integriert werden, ohne erneut der Gruppe beitreten zu müssen.

## 29.5 Lokalisierungsunterstützung

Dem *uBeeMe*-Gruppenkommunikationsprotokoll *Moversight* ist eine Lokalisierung vorgeschaltet, die die Transportadressen der Kommunikationsteilnehmer ermittelt. Jedem Benutzer ist eine feste Benutzerkennung zugeordnet, welche auf die jeweils gültige Transportadresse abgebildet wird. Das Ziel der Lokalisierungsunterstützung ist es, eine möglichst dezentrale und an die Umgebung adaptierbare Lösung zur Lokalisierung von Teilnehmern in Ad-hoc- und Infrastruktur-Umgebungen zu schaffen, die ein standardisiertes Signalisierungsprotokoll verwendet. Hierfür wurde das *Extensible Messaging and Presence Protocol* (XMPP) [12] ausgewählt. XMPP bietet eine Sammlung von offenen Technologien zur Umsetzung von Instant-Messaging (IM), Voice-over-IP (VoIP) und Kollaborationen an. Die Architektur von XMPP basiert auf einer dezentralen Client/Server-Struktur. Mehrere zentralisierte Netze werden durch Server zu einem Netz zusammengeschlossen, wodurch ein Single Point of Failure oder eine Überlastsituation verhindert wird. Während Teilnehmer mit Internetzugang sich direkt an einem XMPP-Server anmelden und authentifizieren können, sollen sich Teilnehmer in Ad-hoc-Umgebungen über die XMPP-Protokollerweiterung (XEP) Serverless Messaging finden. Dabei verwenden Benutzer stets dieselbe Benutzerkennung (JID). So ist es möglich, explizit nach einem Benutzer zu suchen, unabhängig davon, in welcher Umgebung er sich gegenwärtig befindet. Zur Kopplung der Ad-hoc-Netze mit der konventionellen XMPP-Infrastruktur wird die Weiterleitungsfunktionalität des *Unified Networks* genutzt. Zur Erbringung dieser Funktionalität agieren die Knoten mit Internetzugang als Gateway. Ein Gateway verfügt beispielsweise zusätzlich zum WLAN-Modul über ein UMTS- oder GPRS-Modul. Folglich können die Knoten eine dedizierte Verbindung zum XMPP-Server herstellen. Dies hat den Vorteil, dass sich jeder Knoten direkt gegenüber dem Server authentifiziert. Weiterhin ist so die Kompatibilität zum XMPP-Standard gewährleistet. Es kann die bestehende Server-Infrastruktur genutzt werden. Des Weiteren wird den Nutzern der *uBeeMe*-Plattform auf diese Weise die Möglichkeit gegeben, eine beliebige JID verwenden zu können und nicht auf eine von der Plattform bestimmte Benutzerkennung zurückgreifen zu müssen. Beispielsweise können die Anwender des Instant-Messengers *Google Talk* ihr Konto und die damit verknüpften Kontakte auf der *uBeeMe*-Plattform weiterverwenden.

In Abb. 29.8 ist eine solche Netzstruktur dargestellt. Die mobilen Teilnehmer bilden ein Ad-hoc-Netz und treten einer Multicast-Gruppe bei. Jeder Knoten wertet die in der Multicast-Gruppe publizierten Nachrichten aus und erstellt basierend darauf eine Teilnehmerliste. Das Gateway befindet sich in beiden Umgebungen und verbindet die Ad-hoc- und die Infrastruktur-Umgebung. Die Gateway-Funktionalität wird von den Knoten im Ad-hoc-Netz erbracht, die über einen Internetzugang verfügen. Die Knoten stellen diese Funktionalität zusätzlich zu den anderen Plattformfunktionalitäten bereit. Ein dedizierter Knoten, der ausschließlich als Gateway agiert, ist nicht erforderlich. Besitzt ein Knoten keinen Zugang zu einem Ad-hoc-Netz, jedoch einen dedizierten Internetzugang, dann kann der korrespondierende XMPP-Server direkt kontaktiert werden.



**Abb. 29.8** Kopplung eines Ad-hoc-Netzes mit einer XMPP-Infrastruktur

Zusätzliche Lokalisierungsdienste werden über die XMPP-Basisfunktionalität und die XMPP-Protokollerweiterungen bereitgestellt. Dies betrifft insbesondere XEP-0166 – *Jingle*, XEP-0174 – *Serverless Messaging* und XEP-0250 – *C2C Authentication using TLS*.

Zusätzlich wurde eine Verschlüsselung der XMPP-Kommunikation für beide Umgebungen eingeführt. Die Verschlüsselung und Authentifizierung der Client/Server-Verbindung ist ein fester Bestandteil der XMPP-Kernspezifikation. Dagegen wird die Authentifizierung der Teilnehmer in der Ad-hoc-Umgebung durch XEP-0250 und *Secure Remote Password* (SRP) [13] realisiert. Das SRP-Protokoll wurde ursprünglich zur Client/Server-Authentifizierung entworfen. Um SRP auch für eine paarweise Authentifizierung in Ad-hoc-Umgebungen verwenden zu können, waren Anpassungen erforderlich. Auf beiden Knoten wird während des Authentifizierungsvorgangs das Passwort eingegeben, welches die Parteien z. B. mündlich vereinbaren. Über den authentifizierten und verschlüsselten Kommunikationskanal wird anschließend die Lokalisierung durch Jingle, d. h. der Austausch der Transportadressen, durchgeführt. Die Umgehung von NAT-Systemen durch Techniken wie *Interactive Connectivity Establishment* (ICE) [14] ist heutzutage notwendig, da Internetanbieter zur Regulierung und zum Schutz des Datenverkehrs in mobilen Umgebungen NATs einsetzen.

Jingle wurde ausgewählt, da es auch die Lokalisierung von Teilnehmern hinter NAT-Systemen ermöglicht. Weiterhin sind die notwendigen XML-Nachrichten zum Adressaustausch bereits definiert und machen so die Festlegung eines eige-

nen Nachrichtenformats überflüssig. Befindet sich der zu lokalisierende Teilnehmer hinter einem NAT-System, werden die entsprechenden Informationen zur NAT-Traversierung übermittelt. Die Jingle *ICE-UDP Transport Method* spezifiziert hierfür die Attribute unter Verwendung des ICE-Protokolls. Eine NAT-Traversierung ist nur für die Infrastruktur-Umgebung und nicht für Verbindungen innerhalb der Ad-hoc-Umgebung relevant. Jingle dient in der Ad-hoc-Umgebung ausschließlich dazu, die Transportadressen geordnet über den authentifizierten und verschlüsselten XMPP-Kommunikationskanal auszutauschen. Der ursprüngliche Einsatzzweck, die Realisierung einer direkten P2P-Kommunikation zwischen den Knoten, wird in einem Ad-hoc-Netz aufgrund der fehlenden Server-Infrastruktur bereits durch das *Serverless Messaging* umgesetzt. Zum Adressaustausch wird in beiden Umgebungen einheitlich die Jingle-Erweiterung verwendet.

## 29.6 Sichere Kommunikation

Die Einhaltung von Sicherheitsanforderungen ist für viele kollaborative Anwendungen unerlässlich, da die mobile, drahtlose Kommunikation in den von uns betrachteten Szenarien besonders anfällig für böswillige Angriffe ist. Aus diesem Grund enthält die *uBeeMe*-Plattform eine Sicherheitskomponente, die optional aktiviert werden kann und Basisdienste wie Authentifizierung, Autorisierung, Verschlüsselung und Datenintegrität bereitstellt. Die Sicherheitsunterstützung fokussiert sich dabei besonders auf die Absicherung der Gruppenkommunikation und die zuverlässige Authentifizierung der Partner. Für letzteres wurde ein sicheres passwortbasiertes Authentifizierungsprotokoll entwickelt. Durch die passwortbasierte Authentifizierung soll insbesondere die Flexibilität der Kommunikationspartner unterstützt werden. Grundlage der vertraulichen Kommunikation ist die Nutzung eines gemeinsamen Gruppenschlüssels, dessen Verwaltung die Kernkomponente der *uBeeMe*-Sicherheitsarchitektur bildet. Entsprechend dem verteilten Charakter der *uBeeMe*-Plattform wird dafür ein verteiltes Schlüsselmanagement-Protokoll unter Nutzung von elliptischer Kurvenkryptografie eingesetzt, das speziell für kleine mobile kollaborative Gruppen ausgelegt ist. Es stellt eine Variante des VTKD-Protokolls [15] dar. Beide Protokolle können in dieser kurzen Übersicht zur *uBeeMe*-Plattform nicht hinreichend anschaulich beschrieben werden. Sie werden getrennt publiziert.

## 29.7 Entwicklung kollaborativer Apps

Mit der *uBeeMe*-Plattform lassen sich auf einfache Art und Weise kollaborative Anwendungen (Apps) entwickeln. Dies soll am Beispiel einer App für das bekannte Brettspiel „*Mensch ärgere dich nicht*“ erläutert werden. Die Anwendung setzt die bekannten Spielregeln um. Auch die Oberfläche (siehe Abb. 29.9) zeigt das vertraute Brettspiel. Sie wird synchron auf allen mobilen Endgeräten angezeigt. Daneben



Abb. 29.9 Oberfläche der „Mensch ärgere dich nicht“-App

können die Spieler während des Spielens miteinander chatten oder eine Audiokonversation führen. Netzwechsel von Spielern führen zu keiner Unterbrechung. Des Weiteren gibt es eine Kontaktliste, welche den Status der Kontakte anzeigt und es ermöglicht, diese zu einem Spiel einzuladen.

Alle *uBeeMe*-Komponenten sind als Module realisiert. Ein Modul kann Daten vom Netz empfangen bzw. an das Netz senden. Module können auch miteinander verkettet werden, wodurch die Daten eines Moduls beim Senden an das Nachfolger-Modul übergeben werden. Jedes Modul in der Kette kann die Daten verarbeiten oder erweitern. Am Ende einer solchen Kette steht stets ein UDP- oder TCP-Modul, das die Daten an das Netzinterface übergibt. Für die Realisierung des Spiels bietet es sich an, je ein Modul für die drei Funktionen der Anwendung (Chat, Audio-Konversation und Spiellogik) zu implementieren.

Abbildung 29.10 zeigt eine schematische Darstellung der Anwendung. Sie besteht aus mehreren Modulen. Alle dunkel dargestellten Module sind Teil der *uBeeMe*-Plattform und können ohne Anpassungen direkt verwendet werden. Die übrigen Module wurden speziell für das Spiel implementiert. Sie nutzen allerdings einige *uBeeMe*-Kernkomponenten, wie das Event-System und die Serialisierung, was die Entwicklung dieser Module erheblich vereinfacht.

Das *Chat*-Modul ist für das Versenden von Chat-Nachrichten verantwortlich. Wenn ein Spieler eine Nachricht eingibt, wird diese an das *Chat*-Modul übergeben. Das Modul nutzt die Serialisierungsfunktion der *uBeeMe*-Plattform, um den zu sendenden Text in ein Datenpaket zu verpacken. Beim Empfangen einer Nachricht wird diese vom *Chat*-Modul deserialisiert und an die grafische Oberfläche zur Ausgabe übergeben.

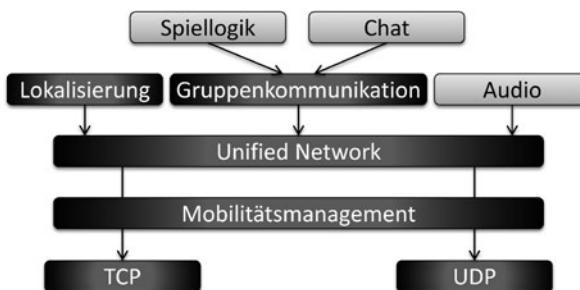


Abb. 29.10 Aufbau der „Mensch ärgere dich nicht“-App

Das *Spiellogik*-Modul übernimmt die Umsetzung der Spielregeln. Dafür implementiert dieses Modul ein Datenmodell, welches die Repräsentation aller Felder des Spielbretts sowie die Position der Spielfiguren enthält. Die *Spiellogik* wurde mittels eines Zustandsautomaten implementiert. Dadurch wird festgelegt, wann ein Spieler würfeln darf und wann eine Spielfigur bewegt wird. Sobald der Nutzer eine Aktion (z. B. Würfeln) ausführt, wird ein Zustandswechsel ausgelöst, bei dem eine *uBeeMe*-Funktion aufgerufen wird, um ein Datenpaket zu versenden. Das Empfangen einer Nachricht löst ebenfalls einen Zustandswechsel aus; die grafische Darstellung wird aktualisiert. Das *Spiellogik*-Modul kennt drei verschiedene Typen von Nachrichten um folgende Informationen zwischen den Spielern auszutauschen: Würfelergebnis, Ziehen einer Spielfigur, Wechsel des aktuellen Spielers. Die *uBeeMe*-Plattform bietet Mechanismen, um diese verschiedenen Nachrichtentypen zu definieren.

Für das Versenden der Chat- und *Spiellogik*-Nachrichten wird das Gruppenkommunikationsprotokoll *Moversight* eingesetzt, um eine geordnete und verlustfreie Übertragung der Nachrichten zu gewährleisten. Das *Audio*-Modul ist für das Aufnehmen und Wiedergeben der Audiokonversation sowie das Versenden der Audiodaten verantwortlich. Die Gruppenkommunikation wird für den Datentransport der Audiodaten nicht genutzt, da eine wiederholte Übertragung von Audiodaten bei auftretenden Paketverlusten nicht gewünscht ist. Für die Erstellung der Kontaktliste und die Bestimmung der IP-Adressen der Mitspieler wird das Lokalisierungsmodul eingesetzt. Der Netzzugang und etwaige Netzwechsel werden für die App transparent durch die Module *Unified Network* und *Mobilitätsmanagement* bereitgestellt. Sie wurden in den vorstehenden Abschnitten eingehend beschrieben.

## 29.8 Ausblick

Mobilität, Kooperation und ubiquitärer Netzzugang werden in den nächsten Jahrzehnten noch stärker als bisher die Entwicklung von Anwendungslösungen prägen. Dazu sind flexible und adaptive Lösungen gefragt, die bestmöglich kombiniert werden können, um Synergieeffekte zu erzielen. Die Bereitstellung von wichtigen dafür benötigten Funktionen in einer Plattform wie *uBeeMe* soll den Anwender von der

Entwicklung wichtiger Basiskomponenten entlasten. Durch ihren modularen Aufbau wird eine einfache Erweiterbarkeit, Aktualisierung und Pflege der Plattform unterstützt. Außerdem ist es Drittanbietern möglich, eigene Komponenten zu integrieren. Ein wesentlicher Aspekt der Entwicklung war der sorgfältige Umgang mit den begrenzten Ressourcen mobiler Endgeräte (bspw. Speicher, CPU, Energiereserven). Die Plattform besitzt keine zentralen Infrastrukturelemente. Dies ist dem Einsatz in infrastrukturlosen Ad-hoc-Szenarien geschuldet. Die *uBeeMe*-Plattform ist als Prototyp realisiert. Gegenwärtig steht die Portierung in relevante Betriebssystemumgebungen im Mittelpunkt. Parallel dazu wird an der Entwicklung von mobilen kollaborativen Anwendungen gearbeitet. Hier liegt der Fokus vor allem auf Anwendungen im Gesundheitswesen und in der Altenbetreuung.

## Literaturverzeichnis

1. Lokalisierungsdatenbank OpenBMAP <http://openbmap.org>, 05. Juli 2011
2. Lokalisierungsdatenbank SkyHook <http://www.skyhookwireless.com>, 05. Juli 2011
3. Marquez-Barja J, Calafate CT, Cano J-C, Manzoni P (2011) An Overview of Vertical Handover Techniques: Algorithms, Protocols and Tools. *Computer Communications* 34(8):985–997
4. Nicholson A J, Wolchok S, Noble B D (2010) Juggler: Virtual Networks for Fun and Profit. *IEEE Transactions on Mobile Computing* 9(1):31–43
5. Birman K, El Abbadi A, Dietrich WC, Joseph T, Raeuchle T (1985) An Overview of the ISIS Project. *IEEE Distributed Processing Technical Committee Newsletter*
6. Moser LE, Melliar-Smith PM, Agarwal DA, Budhia RK, Lingley-Papadopoulos CA (1996) Totem: a Fault-tolerant Multicast Group Communication System. *Communications of the ACM* 39(4):54–63
7. Zühlke M, König H (2002) GCP – a Group Communication Protocol for Supporting Closed Groups in the Internet. In: IFIP TC6 WG 6.7 7<sup>th</sup> International Conference, SMARTNET 2002, Kluwer Academic Publishers, pp 211–227
8. Banerjee S, Bhattacharjee B, Kommareddy C (2002) Scalable Application Layer Multicast. In: ACM SIGCOMM 2002, Pittsburgh, PA, USA, pp 205–217
9. Buyukkaya E, Abdallah M, Cavagna R, Hu SY (2008) Group Dual-Overlay State Management for P2P NVE. In: Proc. of the 14<sup>th</sup> IEEE International Conference on Parallel and Distributed Systems (ICPADS 2008), IEEE Computer Society, pp 817–822
10. Lamport L (1978) Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM* 21(7):558–565
11. Bless R, Hübsch C, Mies S, Waldhorst OP (2008) The Underlay Abstraction in the Spontaneous Virtual Networks (SpoVNet) architecture. In: Proc. 4<sup>th</sup> EuroNGI Conference on Next Generation Internet Networks (NGI 2008), Krakow, Poland, pp 115–122
12. Saint-Andre P (2004) Extensible Messaging and Presence Protocol (XMPP): Core. RFC 3920, IETF
13. Wu T (2000) The SRP Authentication and Key Exchange System. RFC 2945, IETF
14. Rosenberg J (2010) Interactive Connectivity Establishment (ICE): A protocol for network address translator (NAT) traversal for offer/answer protocols. RFC 5245, IETF
15. Liu F, Koenig H (2005) Secure and Efficient Key Distribution for Collaborate Applications. IEEE CollaborateCom 2005

---

# Kapitel 30

## Entwicklung mobiler Betriebssysteme im Lichte neuer Apps, veränderter Herausforderungen und der Virtualisierung

Uwe Baumgarten, Andreas Bernhofer und Robert Dörfel

**Zusammenfassung** In den letzten Jahren hat sich das Feld der Betriebssysteme für Smartphones und andere mobile Endgeräte deutlich verändert. Neue Technologien und erhebliche Fortschritte erlauben inzwischen, dass das mobile Internet wirklich zu leben beginnt und sich zunehmend Marktanteile sichert. Damit treten weitere Herausforderungen, wie die Integration von Webtechnologien, und der Wunsch nach Vereinfachungen mehr in den Vordergrund. Hier setzt das Papier an, indem die aktuellen Entwicklungen herausgearbeitet und Lösungskonzepte für die bestehenden Herausforderungen vorgestellt werden. Wichtige Teilbereiche sind dabei die Berücksichtigung der Programmiermodelle, der Einsatz ressourcenschonender Mechanismen und Strategien, wie dies zum Beispiel beim Powermanagement mobiler Betriebssysteme erforderlich ist, sowie der Einsatz von Virtualisierungs-techniken.

### 30.1 Einführung und Motivation

Innerhalb weniger Jahre hat das gesamte Feld des „mobile computing“ einen rasanten Wandel erfahren (siehe auch Schilit in [1]) und mobile Applikationen, kurz als mobile Apps bezeichnet, sind in aller Munde, laufen auf diversen mobilen End-

---

Uwe Baumgarten  
Technische Universität München, Fakultät für Informatik,  
E-mail: baumgaru@in.tum.de

Andreas Bernhofer  
Technische Universität München, Fakultät für Informatik,  
E-mail: andreas.bernhofer@tum.de

Robert Dörfel  
Technische Universität München, Fakultät für Informatik,  
E-mail: doerfel@in.tum.de

geräten und stifteten damit fortwährend und überall, also ubiquitär und „pervasive“, Nutzen in nahezu allen Lebensbereichen.

Wieso ist dies möglich geworden? Zuerst die technologischen Entwicklungen auf der Seite der Hardware der mobilen Endgeräte zu nennen. Die Prozessoren haben mit Taktraten bei einem GHz die Leistungsfähigkeit von PCs oder Laptops fast erreicht, erste Endgeräte mit Mehrkernprozessoren sind auf den Markt gekommen (siehe die Argumentation für die ARM-Dual-Core-Prozessoren in [2]) und die Bedienung erfolgt komfortabel und intuitiv über kapazitive Touchscreens. Zudem ist eine Konnektivität mit der Umwelt der mobilen Endgeräte sowohl im Weitverkehrs- bereich durch die Mobilfunknetze als auch im Nahbereich (u. a. WLAN, Bluetooth) erreicht, deren Datenraten inzwischen als nahe an der Festnetzerfahrung empfunden werden, und das bei Kosten, die sich in einem für viele Nutzende verträglichen Rahmen halten. Neben der Hardware hat sich aber auch die Softwareseite deutlich weiterentwickelt. Apps – und hier insbesondere die mobilen Apps – haben einen Reifegrad und eine Verbreitung erreicht, die der klassischen Programmierung zunehmend den Rang ablaufen. So lassen vielfältige Schnittstellen Sensoren jedweder Art leicht nutzbar werden. Dabei nehmen die Entwicklungswerkzeuge viel Arbeit ab und der Vertrieb und die Verbreitung von Apps hat über angepasste Geschäftsmodelle eine breite Akzeptanz gefunden. Die Konsequenz aller dieser Entwicklungen ist, dass der Markt der Apps im Konsumentenbereich in ihren App-Stores und Marktplätzen ohne Ende boomt.

Aber dieser Fortschritt ist nicht in allen Bereichen zu beobachten. Zu den sich etwas anders entwickelnden Feldern gehören drei Bereiche: die Betriebssysteme, der Lebenszyklus der Apps und ihr gesamtes Management sowie die Offenheit der Applikationslandschaft. Der Bereich der Betriebssysteme (kurz BS) ist einerseits sehr stark geprägt von den unterschiedlichen Ausrichtungen der treibenden Kräfte, also unter anderen den Unternehmen Apple, Google, HP, HTC, Microsoft, Nokia, RIM und Samsung, einige der wichtigen Vertreter, die allesamt mit Smartphones und deren mobilen Betriebssystemen erfolgreich sind. Im Wesentlichen setzen sie auf bewährte und ausgereifte BS-Technologien, versuchen aber, neue Entwicklungen, wie zum Beispiel die Virtualisierung, einzufangen. Details dazu sind im Abschn. 30.2, „Betriebssysteme“, zu finden. Am Markt resultiert dadurch eine Fülle unterschiedlicher Endgeräte mit einem knappen Dutzend verschiedenartiger und kaum austauschbarer Betriebssysteme. Auf der Seite der Applikationslandschaft steht schon am Anfang des Lebenszyklus eine große Herausforderung. Können Apps unabhängig von der jeweils eingesetzten Endgeräte-BS-Kombination entwickelt und auf allen gleichermaßen betrieben werden? Verständlicherweise versuchen viele Anbieter sich auf dem Markt über die Apps zu differenzieren. Auf der anderen Seite besteht der Wunsch nach einer Vereinheitlichung, der Integration und der Kooperation. So ist es für viele Unternehmen, die ihre Kunden mit ihren eigenen mobilen Apps versorgen möchten, ein beträchtlicher Zusatzaufwand, die Apps ihren gesamten Lebenszyklus für unterschiedliche Plattformen, die häufig auch noch unterschiedliche Technologien verwenden, vorzuhalten. Details dazu werden im Abschn. 30.3, „Applikationslandschaft“, aufgezeigt. Dort wird ebenfalls auf den dritten Bereich eingegangen,

der sich um mögliche Standards und Weiterentwicklungen der Applikationslandschaft kümmert.

Weitere Aspekte, auf die hier aber nur am Rande eingegangen werden soll, sind die Sicherheit nebst der Zuverlässigkeit, der Zugriff auf entfernte Daten und Dienste, wie er sich in den Entwicklungen rund um Cloud-Computing darstellt, und die Personalisierung der mobilen Endgeräte, von der die gesamte Entwicklung stark geprägt ist. Hier ist insbesondere die teils gemeinsame oder sogar gleichzeitige Nutzung der mobilen Endgeräte zu privaten und geschäftlichen Zwecken zu erwähnen.

## **30.2 Betriebssysteme für Smartphones, Herausforderungen und Stand der Technik**

Die Entwicklung der Betriebssysteme kann auf eine lange Tradition zurückblicken. Aber erst seit dem Übergang auf Smartphones gewinnen die dort entwickelten Konzepte auch zunehmend sichtbar für mobile Endgeräte an Bedeutung. Einige wesentliche und aktuelle Vertreter werden später kurz beschrieben. Zuvor werden aber noch einige wichtige Merkmale benannt, die für mobile Endgeräte und insbesondere Smartphones eine besondere Bedeutung haben. Dabei kommt der Aufgabe der Überwindung der Heterogenität in vielen Teilbereichen der Hardware für Smartphones eine besondere Bedeutung zu, auf die nachfolgend im Kontext der Virtualisierung näher eingegangen wird.

### **30.2.1 Stand der Technik**

Bevor der Stand der Technik für die derzeit aktuellen mobilen Betriebssysteme kurz aufgezeigt und eingeordnet wird, sollen einige Merkmale der mobilen Endgeräte und ihres Einsatzes benannt werden, da sie für die Entwicklung eine wichtige Rollen gespielt haben. Auf der Hardwareseite sind ganz klar die Vielfalt der eingesetzten Hardwaretechnologien sowie die Hardwarebausteine zu nennen (von den Prozessoren über die Speicher, die Ein-Ausgabegeräte bis hin zur Kommunikation). Alle Hardwareressourcen stellen sehr knappe Betriebsmittel dar, die sehr sorgfältig verwaltet werden müssen. Die jeweils zur Verfügung stehende Energie sowie das diese verwaltende Powermanagement ist ein prominentes Beispiel dafür. Ein sehr wichtiges Einsatzmerkmal ist, dass Smartphones in der Regel personalisierte Endgeräte sind, die zu einem sehr persönlichen, auf die jeweils Nutzenden zugeschnittenen Einsatz führen. Entstanden ist dies einerseits durch die Verbreitung der Mobiltelefonie, die mit den Handys einen sehr individuellen Bezug zu den Nutzenden hergestellt hat, und andererseits durch die Bereitstellung von Funktionen wie MP3-Player und Digitalkamera, welche das Smartphone zum alltäglichen persönlichen Begleiter werden lassen. Die eingesetzte SIM-Karte ist die Basis für die Individualisierung

und aufbauend darauf für die Personalisierung. Neben dem Konsumentenbereich hat aber auch das Geschäftsumfeld seine Tradition, die in schon lange eingesetzten Organisern begründet ist, bei denen Termine, kundenbezogene Daten und E-Mails tragende Säulen waren.

Alle diese Merkmale sind in den heutigen Smartphones mit ihren mobilen Betriebssystemen und darauf laufenden Apps vereinigt. Die hier betrachteten Vertreter von Smartphones (nebst Betriebssystem) sind diejenigen von Apple (iOS), Google (Android/Linux), HP (WebOS), Microsoft (Windows Mobile, Windows Phone 7), Nokia (Symbian; MeeGo), RIM (BlackberryOS) und Samsung (Bada). Auf weitere Anwendungsdomänen, wie etwas Realzeitsysteme, Consumer Electronics im Unterhaltungssektor oder die Automotive-Domäne wird nicht weiter eingegangen.

Für alle diese Betriebssysteme gilt, dass die jeweiligen Betriebssystemkerne die grundlegenden Konzepte und Abstraktionen zur Verfügung stellen, einschließlich der Verwaltung der Betriebsmittel. Viele der oben genannten Systeme basieren auf Linux- oder ähnlichen Kernen. Wichtige Unterschiede bestehen darin, welche Funktionen oder Dienste des Betriebssystems den Anwendungen zur Verfügung gestellt werden sollen und wie dies geschehen soll (zum Beispiel durch Ausführung im System- oder im User-Mode). Entscheidungen hierfür liegen in Bereichen wie dem Multiprogramming, der Verwendung von Hintergrundprozessen, der Verwendung synchroner Kommunikation bzw. asynchroner Ereignisse oder dem Zugriff auf Hardwareschnittstellen. Hier sind beim Smartphone insbesondere Lokalisierungskomponenten (GPS, Kompass), die Aufnahmemöglichkeiten (Kamera, Mikrophon) oder der lokale persistente Speicher (z. B. als Dateisystem) zu nennen.

Weitere Unterschiede zeigen sich deutlich bei der Vorbereitung für den Einsatz zur Ausführung von Apps durch die Middleware und die dazu verwendeten Frameworks. Hier spielen unter anderen die Programmiermodelle mit der Verwendung von Nebenläufigkeit wichtige Rollen.

Die aktuellen Herausforderungen werden derzeit aus zwei Richtungen angegangen. Einerseits wird verstärkt auf die Ideen und Erfahrungen mit der Virtualisierung zurückgegriffen und andererseits wird zunehmend die Gesamtsicht von Smartphones als Plattformen zur Ausführung von Apps unter die Lupe genommen. In beiden Richtungen werden Wege eröffnet, um die Heterogenität auf allen Ebenen (von der Hardware über die Betriebssysteme bis zu den App-Programmier-Frameworks) zu überwinden und eine Integration und Interaktion in weltweit vernetzten und verteilten Systemen angemessen unterstützen und nutzen zu können. Auf die Grundlagen der Virtualisierung wird in Abschn. 30.2.2, „Virtualisierung“, eingegangen. Der Plattformgedanke wird in Abschn. 30.3, „Applikationslandschaft“, aufgegriffen.

### **30.2.2 Virtualisierung**

Die Virtualisierung ist eine Technologie, die schon lange bekannt ist und zum Beispiel seit vielen Jahrzehnten im Bereich der Großrechner als Virtual Machine Mo-

nitor (VMM) zum Einsatz kommt. Damit können die Ziele der Unabhängigkeit von der jeweiligen Hardware (Standard für Großrechner), der Unabhängigkeit von anderen virtuellen (oder realen) Maschinen untereinander (Nutzung für die Sicherheit) oder die Ortsungebundenheit (Nutzung für verteilte Systeme) erreicht werden. Grundlagen zu den virtuellen Maschinen sind u. a. in [3] zu finden. In den letzten Jahren wurde der Einsatz der Virtualisierung wieder verstärkt untersucht und differenziert, so dass einerseits die Systemvirtualisierung (hier werden Maschinen als Ganzes virtualisiert) und andererseits die Prozessvirtualisierung (hier wird eine Ausführungsumgebung virtualisiert) entstanden sind (siehe dazu [4]). Dabei finden die Virtualisierungstechniken auch im Bereich der PCs Anwendung mit weitgehend ähnlichen Zielen, teils aber mit zusätzlicher spezieller Hardwareunterstützung auf der Seite der eingesetzten Prozessoren. Selbstverständlich sind diese Technologien auch für eingebettete Systeme sehr nützlich, auch wenn sich die Ziele wegen der knappen Ressourcen durchaus verändert haben. Dies wird zum Beispiel an den Entwicklungen um die Hypervisor und den Einsatz des OKL4-Mikrokerns erkennbar (siehe [5]). Hier wird verstärkt den Anforderungen der (mobilen) Endgeräte im Bereich der Consumer Electronics Rechnung getragen – namentlich der Berücksichtigung knapper Ressourcen für die Kommunikation, dem Speicher (kleiner Fußabdruck), der Realzeitfähigkeit, aber auch den sehr differenzierten Konzepten der IT-Sicherheit. Hinzu kommt die häufig knappste Ressource, die Energie. Alle diese Anforderungen finden sich in heutigen Smartphones wieder. Die aktuellen Entwicklungen der Open Kernel Labs (siehe allgemein in [6]) gehen an dieser Stelle sogar noch etwas weiter, indem sie mit dem OKL4 Microvisor die Verschmelzung von Hypervisor und Microkernel vorstellen (siehe [7]), so dass eine sehr mächtige Basistechnologie entsteht, die für den Bereich der mobilen Endgeräte sehr nützlich ist.

Die Virtualisierung kann aber für Smartphones auch noch anders eingesetzt werden. In ihrem Ansatz „Virtual Smartphone over IP“ versuchen Chen und Itoh, das Problem der knappen Ressourcen (hier insbesondere der Rechenfähigkeit – also der Prozessorleistung) dadurch zu entschärfen, dass entfernte virtuelle Instanzen des jeweiligen mobilen Endgerätes Berechnungen auf leistungsfähigen Server-Rechnern ausführen und damit das mobile Endgerät deutlich entlasten. Bei der Implementierung kommt dann zudem die Technik der Prozessvirtualisierung zum Einsatz, da die Applikationen sowohl lokal im Smartphone als auch entfernt auf einem Server als Android-Applikationen auf der Basis der Dalvik-VM realisiert sind.

Durch den Einsatz der Virtualisierungstechnik kann die Heterogenität zu weiten Teilen überwunden werden. Dieser Vorteil kann aber schnell geschmälert werden, wenn durch die Weiterentwicklungen der virtuellen Maschinen, der SDKs oder der Bibliotheken in verschiedene Richtungen viele Varianten oder Versionen entstehen, die nicht mehr unmittelbar zueinander kompatibel sind. Dies ist zum Beispiel in den SDKs für Android erkennbar. Hier besteht die Gefahr, dass sich die Fragmentierung des Marktes entlang unterschiedlicher Versionen zu einem andauernden Handicap entwickelt.

### 30.3 Applikationslandschaft – Aktuelle Entwicklungen und Einsatzfelder

Mit der starken, kontinuierlichen Weiterverbreitung von Smartphones haben Apps als kleinere oder größere Programme eine Bedeutung und Aufmerksamkeit erlangt, die bis dato für „klassische“ Programme und die Softwareentwicklung eher undenkbar war. Demzufolge gewinnen die Applikationslandschaften, die Lebenszyklen von Apps und die Werkzeuge zu deren Erstellung und Vertrieb immer stärker an Bedeutung.

Für den Einsatz und die Nutzung von AppStores und Marktplätzen war sicherlich Apple der Vorreiter, der frühzeitig (zahlende) Kunden an einen Ort versammelt hat und diese einheitlich – sowohl im Hinblick auf die technischen als auch die geschäftlichen Aspekte – bedienen konnte. Andere Anbieter folgten diesem Modell. Aber es haben sich auch Alternativen (zum Beispiel für Android) ergeben.

Neben dem verständlichen Versuch der Anbieter, sich über die Apps von den Mitbewerbern zu unterscheiden, besteht zunehmend der Wunsch nach Integration und Interoperabilität. Eine App nur ein Mal entwerfen und programmieren zu können bzw. zu müssen, so dass sie auf allen Plattformen zum Einsatz und zur Nutzung kommen kann, erweist sich als sehr verständlicher Wunsch. Die Wege dorthin sind sehr vielfältig, stoßen aber schnell an Grenzen, die durch die Marktteilnehmer und deren Unternehmensstrategie vorgegeben werden bzw. durch die darunterliegenden Plattformen mit spezialisierter Hardware, eingeschränkten Schnittstellen zum Betriebssystem oder speziellen Programmiermodellen gegeben sind.

Die aktuellen Entwicklungen lassen Folgendes erkennen. Für den Vertrieb von Apps haben sich AppStores und Marktplätze als sehr erfolgreicher Vertriebskanal etabliert. Sie bilden damit einen Teil des Cloud-Computing, bei dessen Bezahlmodellen, den Sicherheitsarchitekturen und der Vertrauensbildung allerdings noch deutlicher Diskussions- und Lösungsbedarf besteht.

Für die App-Entwicklung und für den ganzen Lebenszyklus stellt sich die Situation anders dar. Hier setzt sich bei vielen Anbietern die Heterogenität der Plattformen (also der Endgeräte und insbesondere ihrer Betriebssysteme) fort. Ein guter Überblick dafür ist in [8] zu finden. Wichtige Merkmale sind die verwendete Programmiersprache, die Reichhaltigkeit der verwendbaren BS-Schnittstellen, die Einsetzbarkeit von Webtechnologien und die Quelloffenheit.

Daraus resultiert unmittelbar, dass die Apps in solche zu unterscheiden sind, die auf die unterliegende Plattform (in Programmiersprache, Betriebssystem und Schnittstellen) zugeschnitten sind und solche, die davon nicht abhängig sind. Erstere sollen als native Apps bezeichnet werden. Für letztere, also die unabhängigen Apps, bieten sich seit geraumer Zeit die Webtechnologien an, die als WebApps bezeichnet werden. Hier wird durch den jeweiligen Browser eine weitere Virtualisierungsschicht eingezogen, so dass für die WebApps als Ausführungsumgebung nur der Browser zur Verfügung steht. Ein Zugriff auf das Betriebssystem und das mobile Endgerät ist damit unmittelbar nicht möglich. Soweit betrachtet sind die nativen

Apps den WebApps an Nutzungsmöglichkeiten deutlich überlegen und erlauben damit ein größeres Spektrum von Apps. Zudem sind die programmiersprachenspezifischen Laufzeitsysteme häufig in der Lage, die Apps performanter auszuführen, was insbesondere bei Multimediaanwendungen nicht zu vernachlässigen ist. Um diesen Nachteil der WebApps zu mildern, erlauben es die Weiterentwicklungen rund um HTML5, die meistens in Java Script geschriebenen WebApps um viele Eigenschaften und Zugriffsmöglichkeiten anzureichern, so dass eine den nativen Apps ähnliche Vielfalt entsteht. Diese Weiterentwicklungen und Standardisierungen werden stark von W3C vorangetrieben (siehe [9]), um eine gleichgeartete Implementierung in möglichst vielen Browser-Engines zu erreichen.

Für die oben genannten Systeme bedeutet dies exemplarisch das Folgende: Native Apps für iOS werden in Objective-C geschrieben und können eine eingeschränkte API nutzen. Den WebApps im iOS-Browser steht HTML5 zur Verfügung (Webkit-Engine). Die Plattform ist nicht quelloffen.

Im Gegensatz dazu werden native Apps für Android in Java geschrieben und nutzen als Ausführungsumgebung eine virtuelle Maschine (Dalvik). Sie können aber auch auf native Eigenschaften über Java oder C zugreifen. Die Plattform ist quelloffen. Browser können auch hier HTML5 einsetzen.

Ebenfalls quelloffen sind die Symbian- und MeeGo-Plattformen, bei denen die nativen Apps in C/C++ programmiert werden. Die Browser können ebenso HTML5 einsetzen.

Wie sich aus der Aufzählung ablesen lässt, sind neben nativen Apps in den meisten Fällen auch WebApps möglich. Damit stellt sich die Frage, welcher Art der Apps die Zukunft gehören wird. Diese Frage wird derzeit intensiv diskutiert. Hier lässt sich auch ein Bezug zurück zu den Betriebssystemen bilden. Wenn auf einem (mobilen) Endgerät nur noch WebApps zur Ausführung kommen, dann kann der Browser die gesamte Betriebsmittelverwaltung übernehmen, so dass von einem Webbetriebssystem gesprochen werden könnte. Derartige Wege sind u.a. bei Chromium OS zu finden. Argumente, die die browsergestützten Entwicklungen in einem erweiterten Sinne stärken, sind auch im Umfeld des „Internets der Dinge“ zu finden (siehe [10]).

Im nächsten Abschnitt werden mögliche Lösungsansätze für Apps, native Apps und WebApps skizziert.

## 30.4 Exemplarische Lösungsansätze und -wege

In diesem Abschnitt sollen zwei Ansätze kurz vorgestellt werden, die in unserer Forschungsgruppe durchgeführt wurden bzw. werden und bei denen das einfache Entwerfen und Umsetzen von Apps wichtig ist. Es handelt sich bei dem ersten Ansatz um eine möglichst einfache mobile App, um insbesondere Studierenden Zugang zu ihren Lehrveranstaltungen, Prüfungen und Ergebnissen zu geben. Als zweiter Ansatz wird das EU-Großprojekt webinos angesprochen, in dessen Rahmen die Idee des „develop once, run anywhere“ umgesetzt werden soll.

### ***30.4.1 App für das Campusmanagement***

Auch die Studierenden setzen in zunehmendem Maße Smartphones für ihre studentischen Angelegenheiten ein. Aus dem Wunsch heraus, sowohl auf Lehrveranstaltungen, deren Termin, aber auch auf Prüfungen und deren Ergebnisse zugreifen zu können, entstand ein Erprobungsprojekt, durch das die Machbarkeit einer App für zwei Plattformen (iPhone und Android) gezeigt werden sollte (siehe [11]). In diesem Rahmen sollten die Webservices eines bestehenden Systems für das Campusmanagement sowie die personenbezogene individuelle Nutzung von Diensten demonstriert werden. Dazu wurde ein hybrider Ansatz gewählt, bei dem die Apps sowohl native als auch Webbestandteile aufwiesen.

Funktionalität und Benutzungsoberfläche der Anwendung wurden vollständig mittels HTML5 und JavaScript als WebApp entwickelt. Dies ermöglicht die nahezu aufwandslose Portierung der Anwendung auf kompatible Smartphones (hier konkret Android- und iOS-Geräte) und die einfache Nutzung über ein Lesezeichen. Um die Anwendung wie eine native App erscheinen zu lassen und zusätzlich eine Verbreitung über den AppStore bzw. den Android Market zu ermöglichen, wurde die WebApp in eine jeweils systemabhängige, aber dafür sehr schlanke, native Anwendung gekapselt, welche im Wesentlichen nur aus einer Browser-Komponente zur Anzeige der WebApp besteht.

Durch diese geschickte Kombination kann die WebApp alle Vorteile einer native Anwendung nutzen (z. B. Offline-Fähigkeit und Auswertung der Hardwaredataten des Gerätes) und ist von einer solchen für den Anwender auch nicht mehr zu unterscheiden.

### ***30.4.2 webinos***

Zu den wichtigsten Zielen des EU-Großprojektes webinos gehört es, die App-Entwicklung durch Nutzung von Webtechnologien zu vereinheitlichen und die so entstehenden Apps auf einem breiten, viele Domänen umfassenden Spektrum von Endgeräten zur Ausführung zu bringen (siehe [12]). Neben Smartphones, PCs und Tablets gehören gerade auch Fernsehgeräte und Autos zu den mit einbezogenen Endgeräten. Aus dem Titel „Secure Web Operating System Application Delivery Environment“ lässt sich die Architektur ablesen (siehe [13]). Die zu entwickelnden WebApps werden aus einer webinos-Cloud bezogen und auf den Endgeräten in einer browsergestützten Umgebung ausgeführt. Alle einbezogenen Endgeräte und Systeme folgen einer webinos-Sicherheitsarchitektur, die sowohl den Vertriebskanal als auch die Ausführungsumgebung sowie den Zugriff auf lokale und entfernte Komponenten und Dienste steuert und überwacht. Das webinos-Konsortium ist mit über 20 spezialisierten Partnern sehr breit aufgestellt, wobei aus Sicht der TUM die Umsetzung in der Automotive-Domäne im Vordergrund steht.

## 30.5 Ausblick

Nachdem die Betriebssysteme in ihrer langen Entwicklungsgeschichte einen – konzeptionell gesehen – hohen Reifegrad erreicht haben, bringen Smartphones, mobile Apps und das Cloud-Computing wieder deutlich mehr Leben und Fahrt in die Entwicklungen. Altbewährtes wird weiter seinen Nutzen finden, aber die deutlich gestiegene Leistungsfähigkeit der mobilen Endgeräte sowie die nahezu flächendeckenden, leistungsstarken und kontinuierlichen Kommunikationsmöglichkeiten lassen ein verteiltes und vernetztes Leben und Arbeiten zu, dessen Ausmaß so bisher nicht realisiert war. Apps werden den Zugang jedweder Information liefern, sei es in Datenbanken oder durch Ablesen von Sensoren. Web-Technologien werden dabei eine vereinfachende und integrierende Rolle spielen. Zudem werden immer weitere Anwendungsdomänen integriert und durch Smartphones zugänglich gemacht, sei es im Automobilbereich (siehe auch [14]), im Verkehrsmanagement, bei der Energieversorgung oder ganz allgemein bei Geschäftsprozessen in allen Lebensbereichen.

## 30.6 Danksagung

Danke an die Mitarbeiter am Lehrstuhl Informatik I13 der TU München.

## Literaturverzeichnis

1. Schilit BN (2011) Mobile Computing: Looking to the Future. *IEEE Computer* 28–29
2. Paulson LD (2011) Dual-Core Processors to Improve Smartphone Performcane. *IEEE Computer* p 16
3. Siegert H-J, Baumgarten U (2006) Betriebssysteme, eine Einführung, 6. Aufl, Oldenbourg Verlag
4. Smith JE, Nair R (2005) The architecture of virtual machines. *IEEE Computer* 32–38
5. Heiser G (2009) Hypervisors for Consumer Electronics. CCNC 2009 Consumer Communications and Networking Conference, IEEE
6. Open Kernel Labs (2011) OKL. <http://www.ok-labs.com/>. Zugegriffen Juni 2011
7. Heiser G, Leslie B (2010) The OKL4 Micorvisor: Convergence Point of Microkernels and Hypervisors. APSys10 First ACM Asia-Pacific Workshop on Systems
8. Tarkoma S, Lagerspetz E (2011) Arching over the Mobile Computing Chasm: Platforms and Runtimes. *IEEE Computer* pp 22–28
9. World Wide Web Consortium W3C (2011) HTML5. <http://www.w3.org/TR/html5/>. Zugegriffen Juni 2011
10. Garcia-Macias JA et al. (2011) Browsing the Internet of Things with Sentient Visors. *IEEE Computer* 46–52
11. Bernhofer A (2011) Konzept und Entwicklung einer plattformunabhängigen Webservice-basierten mobilen Anwendung für CAMPUSonline. Diplomarbeit TU München
12. webinos (2011) webinos Secure Web Operating System Application Delivery Environment. <http://webinos.org>. Zugegriffen Juni 2011
13. webinos (2011) webinos whitepaper. [http://webinos.org/wp-content/uploads/2011/06/-webinos\\_whitepaper.pdf](http://webinos.org/wp-content/uploads/2011/06/-webinos_whitepaper.pdf). Zugegriffen Juni 2011

14. Bose R (2011) Morphing Smartphones into Automotive Application Platforms. IEEE Computer pp 53–61
15. Chen EY, Itoh M (2010) WoWMoM 2010 World of Wireless Mobile and Multimedia Networks, IEEE

---

# Kapitel 31

## Mit Java ins Internet der Dinge aufbrechen

Rainer Eschrich und Guido Burger

**Zusammenfassung** Dieses Kapitel befasst sich mit den Rahmenbedingungen für mobile Anwendungen, Plattformen und Infrastrukturen, die durch den revolutionären Wandel zum „Internet der Dinge“ hervorgerufen werden. Er beschäftigt sich mit der Frage, wie die rasante technologische Entwicklung stetig neue Generationen mobiler Anwendungen hervorbringt und welche Rolle das „Internet der Dinge“ dabei spielt. Fragen der Sicherheit stehen dabei genauso im Mittelpunkt, wie die Möglichkeit, neuartige und innovative Geschäftsmodelle zu entwickeln.

Welche Auswirkungen zieht diese anstehende Transformation nach sich und auf welche Art verändern sich dadurch Geschäftsmodelle und Zielgruppen? Was wird uns das Internet der Dinge bringen? Welche Möglichkeiten erwachsen daraus für unsere Gesellschaft? Welches sind die zentralen technologischen Anforderungen eines „Internet der Dinge“ und welche IT-Service-Infrastruktur ist dazu erforderlich? Wie kann eine flexible und in jeder Hinsicht agile IT-Service-Architektur aufgebaut werden? Worauf gilt es zu achten? Welche Rolle kann Java in einem „Internet der Dinge“ spielen? Und wie könnte ein Ökosystem für mobile Dienste der nächsten Generation aussehen? All diese Fragen werden in dem folgenden Beitrag beleuchtet.

---

Rainer Eschrich  
Oracle Inc., Java Embedded Global Sales Unit, München,  
E-mail: rainer.eschrich@oracle.com

Guido Burger  
Oracle, Stuttgart,  
E-mail: guido.burger@oracle.com

## 31.1 Einführung

*Oft ist die Zukunft schon da, ehe wir ihr gewachsen sind.*

John Steinbeck

In wenigen Jahren haben sich intelligente Handys und darauf laufende Anwendungen in unser Leben integriert. Es ist normal für uns, über Apps und App-Stores zu reden. Apps sind die Markenzeichen der zweiten Generation mobiler Anwendungen. Ein Phänomen tritt dabei in den Hintergrund, das unsere Welt grundlegend verändert: Es kommen täglich neue Geräte auf den Markt, die eine eigene Computer-Intelligenz besitzen und nach einer (mobilen) Netzwerkanbindung verlangen. Das Zeitalter des „Internet der Dinge“ ist angebrochen, ohne dass wir für diese dritte Generation von mobilen Diensten bereit sind. In den kommenden Jahren werden die Karten neu gemischt und verteilt. Sie werden über Sieger und Verlierer in diesem neuen Spiel entscheiden.

## 31.2 Technologischer Fortschritt

### führt zu neuen mobilen Anwendungen und Diensten

*Wenn ich die Menschen gefragt hätte, was sie wollen, hätten sie gesagt, schnellere Pferde.*

Henry Ford

Um zu verstehen, wie diese Karten verteilt werden, muss man die technologischen Rahmenbedingungen und die kommerziellen Markttreiber kennen, die hinter dieser Veränderung stecken.

#### 31.2.1 Generation 1: Digitale Anwendungen für das Handy

Durch die Umstellung des mobilen Telefonnetzes von „analog“ auf „digital“ konnten Daten genauso einfach verteilt werden wie Audiosignale. Damit eröffnete sich für die Netzwerkbetreiber die Möglichkeit, neben Telefongesprächen weitere Geschäftsmodelle zu erschließen. Ermöglicht wurde dies auf der Endgeräte-Seite durch eine neue Generation von Mobiltelefonen. Plötzlich wurden neben den reinen Sprachdiensten auch digitale Dienste angeboten. Den überwiegenden Teil dieser Dienste stellten SMS, Klingeltöne und mobile Spielemärkte, die Milliarden von Euros jährlich umsetzten. Neben den Netzwerkbetreibern gehören die Anbieter des „Feature-Phone“ zu den Gewinnern. Java ME (die Java Mobile Edition) entwickelten sich zu der dominierenden systemübergreifenden Plattform für diese Dienste. Für hardwarenahe Funktionen und Erweiterungen wurden zusätzlich die nativen gerätespezifischen Entwicklungsumgebungen der einzelnen Hersteller verwendet. Die Integration in andere Geschäftsfelder und Backend-Systeme blieb größtenteils aus.

### ***31.2.2 Generation 2: Apps für „Smart Phones“ und andere „Smart Devices“***

Die zweite Generation von mobilen Geräten wurde durch die vollständige Verfügbarkeit der digitalisierten Inhalte – beispielsweise Musik, Videos, Bücher, und Landkarten usw. – vorangetrieben. Die Anwendungen (oder Apps) sind zu den Treibern für diesen Markt geworden. Dabei hat vor allem das Bereitstellen von Online-verkaufsplattformen für Apps sogenannten Appstores in kürzester Zeit ermöglicht, eine große Anwendungsbasis zu erreichen.

Diese neuen Plattformen bewirken zwei Marktveränderungen. Erstens: Klassische Zugangsgeräte wie Desktops, PC, Laptop und vor allem Netbooks werden weniger vom Markt nachgefragt. Zweitens: Die Nachfrage nach günstigen und dynamisch verfügbaren Ressourcen (sogenannten Cloud-Diensten) steigt und hierbei auch die Notwendigkeit, diese Dienste in die Backend-Anwendungen zu transformieren.

Neben der reinen Umsetzung von klassischen Web- oder Desktop-Anwendungen für die neue Welt tritt auch eine neue Art von Programmen auf den Plan. Diese verknüpfen online verfügbare Informationen mit anderen Daten, beispielsweise sensorischen Informationen (Licht, Temperatur, Neigung, etc.), Aufenthaltsorten oder Geschwindigkeiten. Diese neuen „intelligenten“ Anwendungen sind bereits Vorboten der nächsten Generation.

Schon heute ist abzusehen, dass die Vielzahl von verfügbaren Plattformen die Entwicklung von systemübergreifenden Lösungen sehr kostspielig machen wird. Der Grund ist, dass die Anwender bei ihrer Auswahl von Hard- und Software nicht beachteten, dass diese auch mit anderen IT-Systemen kommunizieren müssen. Dies erfordert die Unterstützung offener Standards und Möglichkeiten der Interoperabilität über Grenzen hinweg.

### ***31.2.3 Generation 3: Mobile Anwendungen für das „Internet der Dinge“***

Das Internet der Dinge (1999 von Kevin Ashton angekündigt [1]) meint das Internet zu dem Zeitpunkt, wo die „Dinge“ oder auch Gegenstände unserer Umwelt in ein Netzwerk eingebunden werden. Dank der großen technologischen Fortschritte des letzten Jahrzehnts sind wir heute an dem Punkt, wo dies Realität werden kann. Die größte Veränderung gegenüber dem klassischen Internet ist nicht nur die steigende Zahl neuer Endgeräte und die Netzwerkanbindung zu jeder Zeit. Wirklich neu ist die Möglichkeit, heute alles – und zwar innerhalb der gesetzlichen Rahmenbedingungen und des Datenschutzes (Apps, Dinge, Geräte, Sensoren, Backend, etc.) *miteinander* zu verbinden. Damit wird die Kommunikation unter den Geräten eine vergleichbare Bedeutung haben wie die Verbindung mit den Datenzentren. Durch intelligente Software in den Geräten werden diese zu lokalen Verteilern (sogenannten „Smart-Hubs“) der Kommunikation und des Datenflusses. „Smarte Anwendungen“ arbeiten

verteilt über unterschiedliche Geräte und Dinge. Sie erfassen multiple Sensorik und Daten und erzeugen daraus neuartige Informationen und Dienste.

### 31.3 Beispiele für das Internet der Dinge

Beispiele für das Internet der Dinge gibt es seit geraumer Zeit. Die meisten Systeme kommen bisher in der Forschung zum Einsatz. Dennoch sind einige interessante Beispiele nachfolgend aufgeführt:

Ein historisches Beispiel für das „Internet der Dinge“ ist das Mars-Rover-Projekt. Als die Mars Rover im Jahre 2004 auf die Reise zum Mars geschickt wurden, wurden schon damals wichtige Konzepte, wie autonomes und sicheres Fahren, elektrischer Fahrzeugantrieb, vernetzte Fahrzeuge und eine dynamisch anpassbare Plattform eingesetzt. Die Rove beweisen bis heute weit über ihrer erwartete Lebensdauer von drei Monaten hinaus den Erfolg dieses Java-basierten Systems [2, 3].

Das WikiCity-Projekt Rom vom MIT aus dem Jahre 2007 ist möglicherweise das erste großflächige Beispiel für das „Internet der Dinge“. In Rom (Italien) wurden Sensoren, Dienste und Wissen miteinander vernetzt. Einwohner und Reisende konnten in Echtzeit erleben, wie die Stadt auf ihre Bedürfnisse reagiert und intelligente Antworten liefert [4].

Ein weiteres Beispiel aus der Forschung ist die Verwendung eines Funksensornetzwerks zusammen mit einem Gateway, um in Echtzeit Veränderungen von Gletschern aufzuzeichnen und damit Rückschlüsse auf die Klimaveränderungen zu ziehen [5].

### 31.4 Wie das „Internet der Dinge“ die Karten neu mischt

*Die Rechenautomaten haben etwas von den Zauberern im Märchen. Sie geben einem wohl, was man sich wünscht, doch sagen sie einem nicht, was man sich wünschen soll.*

Norbert Wiener

Das „Internet der Dinge“ lässt sich am besten begreifen, wenn man es als eine Kombination der Vernetzung der klassischen IT-Welt mit Dingen der klassischen elektromechanischen Welt versteht. In dieser vereinigten Welt treffen die etablierten Spieler der IT auf die Spieler der konventionellen Märkte, wie die Automobilindustrie, das Gesundheitswesen, die Energieversorgung usw. Daraus ergeben sich neue Chancen für beide Seiten. Im Mittelpunkt der nächsten Generation steht dabei der „Smart-Hub“. Der Wettkampf darum hat bereits begonnen. Denn es ist abzusehen, dass Wissens-Vorsprung zu Wettbewerbsvorteilen führen kann.

Dabei geht es weniger um die Technologie oder um die Anwendungen. Es geht darum, die Aufmerksamkeit des Verbrauchers auf sich zu ziehen und ihm den größtmöglichen Nutzen zur Verfügung zu stellen. Wer im Auge des Kunden im Mittelpunkt steht, wird den höchsten Nutzen ziehen können und die besten Absatzmöglichkeiten.

lichkeiten haben. Vor allem die klassischen Industrien von deren Erfolg momentan Deutschland stark profitiert werden sich dieser Herausforderung stärker als bisher stellen müssen.

### ***31.4.1 Aufmerksamkeit – Die Marke neu definiert***

Um in diesem Markt bestehen zu können, werden klassische Hersteller gezwungen sein, ihre Marken neu zu definieren und ihre Strategien auf „Multi-Channel“ auszurichten. Eine Webseite reicht nicht mehr, um eventuell eine App oder ein Spiel anzubieten. Das Web 2.0 hat eine grundlegende Veränderung bewirkt [6]. Eine *neue* Marke muss mit dem Kunden in einen ständigen Dialog treten und ihm vor allem zuhören. Die Marke muss ein Zugehörigkeitsgefühl beim Käufer auslösen und Teil des sozialen Netzes werden. Im „Internet der Dinge“ ist der Kunde ständig im Mittelpunkt einer umfassenden Dienstleistungswelt. Hier wird die Marke persönlicher Freund und Ratgeber. Dafür müssen Marken weit über das klassische „Branding“ hinaus zu einer eigenen Persönlichkeit entwickelt werden.

### ***31.4.2 Fähigkeit – Die Kernkompetenz neu definiert***

Die integrierten Produkte der Zukunft werden mehr als Dienstleistungen denn als Objekte wahrgenommen. Immer stärker rücken bei Verbraucherentscheidungen immaterielle und soziale Eigenschaften in den Mittelpunkt [7]. Im Fokus stehen dabei Fragen wie zum Beispiel: Wie hilft mir ein Produkt, meinen Alltag zu verbessern, mein Umfeld neu zu gestalten oder auch eventuell meine Persönlichkeit zu entwickeln? Nur die Hersteller, die es schaffen, dem Anspruch ihrer Kunden gerecht zu werden, haben eine Chance, von den Veränderungen zu profitieren. Ein Beispiel: Eine Personenwaage mit eingebautem Netzwerk gewinnt allein noch keine neuen Märkte. Sie benötigt einen Service, eventuell eine Verknüpfung mit dem Blutzuckermessgerät oder dem Diätplan. Risikoreiche Veränderungen im Gesundheitszustand der Person werden sofort angezeigt, und der Nutzer kann auf dieser Basis seine Entscheidungen treffen (z. B. zum Arzt zu gehen oder auch den Notruf zu alarmieren). Diese neue Welt der integrierten Dienste erfordert eine große Kompetenz in Softwareentwicklung, Netzwerktechnologie und System-Modellierung.

### ***31.4.3 Zielgruppen neu definiert***

Die digitale Kompetenz aller Altersgruppen hat in den letzten Jahrzehnten dramatisch zugenommen; so haben heute in Deutschland mehr als 75 % der Haushalte Zugriff zum Internet (laut OECD 2009). Die digitale Kompetenz hat grundlegende

Auswirkungen auf den Auswahlprozess bei der Anschaffung eines Produktes. Jeder Käufer sucht sich seinen persönlichen Erfahrungsraum, um sich ein Urteil zu bilden und eine Entscheidung zu treffen. Dabei werden mehr Datenquellen, mehr Beurteilungen Dritter in Foren und mehr Informationen gesucht und in die eigentliche Kaufentscheidung einbezogen [8].

Die Zielgruppen lassen sich deshalb heute nicht mehr so gut durch die klassischen Kriterien (wie Alter, Geschlecht, soziale Herkunft etc.) definieren, sondern eher durch individuelle (Internet)-Verhaltensmuster. Aus diesem Grund ist es wichtig, das digitale Leben der Anwender zu berücksichtigen und eine digitale Kompetenz stetig weiter zu entwickeln. Interaktion mit dem Kunden zu jeder Zeit und über unterschiedliche Kanäle scheint inzwischen selbstverständlich. Ebenso sind Service und Lieferung an jeden Ort und die Bereitstellung unterschiedlichster Bezahlmethoden zwingend erforderlich.

### **31.4.4 Geschäftsmodelle neu definiert**

Die Musikindustrie hat als erste erfahren, dass die Digitalisierung des Lebens ein Umdenken in den Geschäftsmodellen erfordert. Waren werden verstärkt immateriell. Das heißt, der Wert wird nicht mehr nur am Produkt, sondern auch an den zur Verfügung stehenden Dienstleistungen festgemacht. Mit anderen Worten: Der Wert verschiebt sich von dem „Was habe ich“ zu „Was habe ich davon“. Diese neue Bewertung erfordert gänzlich neue Geschäftsmodelle und die Integration von vielen Marktteilnehmern in einem Gesamtsystem [9, 10]. Wo in der Vergangenheit das beste „Supply-Chain“-System den höchsten Profit erzielt hat, wird dies in Zukunft das erfolgreichste „Supply- und Service“-System sein. Dabei müssen alle Dienstleistungen im System jederzeit messbar und bewertbar sein. Nur so ist es möglich, verschiedene direkte (pro Zeit, Leistung oder Ansicht etc.) oder indirekte (Werbung, Verlinkung, Empfehlung, Punkte etc.) Bezahlverfahren abzubilden.

## **31.5 Anforderungen an eine erfolgreiche mobile Infrastruktur der dritten Generation**

*Wenn der Wind des Wandels weht, bauen die einen Schutzmauern, die anderen bauen Windmühlen.*  
Chin. Sprichwort

Um auf die Veränderungen erfolgreich zu reagieren, benötigt jede Organisation eine Strategie und eine technologische Infrastruktur, um diese umzusetzen. So sehr die Strategie individuell für jede Organisation ist, so leicht lässt sich die Anforderung an die digitale Infrastruktur mit wenigen Kerneigenschaften beschreiben.

### **31.5.1 Offene Plattformen**

Offene Plattformen schaffen erfolgreiche und langfristig rentable Ökosysteme. „Offen“ ist hier auf zwei Arten zu verstehen: Erstens ein System, dass auf offenen von einem zentralen Kontrollgremium getragenen Standards beruht. Zweitens quelloffene Entwicklung (Open Source), bei der verschiedenen Marktteilnehmer gemeinschaftlich an einer Implementierung einer Lösung arbeiten. Eine wirklich offene Plattform integriert beides. Der Vorteil dieser offenen Plattformen liegt in der Tatsache, dass sich externe Intelligenz und Innovation zielgerichtet in das IT-System integrieren lassen. Kommerzielle Produkte, Support und Dienstleistungen basierend auf diesen offenen Plattformen bieten dann größtmögliche Sicherheit und Qualität für den geschäftlichen Einsatz. Natürlich ist Offenheit kein Garant für Erfolg. Schließlich erfordert es eine Anfangsinvestition, um die Plattform bekannt zu machen. Aus den genannten Gründen sind die meisten „neuen“ erfolgreichen Plattformen abgeleitet von existierenden erfolgreichen Plattformen. Vor allem das Einbinden einer großen existierenden Entwicklergemeinschaft kann den Marktzugang deutlich beschleunigen.

### **31.5.2 Sicherheit**

In einer Welt, wo alles mit allem integriert ist, ist Sicherheit eine der zentralen Herausforderungen an jedes IT-System. Sicherheit hat zwei wesentliche Anforderungen: Zum einen die technologische Integration der Sicherheitsmaßnahmen in die verwendeten Plattformen. Zum anderen auch die Fähigkeit der Nutzer, die Sicherheitskomponenten (organisatorisch) richtig anzuwenden.

Im „Internet der Dinge“ muss festgelegt werden, wann welche Anwendung auf ein anderes System zugreifen darf und wer oder welches System meine persönlichen Informationen in welchem Maß einsehen kann. Um dies bei der Vielzahl von Diensten und Situationen beherrschbar zu machen, müssen neue Sicherheitskonzepte entwickelt werden, die Sicherheit nach bestimmten Profilen und Risikoklassen automatisch bewerten und garantieren. Wichtig ist, dass der Grad der Sicherheit für den Anwender jederzeit begreifbar, überprüfbar und verständlich ist.

### **31.5.3 Kontextbewusst**

Die wunderbare Welt der neuen Technologie hat eine große Achillesferse für uns Menschen: *Stress*. Die Anzahl von Daten und Ereignissen die auf unser Leben einwirken, und die Summe der Entscheidungen, die uns abverlangt werden, sind größer als je zuvor. Dieser Tatbestand wird in den nächsten Jahren noch zunehmen. Unter diesen Rahmenbedingungen besitzen Systeme eine große Attraktivität, die uns Menschen einen Teil dieser Last abnehmen und auf diese Weise unser Leben einfacher machen.

cher machen und unsere Mobilität erhöhen. Ein triviales Beispiel wäre ein Navigationsgerät, das neben dem Reise-Ziel auch den Tankstatus berücksichtigt, dabei die möglichen Tankstellen in der Nähe nach Preis, Wartezeiten und Streckenkomfort sortiert und dem Fahrer diese als Routen-Vorschläge unterbreitet. Der Autofahrer kann dann für sich die optimale Auswahl treffen. Das intelligente Navigationsgerät berücksichtigt dann auch noch dynamisch etwaige Verkehrsbehinderungen. Damit ist nicht nur dem einzelnen Autofahrer geholfen sondern das gesamte System profitiert. Im Idealfall könnten so Staus vermieden werden.

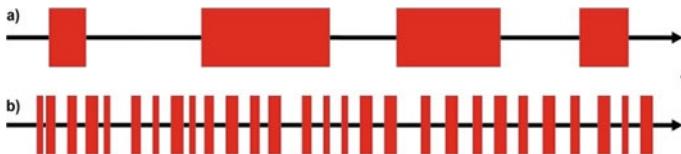
Dies zu erreichen erfordert ein neues Modellierungsparadigma für Systeme. Bisherige Systeme arbeiten ereignisgesteuert; ist z. B. der Inhalt in meinem Tank zu gering, leuchtet ein Anzeigenlicht auf, da das System auf *eine* Funktion hin ausgerichtet ist. Im „Internet der Dinge“ stehen dem System multiple Daten und Informationen zur Verfügung, die es miteinander verketten kann. Auf dieser Basis können dem Anwender Vorschläge unterbreitet werden. Dieses Paradigma sei an dieser Stelle als „kontextbewusst“ bezeichnet.

Es ist abzusehen, dass die Wandlung hin zu situationsabhängigen, „smartten“ Systemen aufgrund der Komplexität der Verknüpfungen nicht über Nacht passieren wird. Diese neuen Dienste erfordern eine sorgfältige Modellierung von Verhaltenssituationen, neuen Datenmodellen und das Einbeziehen und Klassifizieren von verschiedenen Daten- und Informationsquellen – eine komplexe Aufgabenstellung, die jedoch weitreichende Vorteile für Anbieter und Kunden haben wird.

Um kontextbewusste Anwendungen abbilden zu können, muss die gesamte Infrastruktur so aufgebaut sein, dass Entscheidungen in Echtzeit getroffen werden können. Dabei müssen lokale Signale zeitnah aufgenommen, lokal vor-bearbeitet und an Backend-Systeme übertragen werden. Dort muss eine Infrastruktur zur Verfügung stehen, die komplexe Daten aus verschiedenen Quellen miteinander verknüpft und unter Berücksichtigung von sich ändernden Parametern, Logiken und Geschäftsprozessen auswertet. Diese Infrastruktur muss dann eine Antwort an das lokale System oder ein anderes System zurückmelden. Dies erfordert eine zielgenaue Abstimmung zwischen lokaler Logik-Plattform, Netzwerkinfrastruktur und Backend-Systemen, denn dem Anwender sollen die Informationen und Entscheidungsvorlagen zeitnah und hochverfügbar zur Verfügung stehen.

Noch eine weitere Schlussfolgerung lässt sich aus diesem Paradigmenwechsel ableiten: Um nämlich erfolgreich zu sein, muss man dem Kunden die richtigen Anwendungen für den gebotenen Kontext zur Verfügung stellen. Diese Wandlung hin zur kürzeren aber häufigeren Interaktion mit dem Internet kann man heute bereits erkennen (s. Abb. 31.1). Dabei zählt offensichtlich nicht die Masse der Anwendungen, sondern dass sie den vertrauten Lebensgewohnheiten entsprechen und als Nutzen wahrgenommen werden. Wenige passende Anwendungen im geeigneten Kontext sind offensichtlich viel nützlicher als eine unüberschaubare Menge von belanglosen Anwendungen.

Der bedeutendste Vorteil für die Hersteller der klassischen Industrien ist, dass sie diesen Kontext kontrollieren. Sie kennen ihre Kunden genau und sind deshalb in der Lage, die richtigen und erfolgreichen Dienste anzubieten. Es wird sich zeigen,



**Abb. 31.1** Darstellung der Internetnutzungsveränderung durch Smartphones: **a** vorheriger zeitliches Verhalten und **b** heutiges Verhalten (nach [11])

welche Hersteller diese Chance nutzen und beginnen geeignete Infrastrukturen und Know-how aufzubauen, um in diesem Rennen vorn dabei zu sein.

### 31.5.4 Modular und verteilt im Netz lauffähig

Die Topologie des „Internet der Dinge“ ist eine Mischung zwischen Maschen- und Stern-Struktur [12]. Daher sind die intelligenten Hubs sternförmig mit den lokalen Sensoren verbunden und sowohl untereinander als auch mit den Backend-Systemen mittels Maschen-Struktur vernetzt [13]. In einem solchen, hochgradig vernetzten und verteilten System ist es naheliegend, auch die Laufzeitumgebung modular und verteilt zu gestalten. Wichtig ist hierbei, dass der gesamte Lebenszyklus einer Anwendung, von der Erst-Verteilung über das Management, das Update und die Verschiebung bis hin zur Löschung plattformseitig unterstützt wird. So können Systeme dezentralisiert, dynamisch und über viele Knoten hinweg zusammenarbeiten. Im Internet der Dinge wird das Netzwerk der Computer [14].

### 31.5.5 Hardware und Betriebssystem-Unabhängigkeit

Welche CPU-Architektur, welcher Chipsatz oder welches Betriebssystem für die Hardware der digitalen Hubs eingesetzt wird, ist nicht vorhersehbar. Der Markt dafür ist zu dynamisch. Des Weiteren ist anzunehmen, dass verschiedene digitale Hubs mit unterschiedlichen Technologien realisiert werden. Sicher ist, dass die eingesetzten Systeme weit leistungsfähiger und gleichzeitig sparsamer sein werden als heutige. Angesicht dieser Tatsachen ist es folgerichtig, eine plattformunabhängige Laufzeitumgebung zur Verfügung zu stellen und die unterliegenden Funktionen und Schnittstellen zu abstrahieren. Eine solche virtualisierte Umgebung ermöglicht auch eine einfachere Pflege, ein einfacheres Management und einen längeren Lebenszyklus für die Anwendungen.

### 31.5.6 Integration mit Sensorik und Nachfeld-Kommunikation

Der digitale Hub der dritten mobilen Generation muss mehr als je zuvor mit seinem Umfeld in Kontakt treten und kommunizieren können. In letzter Zeit treten neben den etablierten kabelgebundenen Kommunikationssystemen (USB, I2C, CAN-Bus etc.) stärker die Funktechnologien in den Vordergrund, wie z. B. Bluetooth und Near Field Communication (NFC), WiFi, ZigBee etc. Alle eingesetzten Technologien haben eine (relativ) kurze Reichweite und einen Fokus auf geringen Energieverbrauch. Sie unterscheiden sich aber wesentlich hinsichtlich ihres Anwendungszwecks, verwendeter Protokolle und Netzwerkaufbau [15–17]. Für die Plattform bedeutet dies vor Allem, dass abstrahierte APIs zur Netzwerkkommunikation zur Verfügung gestellt werden müssen, um Anwendungen und Module leichter zu entwickeln und portabel zu gestalten.

## 31.6 Die Architektur einer Cloud für mobilen Dienste

*Die Wolken gehören zur Erde, nicht zum Himmel.*

Waldemar Bonsels

Eine fundierte Architektur ist das Rückgrat eines erfolgreichen Service-Systems. Im Zeitalter der „Cloud-Services“ muss es das Ziel sein, eine automatisierte, dynamische und bedarfsgerechte Infrastruktur für das „Internet der Dinge“ zur Verfügung zu stellen. Dabei müssen die verschiedenen Teile nicht nur technologisch miteinander und mit bestehenden Infrastrukturen integriert werden. Das System muss auch dem Entwickler einheitlich und transparent zur Verfügung gestellt werden. Und es muss möglich sein, verschiedene Geschäfts- und Preismodelle abzubilden. In Abb. 31.2 ist eine solche Architektur mit den relevanten Bestandteilen skizziert.

**Entwicklungsumgebung:** eine auf die Infrastruktur abgestellte Entwicklungsumgebung mit guten Beispielen, Best Practices und ausführlicher Dokumentation der zu verwendenden APIs.

**Service Repository und Store:** zentraler Datenspeicher und Verzeichnis aller vorhandenen Anwendungen mit angeschlossener E-Commerce-Plattform zur Bereitstellung und Abrechnung von Anwendungen und Diensten.

**Real-Time Service Cloud:** Adaptive, hochverfügbare und dynamisch skalierbare Backend-Laufzeitumgebung und Managementsystem für die im Netzwerk verteilt laufenden Anwendungen mit eingebauter Echtzeit-Analyse und der Möglichkeit, Geschäftsprozesse abzubilden.

**Service, Integration und Data Backend:** zentrale Infrastrukturdienste, die allen Anwendungen mit zentraler Datenhaltung zur Verfügung stehen.

**Service Hub:** Laufzeitumgebung für die mobilen Dienste mit Anschluss an andere Anwendungen, Systeme, Netzwerkelemente, Sensoren etc.

**Client Control Services:** Definition von Kontrolldiensten und Schnittstellen, die die Erstellung von Anwenderschnittstellen in Gestalt von Web, Rich-Client oder „Mashup Services“ ermöglichen.

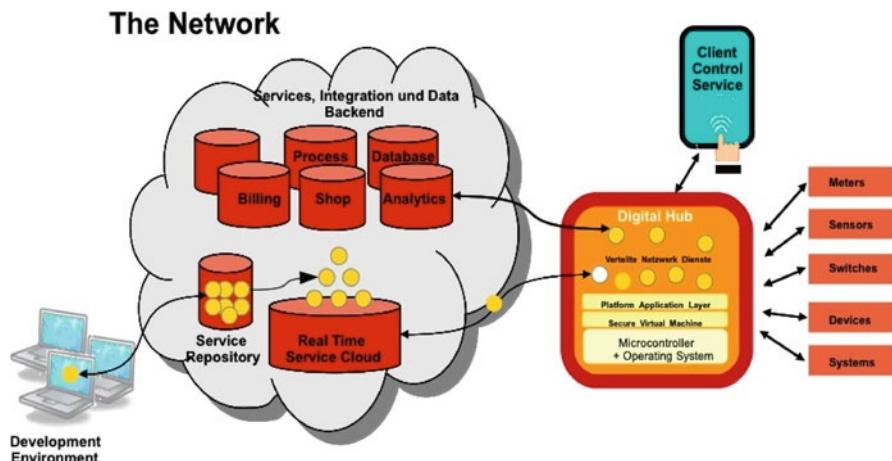


Abb. 31.2 Übersicht über die Architektur einer Cloud für mobile Dienste

Wichtig für eine solche Infrastruktur ist, dass die verwendeten Bauteile aufeinander abgestimmt sind. Nur so ist ein konkurrenzfähiger Aufbau und Betrieb einer derartigen Infrastruktur möglich.

## 31.7 Java, die Plattform für das Internet der Dinge

*Langfristig ist man nur erfolgreich, wenn man weiß, warum man erfolgreich ist.*

Rupert Lay

Java ist die populärste und am meisten verbreitete Programmiersprache unter Entwicklern [18, 19] und wächst weiter stark [20]. Die Java Virtual Machine ist die am häufigsten eingesetzte Laufzeitumgebung für Smartcards mit 5 Milliarden ausgelieferten Java Cards [21]), für Handys mit 3 Milliarden ausgelieferte Geräte [22], für Desktops mit 1.1 Milliarden installierten Laufzeitumgebungen [23] und 950 Millionen jährlichen Downloads [24] Darüber hinaus ist die Java Enterprise Edition (Java EE) das dominierende Programmiermodell für die Entwicklung von Enterprise- und Internet-Backend-Anwendungen. Java bietet alle Voraussetzungen um die Sprache und Laufzeitumgebung für das „Internet der Dinge“ zu werden:

### Offenheit

Die Java-Plattformen werden durch ein Standardisierungsboard (den „Java Community Process“, kurz JCP) unter Beteiligung von Firmen, Universitäten, Ländern, Verbänden und vielen Individuen vorangetrieben. Die Kern-Plattform, die Java Standard Edition, wird gemeinschaftlich von Oracle, IBM, SAP, Apple und vielen an-

deren als Open-Source-Projekt [OpenJDK] entwickelt und ist unter freier Lizenz verfügbar. Dazu gibt es von vielen Firmen zertifizierte kommerzielle Lösungen.

### Sicherheit

Java hat ein integriertes Sicherheitsmodell, das das sichere dynamische Nachladen von Modulen und Anwendungen aus dem Netz ermöglicht [25]. Das sogenannte „Java-Sandbox“-Modell besteht aus drei Teilen, die Code-Integrität (Byte Code Verifier), Programm- und System-Integrität (Class Loader) und Zugriffsintegrität (Security Manager) sicherstellen. Dieses Modell garantiert, dass Programme während ihres gesamten Lebenszyklus individuell überwacht und in ihren Auswirkungen kontrolliert werden können. Basierend auf dieser Basistechnologie lassen sich die Sicherheitsanforderungen des „Internets der Dinge“ als Regeln für die „Java-Sandbox“ hinterlegen, indem man z. B. verlässliche Quellen für Anwendungen definiert.

### Kontextbewusstheit

Kontextbewusste Anwendungen in Java werden seit einigen Jahren erforscht und entwickelt. Die meisten bisherigen Ansätze beruhen darauf, den Kontext auf der Anwendungsebene mittels eines Kontextmanagementsystems (CMS) abzubilden. Die bekannteste Lösung dafür ist das „Java Context Aware Framework“ [26, 27]. Einen anderen Ansatz verfolgen Forscher der Universität Potsdam, die Java um ein spezielles Java-Sprachkonstrukt-„Layer“ erweitern, dass die Kontextänderung direkt abbildet [28]. Vor allem die zweite Möglichkeit wird durch die mit Java 7 unter dem Namen „Da-Vinci-Projekt“ eingeführten Spracherweiterungen zur Förderung von dynamischen Sprachen deutlich vereinfacht [29]. Unabhängig von der gewählten sprachlichen Umsetzung kann man so leicht ein Szenario umsetzen wie: „Ich verlasse das Haus und meine, von mir vergessene Kaffeemaschine schaltet sich aus, sobald mein Auto die Garage verlässt“.

### Modular und verteilt im Netz lauffähig

Von Anfang an war Java darauf ausgelegt, eine verteilte vernetzte Infrastruktur zu ermöglichen, in welcher Programmteile dynamisch nachgeladen werden und Aufgaben dynamisch im Netzwerk verschoben werden können [30]. Mit dem OSGI-Standard wurde diese Grundeigenschaft um ein modulares Komponentenmodell für Java-Anwendungen erweitert. Der nächste Schritt ist die Modularisierung der *Java Virtual Machine* (Java VM oder auch JVM), die mit dem Java SE 8 Projekt Jigsaw [JigsamW] und Java Specification Request (JSR) 284 [JSR284] vorangetrieben wird.

## Hardware- und betriebssystemunabhängig

Die Java-Plattform beruht auf einer systemunabhängigen, effizienten und virtualisierten Laufzeitumgebung, der Java Virtual Machine. Diese Architektur erlaubt Anwendungen, systemunabhängig zu sein. Über die vergangenen 16 Jahre hat sich dieses System ökonomisch und technisch bewiesen. Es hat gezeigt, dass langfristige kontinuierliche Code-Entwicklung und -Pflege sinnvoller ist als das kurzfristige Erreichen der letzten Bit-Performanz oder der optimalen Systemintegration. Unterdessen dient die JVM auch als Laufzeitumgebung für viele andere Sprachen [31].

Um verschiedenen Hardwaremöglichkeiten und Anwendungsfällen Rechnung zu tragen, haben sich über die Zeit verschiedene Plattformen von Java herauskristallisiert: Java EE, Java SE, Java ME CDC und CLDC und JavaCard (s. Abb. 31.3). Diese können im „Internet der Dinge“ für unterschiedlichste Zwecke eingesetzt werden. So ist JavaCard dazu geeignet, hochsichere intelligente Sensoren im Netz bereitzustellen. Java ME CLDC eignet sich zum Beispiel, um auf M2M-Modulen Dienste zur Vernetzung zur Verfügung zu stellen. Java ME CDC kann eingesetzt werden um kompakte Kontrollgeräte oder Gateways zur Verfügung zu stellen. Java SE kann auf hochwertigen Hubs und Embedded-Systemen zum Einsatz kommen, um z. B. eine lokales dezentrales Backend- und Service-System zur Verfügung zu stellen. Der gemeinsame Nenner ist bei allen Plattformen die Beibehaltung der essentiellen Java-Spracheigenschaften. So kann ein Java-Projekt relativ leicht auf eine

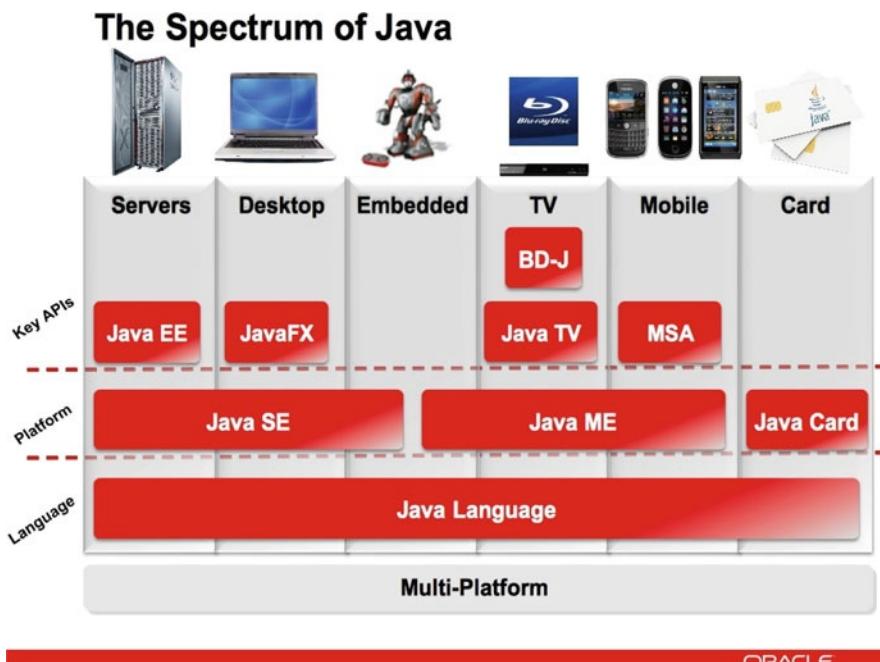


Abb. 31.3 Das Spektrum der Java-Plattform und die durch sie abgedeckten Marktsegmente

„größere“ Plattform migriert werden, wenn Weiterentwicklungen in der Hardware den Einsatz ermöglichen.

### Integration mit Sensorik und Nahfeldkommunikation

Die Java Virtual Machine bietet verschiedene Möglichkeiten, auf Sensorik und Nahfeldkommunikation zuzugreifen. Eine sehr häufig verwendete Möglichkeit besteht darin, native, also in C/C++ oder Assembler geschriebene, System-Interfaces über das Java Native Interface (JNI) innerhalb der Laufzeitumgebung den Java-Programmen zur Verfügung zu stellen [32]. Bei datenorientierten Diensten werde diese oft über das Java Database Interface (JDBC) an die JVM übergeben. Dabei kann man entweder einen JDBC-Treiber schreiben, oder aber die Daten in einer lokalen Datenbank, z. B. BerkleyDB, zwischenspeichern, auf die dann von Java aus zugegriffen wird [33]. Ein drittes Verfahren stellt die Sensorinformationen mittels eines Netzwerkinterface zur Verfügung, z. B. als TCP-Socket oder als Webservice [34]. Neben den hier beschriebenen gibt es noch viele andere Möglichkeiten [35–37].

Unabhängig davon, welche Variante man verwendet, Java erlaubt einem, diese Zugriffe zu kapseln und zu abstrahieren, um sie dann im Netzwerk sicher zur Verfügung zu stellen. So kann man z. B. ein Bezahl-Framework zur Verfügung stellen, das die Verwendung der eingesetzten Technologie, wie etwa NFC-Payment oder Karten-Bezahlung oder Bezahlung über SMS-Dienste, für die darauf aufsetzenden Anwendungen vollständig verbirgt.

## 31.8 Der Weg zum mobilen Service-System

*Jede Reise beginnt mit einem ersten Schritt.*

Chinesisches Sprichwort

Wenn große Veränderungen anstehen, ist es eine häufige Reaktion des Menschen abzuwarten, was passiert, um dann den richtigen Zeitpunkt des Einstiegs zu finden. Die jetzt bevorstehende dritte Phase mobiler Anwendungen bietet eine große Chance für Unternehmen, diesen Wandel mitzugestalten. Wer als Unternehmen langfristig in der digitalen Welt überleben möchte, sollte diese Gelegenheit nicht verpassen. Dabei ist der Einstieg leichter als man denkt und die nächsten Schritte sind bereits klar.

### ***31.8.1 Schritt 1: Einbeziehung aller bekannten Anwendungsarchitekturen***

Wer das Rennen der Plattformen um die Akzeptanz der Kunden, das größte Volumen an Geräten, die größten Profite etc. gewinnt, ist noch ungewiss, denn es hat gerade erst begonnen. Die Kunden wechseln noch häufig und probieren Neues aus. Diese Situation bietet die Möglichkeit zu testen, welcher Service, welche Lösung und welches Geschäftsmodell funktionieren und welche nicht. Dabei sollte man keine zu schnellen Ergebnisse erwarten. Investitionsbereitschaft und eine sorgfältige Analyse tragen zu einer erfolgreichen Strategie bei. In dieser Phase sollten folgende Fragen beantwortet werden:

- Welche Dienste möchte ich anbieten?
- Welche Marktzugänge und Plattformen möchte ich bedienen?
- Welche Geschäftsmodelle stehen mir zur Verfügung?

### ***31.8.2 Schritt 2: Entwicklung einer eigenen Plattformstrategie***

Neben dem Sammeln von praktischen Erfahrungen müssen Unternehmen, Industrien und Länder nach sorgfältige Überlegungen ihre Rolle definieren, die sie bei diesen Veränderungen spielen wollen und können. Dieser Schritt ist der schwierigste. Er erfordert viel Geduld, Zeit, Geld und Mut sowie eine klare Analyse der Stärken und Schwächen. Um im Rahmen der Globalisierung einen relevanten Einfluss ausüben zu können, müssen in dieser Phase Netzwerke und Kooperation gebildet werden, die zukunftsfähig und langfristig dynamisch sind. Hier sollten folgende Fragen auf der Liste stehen:

- Welche Rolle soll meine Organisation im „Internet der Dinge“ spielen?
- Welche Kooperationspartner brauche ich für meine Pläne?
- Wie gleiche ich meine Schwächen aus?

### ***31.8.3 Schritt 3: Aufbau des neuen Service-Systems***

Mit der richtigen Strategie und einem tragenden Netzwerk wird der dritte Schritt einfacher. Es gilt eine Service-Infrastruktur aufzubauen, auf der die Anwendungen der Zukunft betrieben werden können. Wichtig ist hier, bestehende Infrastrukturen zu integrieren und/oder weiterzuentwickeln. In diesem Schritt ist es besonders wichtig, die Infrastruktur offen und erweiterbar zu gestalten. So können Änderungen in der Technologie, in den Geschäftsprozessen und Geschäftsmodellen dynamisch umgesetzt werden. Auch die Definition der Entwicklungsumgebung und der Interfaces fällt in diese Phase wobei diese mit strategischen internen und externen Entwick-

lungspartnern abgeglichen werden, so dass beim Start der Infrastruktur die wesentlichen Kerndienste bereits zur Verfügung stehen. Dafür ist es notwendig, Werkzeuge für die Zusammenarbeit der internen und externen Entwickler wie Foren, Code-Repository etc. zur Verfügung zu stellen und zu testen. In dieser Phase ist es wichtig zu klären:

- Welche Teile der Infrastruktur sollen geöffnet werden und für wen?
- Welche Entwicklungsmöglichkeiten und Systeme biete ich an?
- Wie viel von meiner IP mache ich frei verfügbar?
- Wie baue ich ein externes Gremium und eine Community auf?

### ***31.8.4 Schritt 4: der „Roll-Out“ (Marktzugang)***

Die Infrastruktur, die Geschäftsmodelle und alle strategischen Partnerschaften sind definiert/umgesetzt? Jetzt ist es Zeit, an den Markt zu gehen und das vorher konzipierte Ökosystem zu propagieren. Diese Phase ist wichtig und oft entscheidend für den späteren Erfolg einer Plattform. Die Erfahrungen aus der Vergangenheit haben gezeigt, dass es wichtig ist, zu diesem Zeitpunkt die Lösung der strategischen Partner neben den eigenen Anwendungen vorzustellen. Wichtig sind hier folgende Gedanken:

- Habe ich alle wichtigen Partner an Bord?
- Sind alle Systeme und Dienste marktreif?

### ***31.8.5 Schritt 5: Anwendungsentwicklung und kontinuierliche Weiterentwicklung des Systems***

Wenn das Geschäftsmodell tragfähig ist, finanziert sich die Infrastruktur bald selbst und erwirtschaftet eine gute Rendite für alle Marktteilnehmer. Der letzte Schritt ist dann die kontinuierliche Pflege der Infrastruktur, der Geschäftsprozesse und des Ökosystems. Dafür ist es wichtig, regelmäßig neue Anwendungen und Dienste auszurollen und Innovationen in die Plattform zu integrieren. Sind die Prozesse richtig aufgesetzt, erhält man ständig Rückmeldung von Partnern und Entwicklern, wie das System verbessert und modernisiert werden kann. Setzt man diese um, so ist das System langfristig aktiv und innovativ.

Die essentiellen Überlegungen sind:

- Ist mein Ökosystem weiterhin attraktiv?
- Was kann ich tun, um die Attraktivität weiter zu erhöhen?
- Welche neuen Dinge, Dienste und Geschäftsmodelle kann ich anbieten?

## 31.9 Schlussfolgerungen

*Nicht weil es schwer ist, wagen wir es nicht, sondern weil wir es nicht wagen, ist es schwer.*  
Seneca

Eine aufregende Zeit steht uns bevor. Das Zeitalter des „Internet der Dinge“ wird unser Leben nachhaltig verändern. Es ist eine neue Umwelt, mit vielen Chancen, Möglichkeiten und Entdeckungen, aber auch mit großen Herausforderungen. Es ist jetzt Zeit, die mobilen Dienste und Infrastrukturen der nächsten Generation zu planen und mit der besten Technologie auszustatten, die zur Verfügung steht.

Lassen Sie uns gemeinsam auf diese Entdeckungsreise gehen.

### Trademarks

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

### Safe Harbor Statement

The above is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decision. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

## Literaturverzeichnis

1. <http://www.rfidjournal.com/article/view/4986>, Juni 2011
2. CNN Webseite 2004: Internet language runs remote-controlled Mars rover. <http://edition.cnn.com/2004/TECH/space/01/16/space.mars.java.reut/index.html>, Juni 2011
3. Spiegel Online, 25.5.2011: „Spirit“ antwortet nicht <http://www.spiegel.de/wissenschaft/weltall/0,1518,764841,00.html>, Juni 2011
4. WikiCity, sensible city lab, Massachusetts Institute of Technology (MIT). <http://senseable.mit.edu/wikicity/rome/>, Juni 2011
5. Webseite des Glacsweb projects der University of Southampton. <http://glacsweb.org/>, Juni 2011
6. Präsentation: „Social Brand Marketing For Web 2.0“. <http://www.slideshare.net/lazerow/social-brand-marketing-for-web-20-presentation>, Juni 2011
7. [www.razorfish.com/reports/DigConsStudy.pdf](http://www.razorfish.com/reports/DigConsStudy.pdf), Juni 2011
8. [www.booz.com/media/file/The\\_Urgent\\_Need\\_For\\_Companies\\_To\\_Adapt\\_To\\_Web\\_2\\_0\\_FINAL.pdf](http://www.booz.com/media/file/The_Urgent_Need_For_Companies_To_Adapt_To_Web_2_0_FINAL.pdf), Juni 2011
9. <http://globalhumancapital.org/the-knowledge-economy-the-ultimate-context-for-understanding-the-future/>, Juni 2011

10. <http://www.wantchinatimes.com/news-subclass-cnt.aspx?cid=1202&MainCatID=12&id=20110421000006>, Juni 2011
11. From Apps To Everyday Situations. [http://www.ericsson.com/res/docs/2011/silicon\\_valley\\_brochure\\_letter.pdf](http://www.ericsson.com/res/docs/2011/silicon_valley_brochure_letter.pdf), Juni 2011
12. Wikipedia Artikel zu Netzwerktopologien.  
[http://en.wikipedia.org/wiki/Network\\_topology](http://en.wikipedia.org/wiki/Network_topology), Juni 2011
13. Wikipedia Artikel zu Maschen-Netzwerken.  
[http://en.wikipedia.org/wiki/Mesh\\_networking](http://en.wikipedia.org/wiki/Mesh_networking), Juni 2011
14. John Gage und Bill Joy, beide bei Sun Microsystems, prägen den Satz „the network is the computer“. <http://tincan.amplify.com/2009/09/02/the-network-is-the-computer/>, Juni 2011
15. Wikipedi Eintrag zu Near field Communication. [http://en.wikipedia.org/wiki/Near\\_field\\_communication](http://en.wikipedia.org/wiki/Near_field_communication), Juni 2011
16. Zigbee Organization Home Page. <http://www.zigbee.org/>, Juni 2011
17. Eintrag auf Wikipedia mit Vergleich verschiedener Wireless-Standards. [http://en.wikipedia.org/wiki/Comparison\\_of\\_wireless\\_data\\_standards](http://en.wikipedia.org/wiki/Comparison_of_wireless_data_standards), Juni 2011
18. „Evans Data Global Development Survey 2009“
19. TIOBE Programming Community Index for July 2011. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>, Juni 2011
20. Evans Data Cooperation Presse Mitteilung bezüglich des „Evans Data Global Development Survey 2011“. <http://www.evansdata.com/press/viewRelease.php?pressID=175>, Juni 2011
21. Akkumulierte Zahl der eingesetzten JavaCard-Lizenzen nach Meldung an Oracle.
22. Gardner Zahlen zur Verbreitung von Feature Phones mit Java
23. Gardner Zahlen zur Verbreitung von Java auf Desktops und Laptops
24. Oracle-interne Downloadstatistik über Java Runtime Downloads zwischen August 2009 und August 2010
25. Understanding the keys to Java security – the sandbox and authentication. <http://www.javaworld.com/javaworld/jw-05-1997/jw-05-security.html?page=1>, Juni 2011
26. Übersicht zum „Java Context Aware Framework“ auf Mendeley.com. <http://www.mendeley.com/research/the-java-context-awareness-framework-jcaf-a-service-infrastructure-and-programming-framework-for-contextaware-applications-2/>, Juni 2011
27. Homepage des „Java Context Aware Frameworks“. [www.daimi.au.dk/~bardram/jcaf/](http://www.daimi.au.dk/~bardram/jcaf/), Juni 2011
28. Malte Appeltauer und Robert Hirschfeld – Paper zu ContextJ. [http://www.hpi.uni-potsdam.de/hirschfeld/publications/media/AppeltauerHirschfeld\\_2008\\_LanguageAndInfrastructureSupportForContextAwareServices.pdf](http://www.hpi.uni-potsdam.de/hirschfeld/publications/media/AppeltauerHirschfeld_2008_LanguageAndInfrastructureSupportForContextAwareServices.pdf), Juni 2011
29. Die Webseite des Da-Vinci-Projekts zur besseren Unterstützung dynamischer Sprachen auf der JVM. <http://openjdk.java.net/projects/mlvm/>, Juni 2011
30. The Java Language Environment. <http://java.sun.com/docs/white/langenv/index.html>, Juni 2011
31. Wikipedia – Overview of Languages Running on Top of Java VM. [http://en.wikipedia.org/wiki/List\\_of\\_JVM\\_languages](http://en.wikipedia.org/wiki/List_of_JVM_languages), Juni 2011
32. Java Native Interface: Programmer’s Guide and Specification. <http://java.sun.com/docs/books/jni/>, Juni 2011
33. Olson M, Bostic K, Seltzer M, Berkeley DB. [http://www.usenix.org/events/usenix99/full\\_papers/olson/olson.pdf](http://www.usenix.org/events/usenix99/full_papers/olson/olson.pdf), Juni 2011
34. Overview of Networking through JAVA.  
<http://www.roseindia.net/java/network/index.shtml>, Juni 2011
35. Wikibuch über Java Serielle Kommunikation.  
[http://en.wikibooks.org/wiki/Serial\\_Programming/Serial\\_Java](http://en.wikibooks.org/wiki/Serial_Programming/Serial_Java), Juni 2011
36. Arbeiten mit der Mobile Sensor API für Java ME. <http://www.oracle.com/technetwork/java/mobile-sensor-api-172849.html>, Juni 2011
37. Projekt, das eine Sensor API in Java zur Verfügung stellt. <http://code.google.com/p/org-concord-sensor/>, Juni 2011
38. OECD Statistik zur Durchdringung von Internetzugängen in 2009. <http://www.oecd.org/dataoecd/20/59/39574039.xls>, Juni 2011

**Teil IX**

**Marktwirtschaftliche Sicht:**

**Geschäftsmodelle & Marktforschung**

---

# Kapitel 32

## Plattformoffenheit – Implikationen für Anbieter von Mobile Business Apps

Daniel Hilkert, Christoph Burkard, Thomas Widjaja, Thomas Hess und Peter Buxmann

**Zusammenfassung** Business-Software-Anbieter, die zukünftig auch mobile Lösungen anbieten wollen, stehen vor den beiden zentralen Fragestellungen, auf welcher Plattform ihre Lösung angeboten werden soll und ob nur eine oder mehrere Plattformen berücksichtigt werden müssen. Ziel dieses Beitrags ist es, die Parameter aktueller mobiler Plattformen zu ermitteln, die diese Entscheidungen beeinflussen. Als zentrale Stellschrauben einer Plattform werden dazu die vertikale und horizontale Offenheit einer Plattform analysiert. Anhand praktischer Beispiele wird diskutiert, welche Implikationen sich aus Sicht der Anbieter von mobilen Business-Applikationen für die einzelnen Facetten der Offenheit von mobilen Plattformen ergeben.

---

Daniel Hilkert  
Ludwig-Maximilians-Universität München, Institut für Wirtschaftsinformatik und Neue Medien,  
E-mail: hilkert@bwl.lmu.de

Christoph Burkard  
Technische Universität Darmstadt, Fachgebiet Wirtschaftsinformatik | Software Business & Information Management,  
E-mail: burkard@is.tu-darmstadt.de

Thomas Widjaja  
Technische Universität Darmstadt, Fachgebiet Wirtschaftsinformatik | Software Business & Information Management,  
E-mail: widjaja@is.tu-darmstadt.de

Thomas Hess  
Ludwig-Maximilians-Universität München, Institut für Wirtschaftsinformatik und Neue Medien,  
E-mail: thess@bwl.lmu.de

Peter Buxmann  
Technische Universität Darmstadt, Fachgebiet Wirtschaftsinformatik | Software Business & Information Management,  
E-mail: buxmann@is.tu-darmstadt.de

## 32.1 Einleitung

Mit der Einführung des *iPhone* im Jahr 2007 und der kurz darauf erfolgten Eröffnung des *App Store* als zentralem Marktplatz für mobile Applikationen wurde ein bis heute anhaltender Siegeszug einer neuen Generation von Smartphones begründet. Während zuvor die Attraktivität eines mobilen Endgeräts vor allem durch die Qualität der Hardware-Ausstattung (MP3-Player, Kamera, etc.) determiniert wurde, gelten heute der Umfang und die Qualität des Angebots komplementärer Applikationen („Apps“) als entscheidender Wettbewerbsvorteil<sup>1</sup>. Als Folge dieser Entwicklung ist ein Markt für mobile Applikationen entstanden, in dem zwischenzeitlich weltweit mehrere Milliarden USD jährlich umgesetzt werden<sup>2</sup>.

Während die genannten Entwicklungen bislang vor allem auf den Erfolg im Segment der Endkunden zurückzuführen waren, wird das mobile Umfeld zunehmend auch für Anbieter von Business-Applikationen interessant<sup>3</sup>. Als *mobile Business-Apps* werden dabei Anwendungen bezeichnet, deren Funktionalität einen Geschäftsprozess unterstützt und die für den Einsatz auf einem mobilen Endgerät wie Smartphones oder Tablet-PCs entwickelt und optimiert wurden<sup>4</sup>. Das Spektrum mobiler Business-Apps reicht damit von vergleichsweise einfachen Produktivitäts-Tools, wie einer iPad-Whiteboard-Lösung<sup>5</sup>, bis hin zu komplexen Anwendungen, die bspw. Geschäftsprozesse eines CRM- oder ERP-Systems mobil verfügbar machen<sup>6</sup>.

Vor einer besonderen Herausforderung stehen dabei insbesondere Anbieter „klassischer“ Business-Software, die zukünftig auch mobile Lösungen anbieten wollen. Diese Anbieter unterscheiden sich in verschiedenen Parametern gegenüber den etablierten Anbietern von Konsumenten-Apps, wie den technischen Anforderungen oder den Anforderungen an die Distribution der Apps (siehe Abschn. 32.2.2). Die beiden zentralen Fragestellungen solcher Anbieter sind dabei, auf welcher Plattform ihre Lösung angeboten werden soll und ob nur eine oder mehrere Plattformen berücksichtigt werden müssen. Die Offenheit einer Software-Plattform gilt dabei für beide Fragestellungen als zentrale Stellschraube, unter der die Entscheidungen zur

<sup>1</sup> Dieser Trend manifestiert sich auch in der nachhaltigen Veränderung der Marktstruktur von Smartphonebetriebssystemen: Während Nokias vormals dominierendes Symbian-Betriebssystem zunehmend Marktanteile verliert (von 46 % in 2009 auf ca. 36 % in 2010), gewinnen Apples iOS (ca. 16 % Wachstum in 2010) und Android (ca. 22 % Wachstum in 2010) signifikant hinzu (vgl. <http://www.gartner.com/it/page.jsp?id=1543014>).

<sup>2</sup> Vgl. dazu Apples Keynote im März 2011, in der bekannt gegeben wurde, dass allein Apple bislang über 2 Milliarden USD an Erlösen aus dem iTunes App Store an die Entwickler ausgezahlt hat.

<sup>3</sup> Vgl. bspw. <http://online.wsj.com/article/SB10001424052702303891804575576380032084938.html>.

<sup>4</sup> In Anlehnung an [1], die „electronic business services“ als Anwendungen definieren, die Geschäftsprozesse unterstützen (S. 77).

<sup>5</sup> Vgl. bspw. die App „Whiteboard: Collaborative Drawing“ (<http://www.Greengar.com/apps/whiteboard/>), die es einer Arbeitsgruppe ermöglicht, mit verschiedenen mobilen Endgeräten gemeinsam auf einem virtuellen Whiteboard zu arbeiten.

<sup>6</sup> Vgl. dazu die mobilen Apps von SAP oder salesforce.com.

Konfiguration und Gestaltung einer Software-Plattform und des zugehörigen Eco-systems<sup>7</sup> subsumiert werden. In diesem Beitrag soll daher untersucht werden, welche Implikationen sich aus Sicht der Anbieter von mobilen Business-Applikationen für die einzelnen Facetten der Offenheit von mobilen Applikationsplattformen ergeben.

Dieser Beitrag ist dazu wie folgt aufgebaut: Zunächst werden die Hintergründe zur Plattformoffenheit und zu den Anbietern von Business Apps eingeführt. Darauf aufbauend wird ein Bezugsrahmen von Aspekten der vertikalen Offenheit einer Plattform hergeleitet und anhand konkreter Beispiele aus der Sicht von Business-App-Anbietern diskutiert, welchen Einfluss diese Parameter auf die Plattformaus-wahlentscheidung haben. Anschließend werden Grundlagen der horizontalen Offenheit dargelegt und ausgehend von aktuellen Forschungsergebnissen wird folgende Frage diskutiert: Wie viele und welche Plattformen sollte ein Anbieter bedienen („single vs. multi-homing“)? Abschließend werden in einem Fazit die Erkenntnisse zusammengefasst und ein Ausblick auf weitergehenden Forschungsbedarf gegeben.

## 32.2 Hintergründe

In Vorbereitung der im weiteren Verlauf dieses Beitrags dargelegten Diskussion der praktischen Implikationen von Plattformoffenheit für Anbieter von Business-Apps werden in diesem Abschnitt zunächst die Hintergründe zu Plattformoffenheit einerseits und den Anbietern von Business-Apps andererseits eingeführt.

### 32.2.1 Plattformoffenheit: Begriffliche Abgrenzung und Trade-offs

Unter dem Begriff der „Plattformoffenheit“ werden im Kontext von Software-Plattformen sowohl in der praktischen Diskussion als auch in wissenschaftlichen Beiträgen z. T. sehr unterschiedliche Konzepte subsumiert. So werden bspw. aus einer eher technischen Perspektive Zugriffsmöglichkeiten auf die Funktionalität der Plattform diskutiert. In Abgrenzung dazu stehen in der betriebswirtschaftlichen Diskussion zumeist eher Aspekte der Offenheit des mit einer Plattform assoziierten Marktplatzes im Vordergrund.

Aufgrund der skizzierten inhaltlichen Breite soll zunächst der Begriff der Plattformoffenheit genauer spezifiziert werden. Eisenmann et al. [3] definieren dazu, dass eine softwarebasierte Plattform genau dann als vollständig offen gelten kann,

---

<sup>7</sup> Ecosysteme (aus dem engl. „ecosystems“, im deutschen z. T. auch mit „Ökosysteme“ übersetzt) sind ein aus der Biologie abgeleitetes Konzept, das die Zusammenarbeit und gegenseitige Abhängigkeit von mehreren Anbietern eines gemeinsamen Produktes bzw. Produktsortiments beschreibt [2]. Im Fall mobiler Applikationen besteht das Ecosystem aus dem Betreiber der Software-Plattform (bspw. Apple oder Google), den Anbietern komplementärer Applikationen und den End-kunden.

wenn die Teilnahme, die Entwicklung, die Benutzung und die Kommerzialisierung der Plattform nicht eingeschränkt sind, beziehungsweise wenn alle dennoch vorhandenen Einschränkungen nachvollziehbar sind und für alle Akteure des Plattform-Ökosystems gleichermaßen gelten [3]. Ausgehend von dieser Definition können grundsätzlich zwei Arten der Plattformoffenheit unterschieden werden: Während die Öffnung gegenüber anderen Plattformbetreibern, bspw. durch die Vergabe von Lizenzen, als *horizontale Öffnung* bezeichnet wird, versteht man unter der *vertikalen Plattformoffenheit*, dass auch externen Entwicklern die Möglichkeit gegeben wird, Applikationen beizusteuern und ggf. über den Markt der Plattform zu vertreiben [4].

Die zentrale Problemstellung im Zusammenhang mit der **vertikalen Öffnung** einer Plattform liegt in dem Trade-Off zwischen *Vielfalt und Kontrolle* begründet. Insbesondere wenn der Erfolg einer Software-Plattform eng an das Angebot innovativer Applikationen gebunden ist, wie dies bspw. bei den oben diskutierten Betriebssystemen für Smartphones (iOS oder Android) der Fall ist, kann durch eine vertikale Öffnung der Plattform gegenüber externen Entwicklern die Vielfalt komplementärer Innovationen deutlich gesteigert werden [5]. Gleichzeitig kann mit der vertikalen Öffnung jedoch auch ein Verlust an Kontrolle über die Plattform eintreten. Zum einen wird das Management des Plattform-Ökosystems insgesamt erschwert, da mit den externen Entwicklern mehr Akteure und deren Interessen berücksichtigt werden müssen [6]. Zum anderen verliert der Plattformbetreiber auch zu einem gewissen Grad die Kontrolle über die Weiterentwicklung der Plattform, bspw. hinsichtlich der inhaltlichen Fokussierung. So wäre es zum Beispiel denkbar, dass externe Entwickler einer Plattform es mehrheitlich vorziehen, Spiele anzubieten, obwohl die Plattform für Geschäftsanwendungen entworfen wurde.

Aufgrund der Wechselbeziehung zwischen der *Förderung der Adoption* und dem *Abschöpfen von Gewinnen* stellt auch die **horizontale Öffnung** einer Plattform eine komplexe Managemententscheidung dar. Die horizontale Öffnung (bspw. durch höhere Kompatibilität oder Lizenzierung) einer Plattform führt in der Regel zu einem größeren Ökosystem und kann auch die Adoption beschleunigen, da die Angst der Kunden, an einen einzelnen Betreiber gebunden zu sein, reduziert wird. Die horizontale Öffnung einer Plattform führt jedoch üblicherweise auch zu einem geringeren Gewinnanteil des Betreibers, da Einstiegsbarrieren für Konkurrenten sinken und sich damit der Wettbewerb zwischen den Plattformen intensiviert.

### 32.2.2 Die Anbieter von Business-Apps

Als Grundlage der in den beiden folgenden Abschnitten dargelegten Analyse zu den speziellen Bedürfnissen von Business-App-Anbietern wird hier zunächst dargelegt, wie sich Anbieter von Business-Software von den etablierten Anbietern mobiler Apps unterscheiden.

Ein zentraler Unterschied liegt hierbei zunächst in den verfolgten *Erlösmodellen*. Etablierte Anbieter mobiler Apps versuchen, durch den Verkauf der App oder durch

die Vermarktung von Werbeplätzen in der App direkte Erlöse zu erzielen. Business-Apps stellen hingegen zumeist mobile Ergänzungen von existierenden Business-Software-Lösungen wie bspw. ERP- oder CRM-Systemen dar. Bei solchen Lösungen liegt die Zielsetzung demzufolge darin, den Absatz des Kernprodukts durch attraktive Features der mobilen Lösung zu unterstützen.

Damit einher gehen auch Unterschiede in den *angestrebten Verkaufszahlen* und den Möglichkeiten zur *Distribution der Applikationen*. Anbieter von Konsumenten-Apps streben an, ihre App an möglichst viele Endkunden zu verbreiten, und gehen dazu aufgrund der großen potenziellen Zielgruppe bevorzugt den Weg über die an die Plattform angeschlossenen Marktplätze (bspw. iTunes App Store, Android Marketplace etc.). Business-Apps sind hingegen nur für eine wesentlich kleinere Zielgruppe relevant, daher müssen hier spezifischere Vertriebswege genutzt werden.

Daneben unterscheiden sich die Anbieter auch hinsichtlich ihres *organisationalen Umfelds*. Die Anbieter mobiler Apps sind hier sehr heterogen verteilt (von Amateuren und Einzelentwicklern bis hin zu großen Spieleproduzenten). Im Vergleich dazu kann bei Business-Software-Anbietern davon ausgegangen werden, dass diese bspw. in Bezug auf das zur Verfügung stehende Kapital, die Anzahl der beteiligten Mitarbeiter oder die Softwareentwicklungserfahrung überwiegend deutlich professioneller aufgestellt sind.

## 32.3 Vertikale Öffnung der Plattform

Unter der vertikalen Offenheit einer Plattform versteht man den Grad, zu dem es eine Plattform externen Entwicklern ermöglicht, komplementäre Apps beizusteuern. Die Ausgestaltung von Aspekten der vertikalen Öffnung stellt damit die zentrale Entscheidungsgrundlage für die Plattformauswahlentscheidung von Business-App-Anbietern dar.

### 32.3.1 Framework

Als Grundlage der dargestellten Analyse der Bedürfnisse von Business-App-Anbietern wird im Folgenden ein grundsätzliches Framework von Faktoren der vertikalen Plattformoffenheit aus der Perspektive von Applikationsentwicklern vorgestellt (in Anlehnung an [7]).

Aus Sicht der Anbieter von Applikationen lassen sich zunächst zwei zentrale Komponenten einer Plattform voneinander abgrenzen: Auf der einen Seite interagieren App-Anbieter mit der *technischen Plattform*, auf der anderen Seite mit dem Marktplatz der Plattform als zentralem *Distributionskanal* für komplementäre Anwendungen [8]. Übertragen auf aktuelle Smartphoneplattformen umfasst die technische Plattform damit alle Aspekte, die im Zusammenhang mit der technischen Entwicklung von Applikationen stehen, wie bspw. APIs und SDKs, aber auch ent-

**Tabelle 32.1** Framework der vertikalen Plattformoffenheit

	Transparenz	Zugänglichkeit
Technische Plattform	Transparenz der technischen Plattform und der zugehörigen Lenkungsentscheidungen	Ausmaß, in dem die technische Plattform den Entwicklern die Erstellung von Applikationen ermöglicht
Distributionskanal	Transparenz des Distributionskanals und der zugehörigen Lenkungsentscheidungen	Ausmaß, in dem der Distributionskanal den Entwicklern die Verbreitung der Applikationen ermöglicht

sprechende Dokumentationen und Kommunikationstools wie Blogs oder Foren. In Abgrenzung dazu umfasst der Marktplatz einer Plattform (bspw. iTunes App Store oder Android Marketplace) alle Aspekte, die mit der Distribution der Applikationen verbunden sind. Neben der Gestaltung des Marktplatzes selbst fallen in diese Kategorie vor allem die Regelungen, die sich bspw. aus den allgemeinen Geschäftsbedingungen ergeben, sowie die Transparenz der Kommunikation zu diesen Regelungen.

Zusätzlich zu der Unterscheidung der beiden Komponenten einer Software-Plattform kann auch das Konzept der Offenheit selbst weiter differenziert werden. Im Kontext von Software-Plattformen werden dabei *Transparenz* und *Zugänglichkeit* als zentrale Dimensionen der Offenheit unterschieden [9]. Als *Transparenz* wird das Ausmaß verstanden, zu dem es (bestehenden und potenziellen) App-Anbietern ermöglicht wird, die Entwicklung und Distribution von Applikationen zu verstehen sowie über alle relevanten strategischen Veränderungen der Plattform informiert zu sein. Wichtige Aspekte der Transparenz sind also bspw. eine vollständige und verständliche Dokumentation der API, eine umfassende Informationspolitik zu kurz- und langfristig geplanten Weiterentwicklungen der Plattform oder der Grad, zu dem die Mechanismen des Marktplatzes gegenüber Applikationsentwicklern transparent gemacht werden. Die *Zugänglichkeit* einer Plattform bezieht sich dagegen auf die Möglichkeiten externer Entwickler, Applikationen tatsächlich erstellen und distribuieren zu können, ohne dabei plattformspezifischen Restriktionen unterworfen zu sein. Zentrale Aspekte sind hierbei bspw. der Funktionsumfang der API, die Höhe der Kosten, die direkt mit der Entwicklung (bspw. erforderliche Hard- und Software) oder der Distribution (bspw. Mitgliedsgebühren) von Applikationen einhergehen, oder mögliche Einschränkungen hinsichtlich des Inhalts einer Applikation, die sich aus den Regeln des Marktplatzes ergeben.

Die diskutierten Unterscheidungen, zwischen der technischen Plattform und dem Distributionskanal als Komponenten der Plattform einerseits und der Transparenz und Zugänglichkeit als Dimensionen des Offenheitsbegriffs andererseits, lassen sich zu einem gemeinsamen Framework der Plattformoffenheit zusammenführen [7].

Wie in Tabelle 32.1 dargestellt, ergeben sich insgesamt vier zentrale Dimensionen der vertikalen Plattformoffenheit. Dabei ist anzumerken, dass diese vier Dimensionen unabhängig aber additiv zur Offenheitswahrnehmung der Entwickler

beitragen. So ist bspw. vorstellbar, dass der Zugang zum Distributionskanal einer Plattform vollständig offen ist, gleichzeitig jedoch Restriktionen im Hinblick auf die technische Offenheit existieren (bspw. durch einen eingeschränkten Zugriff auf die Funktionalitäten der Plattform), oder dass eine Plattform zwar in hohem Maße transparent ist, die eigentliche Erstellung von Applikationen aber einschränkt ist (bspw. weil nur speziell zertifizierte Entwickler zugelassen werden).

### **32.3.2 Vertikale Plattformoffenheit aus Sicht von Business-App-Anbietern**

Entlang des vorgestellten Frameworks und unter Berücksichtigung der speziellen Eigenschaften von Business-App-Anbietern soll nun diskutiert werden, welche Bereiche der Plattformoffenheit für Business-Apps relevant sind und welche konkreten Aspekte dabei im Vordergrund stehen.

Hinsichtlich der **Transparenz der technischen Plattform** sind zunächst die *Dokumentation und der technische Support* zu nennen. Aufgrund der geforderten Stabilität der technischen Lösung erwarten Anbieter von Business-Apps eine umfassende Dokumentation und schnellen Support im Fall von technischen Problemen. Diesen Bedarf haben die Anbieter aktueller mobiler Plattformen erkannt und bieten entsprechend – neben den Entwicklerprogrammen für Einzelentwickler und Studenten – auch Zugänge zur Plattform für Unternehmen an. Diese typischerweise etwas höherpreisigen Zugänge (vgl. z. B. die Angebote von Apple und Microsoft) sind im Gegenzug mit umfangreicherem Support verbunden<sup>8</sup>. Eine weitere zentrale Anforderung resultiert aus der Dauer von Versionszyklen. Während mobile Plattformen sehr häufig aktualisiert werden, gelten Plattformen für traditionelle Business-Software als vergleichsweise stabil<sup>9</sup>. Anbieter von mobilen Business-Apps sind daher in besonderem Maße darauf angewiesen, dass der Plattformbetreiber neue Versionen und Änderungen der Plattformschnittstelle *rechtzeitig und transparent ankündigt*.

Auch aus dem Blickwinkel der **Zugänglichkeit der technischen Plattform** ergeben sich relevante Aspekte für Business-App-Anbieter. Business-Apps, die im Kern den mobilen Zugriff auf bestehende Software-Systeme wie bspw. ERP- oder CRM-Lösungen ermöglichen, sollen sich möglichst nahtlos in die bestehende Infrastruktur integrieren. Aus diesem Grund können sich *proprietary Standards*, wie bspw. die nur für Apple-Produkte eingesetzte Programmiersprache Objective-C, als nachteilig erweisen. Die Entwicklung und Wartung der damit entstehenden mobilen Insellslösung ist zumeist mit zusätzlichen Kosten verbunden. Ein weiterer wichtiger Aspekt besteht in der *Hardware-Fragmentierung* der mobilen Endgeräte einer

---

<sup>8</sup> Vgl. bspw. <http://developer.apple.com/programs/ios/enterprise/>.

<sup>9</sup> Während für die Android-Plattform seit der Einführung im Jahr 2008 bis heute 11 z. T. technisch erheblich veränderte Versionen erschienen sind, wurde bspw. WebSphere, eine von IBM angebotene Plattformlösung für Business-Applikationen, im selben Zeitraum nur einmal aktualisiert.

Plattform. Eine hohe Fragmentierung, wie bspw. im Fall der Android-Plattform, kann einerseits den Entwicklungsaufwand erheblich erhöhen, da die verschiedenen Hardware-Spezifikationen berücksichtigt werden müssen (bspw. Display-Größe, Verfügbarkeit von Tastatur oder Kamera etc.). Andererseits kann es jedoch für spezielle Lösungen auch von Vorteil sein, aus einer Vielzahl von Endgeräten auswählen zu können. Ein Beispiel hierfür wäre die Notwendigkeit eines besonders robusten Endgeräts („ruggedized“), bspw. für den Einsatz auf Baustellen.

Die Relevanz der **Transparenz des Distributionskanals** resultiert vor allem aus der durch die Plattformbetreiber eingenommenen Gatekeeper-Rolle. So ist bspw. für die iOS-Plattform festgelegt, dass Applikationen exklusiv nur über den iTunes App Store auf die Endgeräte distribuiert werden können. Für Anbieter von Business-Apps ist es daher besonders wichtig, dass alle Regelungen und Voraussetzungen zum Vertrieb einer App über den jeweiligen *Marktplatz transparent kommuniziert* werden. Beispiele, in denen ohne Angabe von schlüssigen Gründen die Annahme einer App verweigert wurde<sup>10</sup>, werden daher von Business-App-Anbietern besonders kritisch bewertet.

Eine Einschränkung der Distributionsmöglichkeiten auf einen bestimmten Markt-platz ist auch im Hinblick auf die **Zugänglichkeit des Distributionskanals** relevant. Da Business-Apps zumeist nur für eine vergleichsweise kleine Zielgruppe relevant sind (bspw. nur innerhalb einer bestimmten Firma), wäre für viele Apps eine direkte Distribution ohne Umweg über den Marktplatz der Plattform sinnvoller. Diesen Anforderungen kommen die Plattformbetreiber z. T. nach, indem Geschäftskunden die *Distribution innerhalb einer geschlossenen Benutzergruppe* (bspw. nur zertifizierte Geräte von Außendienstmitarbeitern eines Unternehmens) ermöglicht wird.

## 32.4 Horizontale Öffnung der Plattform

Im Gegensatz zu vertikaler Öffnung bezieht sich die horizontale Öffnung auf das Verhältnis verschiedener Plattformen zueinander. Aus Sicht der App-Anbieter folgt aus mehreren im Markt existierenden Plattformen die Fragestellung, welche der Plattformen als Grundlage für eigene Applikationen dienen soll. Ein zentraler Einflussfaktor auf diese Entscheidung ist der Grad der horizontalen Offenheit der Plattformen. Während ein hoher Grad von horizontaler Offenheit die Übertragung von Applikationen auf weitere Plattformen aus Sicht der App-Anbieter erleichtert, kann ein geringer Grad die Übertragung der Applikationen erschweren und damit zu höheren (Entwicklungs-) Kosten führen.

<sup>10</sup> Vgl. bspw. den durch Google für den iTunes App Store eingereichten Voice-over-IP-Dienst, dessen Annahme möglicherweise bewusst und in der Begründung intransparent hinausgezögert wurde.

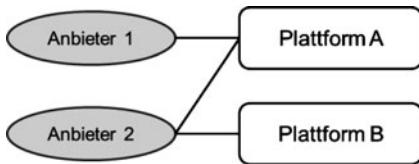


Abb. 32.1 Single- und multi-homing [10]

### 32.4.1 Single vs. multi-homing: Theoretischer Hintergrund

Wenn mehrere Plattformen in einem Markt existieren, können App-Anbieter ihre Applikationen sowohl für eine als auch für mehrere Plattformen anbieten. Abbildung 32.1 verdeutlicht das Szenario: Während App-Anbieter 2 auf beiden Plattformen aktiv ist (und sogenanntes multi-homing betreibt), bietet Anbieter 1 seine Applikation(en) nur auf Plattform A an (sogenanntes single-homing).

Multi-homing kann allgemein aus der Perspektive der App-Anbieter als Trade-off zwischen dem beschränkten (potenziellen) Transaktionsvolumen auf bereits bedienten Plattformen und den Kosten für die Erschließung einer neuen Plattform angesehen werden [11, 12].

In der Praxis spricht aus der Sicht der Business-App-Anbieter einiges für multi-homing: Zum einen können so mehr Endkunden erreicht werden. Zum anderen kann multi-homing die Verhandlungsposition gegenüber anderen Akteuren, wie bspw. dem Anbieter der Plattform, stärken. Dem entgegen steht, dass üblicherweise jeder Beitritt zu einer Plattform mit Kosten zur Überwindung der Eintrittsbarrieren verbunden ist, bspw. weil neuen Entwicklerprogrammen beigetreten werden muss oder bestehende Applikationen an neue Schnittstellen anzupassen sind. Ein hoher Grad an horizontaler Öffnung und die damit einhergehende Kompatibilität der Plattformen kann jedoch diese Kosten verringern. Damit ist horizontale Offenheit ein wichtiger Einflussfaktor auf die Multi-homing-Entscheidung der Business-App-Anbieter.

### 32.4.2 Horizontale Plattformoffenheit aus Sicht von Business-App-Anbietern

In diesem Abschnitt wird eine empirische Untersuchung zum multi-homing von App-Anbietern vorgestellt [13]<sup>11</sup>. Die Untersuchung wurde im Kontext von SaaS-

<sup>11</sup> Die Daten wurden mit einem Softwareframework automatisiert ermittelt. Für den Apps Marketplace wird aufgrund der Vergleichbarkeit in diesem Artikel nur die Kategorie „Products“ betrachtet. Die Untersuchung von Pinpoint beschränkt sich auf die US-amerikanische Version des Marktplatzes. Weiterhin werden aufgrund der Vergleichbarkeit nur die Applikationen betrachtet, die der Kategorie „Online Application“ zugeordnet sind.

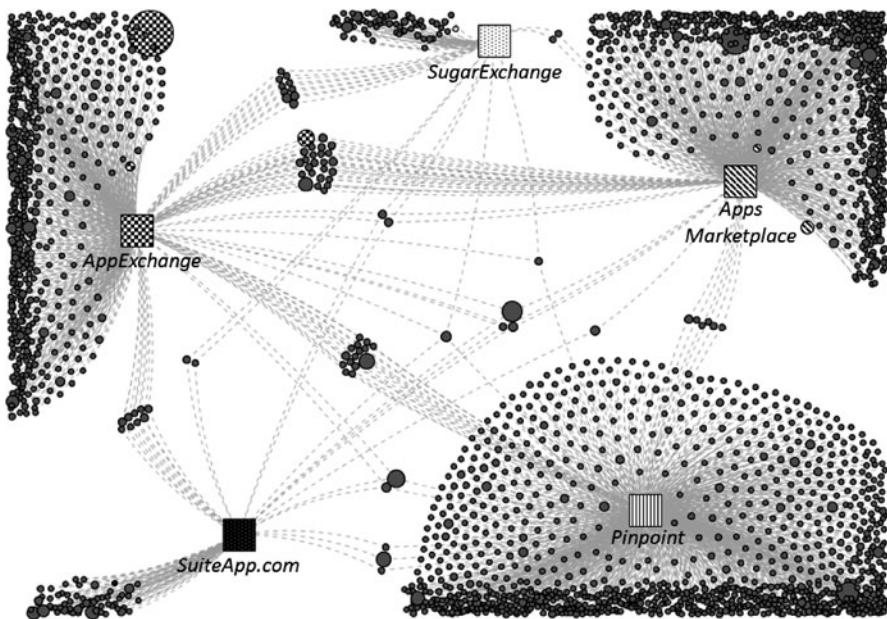


Abb. 32.2 Multi-homing im B2B-Umfeld (28.2.2011)

Geschäftsanwendungen durchgeführt, die Ergebnisse können jedoch auch aus Sicht von Anbietern mobiler Business-Apps interpretiert werden.

Ziel der Untersuchung war die Ermittlung anbieterseitiger Charakteristika verschiedener Business-Plattformen anhand öffentlich zugänglicher Daten. Dabei zeigt die Untersuchung die Verbindungen der Plattformen durch gemeinsame App-Anbieter (vgl. Abb. 32.2) als Folge der horizontalen Offenheit. Eine Verbindung in Abb. 32.2 bedeutet, dass ein App-Anbieter (Kreise) mindestens eine Applikation auf der Business-Plattform (Quadrate) anbietet. Abhängige App-Anbieter, d. h. App-Anbieter, die dem Plattformanbieter zugeordnet werden konnten (weil sie z. B. zum gleichen Gesamtkonzern gehören), sind mit dem jeweiligen Muster der Plattform markiert.

Die Größe der Kreise repräsentiert die auf allen Marktplätzen angebotene Applikationsanzahl.<sup>12</sup> Die Position der Kreise (bspw. die Nähe zu einem Quadrat) hat hingegen keine Bedeutung. Auffällig ist, dass lediglich ein abhängiger App-Anbieter (kleiner kariertes Kreis; *salesforce.com*) auf zwei Marktplätzen aktiv ist. Weiterhin ist ersichtlich, dass keiner der Anbieter Applikationen auf allen fünf untersuchten Marktplätzen bereitstellt. 73 Anbieter bieten Applikationen auf zwei Marktplätzen an. Lediglich sechs Anbieter sind auf drei und drei Anbieter auf vier Marktplätzen aktiv. Auch im Kontext mobiler Plattformen werden bspw. die Office-

<sup>12</sup> Je größer, desto mehr Nischenlösungen; der größte, karierte Kreis wurde aufgrund der sehr hohen Nischenlösungsanzahl verkleinert.

Applikation DocumentsToGo oder die Fernwartungsapplikation VNC Viewer auf mehreren Marktplätzen angeboten.

Die App-Anbieter haben vor dem Hintergrund des oben beschriebenen Trade-offs (Kosten für die Erschließung einer Plattform versus potenzielles Transaktionsvolumen) zu entscheiden, ob sie *eine oder mehrere Plattformen* bedienen. Dies ist insbesondere für mobile Anwendungen relevant, denn die Anwender betreiben in diesem Bereich typischerweise single-homing (d. h. kaum ein Endkunde verwendet mehr als ein Mobiltelefon). Dementsprechend kann ein App-Anbieter, der lediglich eine Plattform bedient, nur einen Teil der Konsumenten erreichen. Dies ist gerade für vergleichsweise „kleine“ App-Anbieter wichtig, da diese aufgrund der beschränkten Entwicklerkapazität typischerweise ihr Produkt nur für eine geringe Zahl von Plattformen anbieten können. Falls die App-Anbieter multi-homing anstreben, stehen sie zudem vor der Frage, in welcher Reihenfolge die Plattformen erschlossen werden sollen.

Es existieren bisher noch keine Studien zu der Frage, ob und bis zu welchem Ausmaß multi-homing für Anbieter mobiler Business-Anwendungen ökonomisch sinnvoll ist. Ein wesentlicher Einflussfaktor ist der vom Plattformanbieter angestrebte Grad der horizontalen Offenheit. Gerade für dominante Plattformanbieter ist jedoch Inkompatibilität (d. h. Reduktion der horizontalen Offenheit) und die dadurch entstehende Exklusivität eine Möglichkeit, die eigene Marktposition zu stärken. Multi-homing kann jedoch bspw. für neu im Markt existierende Plattformen von Vorteil sein, da so aus Endkundenperspektive wichtige Applikationen auf die eigene (neue) Plattform übertragen werden.

## 32.5 Fazit und Ausblick

Als zentrale Stellschrauben einer Plattform wurden die vertikale und horizontale Offenheit einer Plattform analysiert. Dabei wurde anhand praktischer Beispiele diskutiert, welche Implikationen sich aus Sicht der Anbieter von mobilen Business-Applikationen für die einzelnen Facetten der Offenheit von mobilen Plattformen ergeben.

Für die vertikale Offenheit wurde ein Framework der vier zentralen Dimensionen dieser Stellschraube vorgestellt. Darauf aufbauend wurden verschiedene Detail-Aspekte von mobilen Software-Plattformen wie bspw. die Dokumentation der technischen Schnittstellen oder die Transparenz der Kommunikation des Plattformbetreibers anhand aktueller Beispiele diskutiert. Hinsichtlich einer weitergehenden Diskussion von konkreten strategischen Implikationen hat sich jedoch gezeigt, dass die existierenden Erkenntnisse noch nicht ausreichen. Um bspw. konkrete Empfehlungen zur PlattformAuswahlentscheidung abgeben zu können, müssten vorhandene Erkenntnisse zu Facetten der Plattformoffenheit (vgl. bspw. [7]) zunächst auf die Zielgruppe der Business-App-Anbieter übertragen werden. Weitergehende Untersuchungen könnten darauf aufbauend für verschiedene Szenarien hinsichtlich der Attribute der verfügbaren Plattformen einerseits und der Anforderungen des Anbie-

ters andererseits konkrete Empfehlungen ableiten und bspw. mit Hilfe von Experten evaluieren.

Für die horizontale Offenheit wurden die theoretischen Hintergründe des multi-homing erläutert. Darauf aufbauend wurde eine empirische Untersuchung zum multi-homing im Kontext von Business-Apps vorgestellt und der Trade-off zwischen der Erschließung einer und mehrerer Plattformen aus Sicht der Business-App-Anbieter diskutiert. Ziel weiterer Forschung ist es, auf Basis der vorgestellten empirischen Ergebnisse [13] die bestehenden theoretischen Ansätze so zu erweitern, dass diese zur Entscheidungsunterstützung von Business-App-Anbietern beitragen können.

Business-Software-Anbieter, die zukünftig auch mobile Apps anbieten wollen, stehen vor den beiden zentralen Fragestellungen, auf welcher Plattform ihre Lösung angeboten werden soll und ob nur eine oder mehrere Plattformen berücksichtigt werden müssen. Ziel dieses Beitrags war es daher, die Parameter einer Plattform zu ermitteln, die diese Entscheidungen beeinflussen – auf Basis dieser Parameter gilt es nun, praxisnahe Handlungsempfehlungen für Business-App-Anbieter abzuleiten.

## Literaturverzeichnis

1. Wenzel S, Neumann S, Bandulet F, Faisst W (2010) Electronic Business Services and their Role for Enterprise Software. In: Software-as-a-Service, Benlian A, Hess T, Buxmann P (Hrsg) Gabler, Berlin, pp 75–91.
2. Moore JF (1993) Predators and Prey: A New Ecology of Competition. Harvard Business Review, pp 75–86.
3. Eisenmann TR, Parker G, Van Alstyne MW (2009) Opening Platforms: How When and Why. In: Platforms, Markets and Innovation, Gawer A (Hrsg) Edward Elgar, Cheltenham, pp 131–162.
4. Buxmann P, Diefenbach H, Hess T (2011) Die Software Industrie – Ökonomische Prinzipien, Strategien, Perspektiven. Springer, Heidelberg.
5. Chesbrough H (2003) Open Innovation: The New Imperative for Creating and Profiting from Technology. HBS Press, Boston
6. Greenstein SM (1996) Invisible Hands Versus Invisible Advisors: Coordination Mechanisms in Economic Networks. In: Public Networks, Public Objectives, Noam E, Nishui-leabhai A (Hrsg) Elsevier Science, Amsterdam
7. Hilkert D, Benlian A, Hess T (2011) The Openness of Smartphone Software Platforms. Erscheint in: INFORMATIK 2011, Berlin
8. Schlagwein D, Schoder D, Fischbach K (2010) Openness of Information Resources – a Framework-Based Comparison of Mobile Platforms. In: 18th European Conference on Information Systems (ECIS), Pretoria, pp 1–16
9. West J, O'Mahony S (2008) The Role of Participation Architecture in Growing Sponsored Open Source Communities. Industry & Innovation 15:145–168
10. Rochet JC, Tirole J (2006) Two-Sided Markets: A Progress Report. Rand J Econ 37:645–667
11. Lee RS (2009) Vertical Integration and Exclusivity in Platform and Two-Sided Markets. SSRN eLibrary
12. Rochet JC, Tirole J (2003) Platform competition in two-sided markets. Journal of the European Economic Association 1:990–1029
13. Burkard C, Draisbach T, Widjaja T, Buxmann P (2011) Software Ecosystems: Vendor-Sided Characteristics of Online Marketplaces. Erscheint in: INFORMATIK 2011, Berlin

---

# Kapitel 33

## Smarter Apps – Motor für Geschäftsmodellinnovationen

Thomas Goetz, Niels Feldmann und Sebastian Schmidt

**Zusammenfassung** Welche Möglichkeit zur Geschäftsmodellinnovation bieten neue Funktionalitäten von Smart Mobile Apps? Dieser Frage sind wir nachgegangen und haben gesehen, dass geolokalisierte Dienste, Augmented Reality und Social-Networking-Komponenten einige Optionen sein können, die es Firmen ermöglichen, ihr Wertversprechen und weitere Teile ihres Geschäftsmodells erfolgreich zu transformieren. Im Kern der Analyse steht die strukturelle und auf Praxisbeispiele angewandte Betrachtung zahlreicher Geschäftsmodellelemente. Es folgt ein Exkurs in den Bereich Mobile Commerce sowie ein Ausblick, der die Welt der Smart Mobile Apps mit dem „Internet der Dinge“ verbindet und somit die Integration von Smarter Apps in Alltagsgegenstände diskutiert. So rückt schon heute die „intelligente Kleidung“ in den Raum des Möglichen, indem Sensoren in die Nähte von T-Shirts integriert werden.

---

Thomas Goetz  
IBM Deutschland GmbH – GBS Strategy and Transformation Consulting  
Partner, Köln,  
E-mail: Thomas.goetz@de.ibm.com

Niels Feldmann  
IBM Deutschland GmbH – GBS Strategy and Transformation Consulting  
Senior Managing Consultant, Frankfurt,  
E-mail: Niels.Feldmann@de.ibm.com

Sebastian Schmidt  
IBM Deutschland GmbH – GBS Strategy and Transformation Consulting  
Senior Consultant, München,  
E-mail: Sebastian.Schmidt@de.ibm.com

### 33.1 Apps reloaded?

„Benötigen wir eine App?“, „Wenn ja, welche?“ und „Welche Chancen und Risiken stellen Apps für mein Geschäftsmodell dar?“ – Dies sind einige der Fragen, die uns derzeit zahlreiche Verantwortliche aus den unterschiedlichsten Industrien stellen.

Dabei erscheinen die Fragen altbekannt: In den 90er Jahren wurde nach der Notwendigkeit eines Webauftritts gefragt und etwas später nach der Bedeutung von Foren für die Nutzer-Interaktion. Die Diskussion über einen Social-Media-Auftritt ist noch brandaktuell, da entwickeln bereits zahlreiche Unternehmen eine große Auswahl an Mobile Apps für Smartphones und Tablet PCs.

In allen Fällen handelt es sich um Technologieoptionen, durch die Plattformen für Anwendungen um mehr oder weniger umfangreiche Funktionen erweitert werden. Auch die gegenwärtig diskutierten Smart Apps sind gewissermaßen nur eine Fortschreibung dieser Linie: Anwendungen werden auf zunehmend mobilere Plattformen portiert, die darüber hinaus einige neue Zusatzfunktionen wie z. B. geolokalisierte Dienste bieten. Doch auch die heute diskutierten Apps für Smartphones und Tablet PCs sind nur ein Zwischenschritt: Am Horizont zeichnet sich bereits das „Internet der Dinge“ ab, das eine Portierung von Anwendungen auf Alltagsgegenstände mit wiederum erweiterten Funktionalitäten, wie zum Beispiel im Bereich der Sensorik und Datenauswertung, mit sich bringen wird.

Somit stellt sich die Frage nach der Einordnung der aktuellen Mobile Apps und ihren Einflussmöglichkeiten auf Geschäftsmodelle – insbesondere in etablierten Industrien:

- Durch welche zentralen Neuerungen zeichnen sich Smartphones / Tablet PCs als Plattform für Mobile Apps aus?
- Welche Arten von Apps lassen sich hinsichtlich der Nutzung neuer Potentiale identifizieren?
- Auf welche Elemente eines Geschäftsmodells haben Mobile Apps den größten Einfluss?
- Welche Bedeutung haben Mobile Apps für die Entwicklung hin zu neuen Plattformen?

Diesen Fragen gehen wir im Folgenden aus der Perspektive von Apps für mobile Endgeräte wie Smartphones und Tablet PCs nach.

### 33.2 Zentrale Neuerungen: Das Paradigma der Smart Mobile Apps heißt „Always connected“

Moderne Smartphones oder Tablets beinhalten in Folge der rasanten Technologie-Entwicklung bei Hardware und Software zunehmend Funktionalitäten, die vor einigen Jahren nur auf Desktop-PCs oder Laptops verfügbar waren, und werden dank intuitiver Bedienung für Nutzer jeder Altersgruppe nutzbar. Dabei sehen wir vier Kernfaktoren, welche das „Eco-System“ der Apps wesentlich prägen:

- **Uneingeschränkte Mobilität in Kombination mit ausreichender Rechenleistung und hochauflösenden Displays** – Zeit- und ortsunabhängige Benutzung reichhaltiger Funktionalitäten auf den Endgeräten mit Zugang zum Internet und Inhalten des World Wide Web. Laut einer Studie zum mobilen Nutzerverhalten verwendeten im Jahre 2010 bereits 11 % der Deutschen ein Smartphone mit deutlich steigender Tendenz [1]. Wer sich heutzutage ein Smartphone zulegt, tut dies in den seltensten Fällen ohne einen mobilen Datenvertrag hinzuzubuchen. Der Schritt in eine „Always-Online“-Gesellschaft vollzieht sich erst jetzt nach und nach vollständig, seitdem der Einsatz mobiler Endgeräte einen permanenten Webzugang mit leistungsfähigen Browsern und intuitiven Benutzer-Interfaces möglich gemacht hat.
- **Geo-Lokalisierung als Enabling Service** – Universelle Verwendbarkeit von Global Positioning System (GPS)-Daten für alle Arten von Applikationen wie z. B. einer ortsbezogenen personalisierten Empfehlung zu Events oder Angeboten, zum Teil mit Social-Networking-Elementen.  
Findet ein Autofahrer nicht auf Anhieb die nächste Tankstelle, kann er sich diese nun mit Hilfe einer App wie z. B. „Clever Tanken“ auf einer Karte anzeigen lassen. Die App „Around Me“ geht sogar noch einen Schritt weiter, indem sie Augmented-Reality-Elemente mit den Geo-Diensten verbindet. Durch einen Sensor schaltet das Smartphone hierbei automatisch die Kamera an und legt über das Sichtfeld des Nutzers die nächstgelegenen Tankstellen, sobald der Nutzer das Telefon von der vertikalen in die horizontale dreht.
- **„Weiche“ Echtzeit-Fähigkeit in Kombination mit Social-Network-Integration** – Sehr geringe Latenz im Rahmen eines Informations- oder Transaktionsprozesses, und damit steigende Geschwindigkeit der bilateralen und multilateralen Nutzerkommunikation, -interaktion und -lebensweise. Wir unterscheiden zwei Arten von Datenquellen. Zunächst können Informationen über ein bestimmtes Umfeld in Echtzeit von einem Anbieter bezogen werden (one-way). Beispielsweise können die Nutzer die Preise der angezeigten Tankstellen einer App vergleichen, Zugverspätungen auf der Anreise zum Bahnhof per Push-Notification erhalten oder sich die Strecke eines vorbeifliegenden Flugzeugs anzeigen lassen. Auf der anderen Seite können Informationen über ein persönliches Netzwerk zwischen mehreren Parteien ausgetauscht werden. Diesen verstärkten Einsatz von *Social-Network-Komponenten* sehen wir als einen weiteren wesentlichen Treiber für die zukünftige Entwicklung von Apps. Die Verbreitung dieser Funktionalitäten geht bereits so weit, dass Endgerätehersteller diese Social-Networking-Elemente in ihr Betriebssystem einbeziehen und damit dem App-Entwickler und dem Kunden „vom Werk aus“ zur Verfügung stellen [2]. Derzeit nutzen rund 250 Millionen Menschen die Facebook-App für Smartphones. Interessant hierbei ist, dass Smartphone-User doppelt so häufig auf Facebook zugreifen wie reine Desktop-User [3]. Der ursprüngliche reine „Always-Online“-User wird unter dem Einfluss von Social Media Teil einer neuen „Always-Connected“-Community.
- **Vorintegrierte Entwicklungs- und Produktionsplattformen mit geringen Eintrittsbarrieren** – Für Entwickler spielt die Verbreitung der Zielplattform die

entscheidende Rolle, wenn sie sich auf ein Endgerät festlegen. Andere Faktoren, wie Qualität der Entwicklungstools, Einarbeitungszeit, geringe Entwicklungskosten müssen sich zwar in den meisten Fällen dem unterordnen, werden aber ebenfalls immer weiter optimiert [4]. Entwicklungsumgebungen für Apps haben mittlerweile ein Niveau erreicht, welches Einzelpersonen oder kleinen Teams mit einer „early beta“-Kultur ermöglicht, kreative Gedanken unmittelbar in lauffähige Apps umzusetzen. Ein gutes Beispiel dafür sind neuerdings zu beobachtende „App-Development Jams“ (z. B. bei innovativen Versicherungsunternehmen), analog zu den bekannten „LAN-Parties“ der Gaming-Generation. Im Rahmen von freiwilligen Wochenendaktivitäten mit Party- oder Grillfestcharakter wird kreativ an möglichen Ideen zu Apps gearbeitet und prototypisch in einem agilen, iterativen Entwicklungsprozess eine erste App-Version bereits an einem Tag umgesetzt. Für die Gestaltung der Benutzeroberflächen wird dabei oft auf bereits bestehende Corporate Identity Templates zurückgegriffen.

Die Kombination dieser vier Kernfaktoren verstärkt die Entwicklung neuer Leistungsmöglichkeiten für mobile Plattformen. Jedes Unternehmen muss sich daher die Frage stellen, wie und wo es in diesem Umfeld gegenüber bestehenden und potentiellen Neukunden präsent sein möchte. Smart Mobile Apps können hierbei einen Weg darstellen, diese neue Welt zu erschließen.

### 33.3 Arten von Mobile Apps hinsichtlich der Nutzung neuer technologischer Möglichkeiten

Werfen wir nochmals einen Blick in die Vergangenheit: Bei der Etablierung jeder neuen Plattform konnten wir verwandte Entwicklungen bzgl. der auf ihr eingesetzten Applikationen beobachten.

In der **ersten Phase** können vier Kernelemente identifiziert werden. Zunächst erfreuen sich kleine Anwendungen großer Beliebtheit, die das neue Medium spielerisch erkunden lassen. Es erfolgt somit eine erste *Signalfunktion*, die Aufmerksamkeit auf das Gerät lenkt. Beispiele hierfür sind die „Augen“, die dem Mauszeiger auf dem Bildschirm folgen, das Glas Bier, das sich virtuell auf dem iPhone trinken oder die Kugel, die sich dank Accelerometers des iPhones durch ein Labyrinth manövrieren lässt. In den seltensten Fällen stammen diese Apps von Unternehmen, sondern zumeist von einzelnen Entwicklern oder direkt von dem Hersteller des jeweiligen Endgeräts. Es wäre sicherlich übertrieben, bei diesen „Spielereien“ bereits von damit verbundenen Geschäftsmodellen zu sprechen. Dennoch sind einzelne Funktionen bereits hier wichtig, um Impulse für neue Geschäftsmodelle zu geben. Daneben erfüllen diese Apps bereits eine *Marktöffnungsfunktion*, indem bestimmte Bedürfnissegmente mit dem neuen konkreten oder auch nur angedeuteten Wertversprechen der App adressiert werden. Insbesondere im Gaming-Sektor finden sich hierzu Anwendungen – positive Erlebnisse werden schnell weiter im Netz verbreitet, und durch die Integration von HighScore-Bestenlisten in einem Online-

netzwerk wird die eigene „Reputation“ in der eigenen Community gesteigert. In dieser ersten Phase können Apps dementsprechend eine *Bindungsfunktion* für den User erzeugen. Grundsätzlich haben aber diese Apps der ersten Stufe vor allem auch eine *Inspirationsfunktion* – und das sowohl für Entwickler wie auch User, um anhand diverser Funktionalitäten der jeweiligen Plattformen den nächsten Schritt hin zu seriösen Applikationen zu finden.

Im **zweiten Schritt** erobern diejenigen Apps den Markt, die Gewohntes in die neue Welt übertragen: Schon früh konnte die Zeitung auch online gelesen werden, inzwischen trägt man sie als App auf dem Smartphone oder Tablet mit sich herum. Dies zeigt, dass etablierte Anbieter ihre Produkte modifizieren/digitalisieren und diese über einen neuen Kanal einer zum Teil neuen Zielgruppe zugänglich machen, was eine inkrementelle Geschäftsmodellinnovation darstellt. In dieser Phase werden alle Möglichkeiten ausgelotet, die einzelnen Komponenten etablierter Geschäftsmodelle durch Einsatz von Apps punktuell zu verbessern oder moderner zu gestalten, ohne jedoch den Charakter des Geschäftsmodells zu ändern. Aus dieser Phase heraus werden auch vielfältige Forderungen nach neuen Funktionalitäten und Qualitäten mobiler Endgeräte getrieben, um die bewährten und akzeptierten Elemente der traditionellen Geschäftsmodelle ohne Abstriche auch in der neuen App-Welt zu nutzen. Beispiele hierfür sind bei Tageslicht gut lesbare Displays, verbesserte Benutzerführungen, Aufrüstung besserer Akustik-Fähigkeiten für Aufnahme und Wiedergabe, Anpassung von Formfaktoren, etc. Auch gleichen sich die Optiken und Benutzerführungen der Apps immer weiter an die etablierten, in den Konsumentenmärkten und der Unterhaltungselektronik geprägten Verfahren an, womit auch hier dann „de-facto“ Standards gesetzt werden.

In einer **dritten Stufe** eröffnen neue „strategische Enabler“, z. B. geo-lokalierte Empfehlungen in „weicher“ Echtzeit (z. B. Foursquare, ImmoWelt), gänzlich neue Möglichkeiten einer Mobile App und helfen dabei, neuartige und teilweise disruptive Geschäftsmodellinnovationen hervorzubringen. Ein historisches Beispiel für eine disruptive Geschäftsmodellinnovation durch strategische Enabler ist Amazon. Bewertungsmöglichkeiten von Nutzern, personalisierte Empfehlungslisten (Stichwort „Recommendation Engine“) und ein schneller Zugriff auf ein umfangreiches Sortiment haben das Unternehmen zum erfolgreichen Online-Buchhändler und Handelsunternehmen werden lassen. Dabei wurde das Internet als Kanal, Wertschöpfungskette und Bindungselement geschickt in allen Komponenten eines Geschäftsmodells auf eine einzigartige, kaum imitierbare Art und Weise genutzt. In dieser dritten Phase können also durch Apps radikale Innovationen und unter Berücksichtigung der in Abschn. 33.2 beschriebenen Kernfaktoren vollständig neue Geschäftsmodelle entstehen. Hierbei können neue Technologien als strategische Enabler voll ausgeschöpft werden und damit ein signifikanter Zusatznutzen für den User geschaffen werden. In einer solchen Phase ist wiederholt mit neuen Geschäftsmodellen zu rechnen, die den sog. „game changer“-Charakter haben, also etablierte Muster erfolgreicher Geschäftstätigkeit verlassen und die Spielregeln von Branchen in Frage stellen, teilweise auch branchen- und geschäftsmodellübergreifende neue Märkte schaffen.

Wie bereits bei der Verbreitung des Internet und der ersten Phase disruptiver Geschäftsmodellinnovationen gilt es auch jetzt wieder, sich genau zu überlegen,

ob und wie man sein Unternehmen auf diesem neuen Feld positionieren könnte, um Chancen und Risiken erfolgreich aufnehmen zu können. Mit anderen Worten – das eigene Geschäftsmodell muss im Kontext der rasanten Entwicklung des „App-Ecosystems“ gegebenenfalls neu bewertet und überarbeitet werden.

### 33.4 Modellierung und Anpassung digitaler Geschäftsmodelle

*„Bis 2015 sollen E-Commerce-Umsätze, die aus Apps heraus getätigten werden, auf knapp 2,4 Milliarden Euro anwachsen – das würde einem Wachstum von 45 Prozent pro Jahr entsprechen.“ [5]*

Der Begriff des Geschäftsmodells wird in der Literatur mehrfach unterschiedlich definiert, doch es herrscht bis heute kein einheitliches Bild über eine allgemein gültige und genaue Definition. Unserer Meinung nach sollte jedoch der Begriff des Geschäftsmodells als Orientierungshilfe in jedem Fall dazu dienen, Schlüsselfaktoren für den Erfolg eines Unternehmens in der neuen App-Welt zu klären. Eventuell kann hierbei bereits ein einziger Hebel entscheidend sein, um den gewünschten Erfolg in einem neuen Kundensegment zu erzielen (z. B. eine neue Kanalwahl über eine App). Beispielsweise kann der Vertrieb von digitalen Medien wie Musik, Filmen oder Büchern mit Hilfe von neuen Abo-Preismodellen zu einem erhöhten Umsatz über eine App führen. Werden Spiele-Apps vertrieben, können neben dem Preis der App vor allem virtuelle Güter, z. B. „Ausrüstungen“ oder Modellkomponenten, den Umsatz steigern.

Damit keiner dieser Hebel unberücksichtigt bleibt, sollte ein Geschäftsmodell einige wesentliche Elemente enthalten, um das betriebliche Leistungssystem eines Unternehmens (inklusive einer klaren Value Proposition wie auch Kunden- und Kanalstrategie) in seinen Erträgen und Aufwendungen abzubilden. Eine gute Möglichkeit diese Elemente strukturiert zu betrachten bietet das „Canvas“-Geschäftsmodellraster nach Alexander Osterwalder und Yves Pigneur, welches wir auch für die Betrachtung von Apps empfehlen.

Das Geschäftsmodell-„Canvas“ besteht dabei aus 9 Kernsegmenten, die im Folgenden für die Anwendung im Mobile-App-Bereich und illustriert anhand von Beispielen aus der Verlagsbranche analysiert werden sollen.

Im Kern des Modells steht der Bereich des *Wertversprechens*. Dieser beinhaltet Lösungen sowie Vorteile (auf unterschiedlichen Ebenen wie z. B. Leistungsfähigkeit, Handhabbarkeit, Design etc.) für das jeweilige Kundensegment. Typische Fragestellungen im Verlagswesen könnten sein: Welchen zusätzlichen Nutzen liefern wir dem Kunden mit digitalen Inhalten? Wie können wir einem generellen Trend zur Entwertung digitaler Inhalte begegnen? Wie können wir eine neue Customer Experience schaffen? Welche Anpassungen sind hierfür bzgl. Design und Benutzerführung notwendig?

Verlage haben ihre digitalen Inhalte zum Teil durch Videos oder eine interaktive Gestiknavigation für Tablets verbessert aufbereiten können.



Abb. 33.1 Businessmodellelemente nach Canvas. Quelle: Osterwalder A, Yves P (2009); IBM

Ein besonders hervorzuhebendes Beispiel, welches der generellen Entwertung digitaler Inhalte entgegensteht, ist das im April 2011 von Al Gore veröffentlichte iPad-, iPhone- und iPod-touch-App „Fachbuch“ mit dem Titel „Our Choice“. Dank zahlreicher interaktiver Elemente in Form von Bildern, Videos und Audiobeiträgen sowie der intuitiven Multitouch-Gestikbedienung wird das Lesen der Beiträge zu einem besonderen Erlebnis. Diese App brachte dem Entwicklungsteam den Gewinn des Apple Design Awards für 2011 ein und könnte die Art und Weise, wie zukünftig digitale Bücher erstellt werden, grundsätzlich verändern. In diesem Zusammenhang sei jedoch erwähnt, dass Nutzer vermehrt personalisierte Fragmente einer „digitalen Zeitung“ konsumieren. Durch die Möglichkeiten von RSS-Readern, den Einsatz von Aggregations-Algorithmen (Google News) und Social Networks sowie durch eine „Read-it-later“-App auf dem Smartphone verlagert sich die Zusammenstellung der Nachrichten vom Verlag zunehmend zum Leser. Inhalte können somit fokussierter und schneller konsumiert werden, was dem Kunden einen zusätzlichen Mehrwert schafft. Ein Beispiel hierfür ist die Flipboard-App für das iPad, die Nachrichten mit Facebook-Neuigkeiten und Tweeds verknüpfen kann.

*Kundensegmente* können anhand einer Vielzahl von Kriterien – wie z. B. Alter, Ort oder Anwendungszweck – beschrieben werden und machen deutlich, ob ein Massen- oder eher ein Nischenmarkt mit einer App erschlossen werden soll. Alle Segmente sollen optimal aus dem Wertversprechen heraus adressiert werden können. Um die Segmente klar definieren zu können, sollte man sich die folgenden typische Fragen stellen: Wer sind unsere wichtigsten Kunden? Für wen wollen wir Werte oder Nutzen schaffen? Welche Altersgruppen wollen wir mit unserer App erreichen?

Ein Blick auf die unterschiedlichen *Kanäle* des Geschäftsmodells kann zeigen, wie bestimmte Kundensegmente allgemein aktuell mit dem Wertversprechen erreicht werden können und welche Kanäle die unterschiedlichen Kundensegmente

wiederum für die Zukunft erwarten. In unserem Beispiel stellt die Mobile App einen eigenen Kanal dar, der neue Transaktions- und Informationsmöglichkeiten schafft. Allerdings sollte eine App von Beginn an ihren Platz in einer größeren Multikanalstrategie finden und hierdurch zusätzlichen Mehrwert für den Kunden generieren. Verlage können somit beispielsweise Abonnenten der Druckausgabe eine vergünstigte oder kostenfreie digitale iPad-Version anbieten.

Im Bereich der *Kundenbeziehungen* können Smarter Apps dabei behilflich sein, diverse Formen der Interaktion mit dem Nutzer neu zu gestalten. Automatisierte Personalisierung, User-generated Content und gemeinschaftliches mobiles Ideenmanagement sind Beispiele für aktuelle Diskussionen zu diesem Geschäftsmodelllement. Im konkreten Fall des Verlagsmanagements könnten hierzu interaktive Dialogelemente in eine App integriert werden, über die Leser mit den Autoren einer Zeitschrift Kontakt aufnehmen können. Leser hätten die Option, Feedback bzw. Leserbriefe einzureichen, einen Artikel weiterzuschreiben (Co-Creation) oder neue Themen vorzuschlagen, über die sie gerne lesen würden. Social-Networking-Elemente würden einen fachlichen Austausch über bestimmte Themen auf der Plattform des Verlages entstehen lassen.

*Wichtige Ressourcen* im Geschäftsmodell geben an, welche Elemente benötigt werden, um das jeweilige Wertversprechen zu erfüllen und die gewünschten Beziehungen und Kanäle einhalten zu können. Im Rahmen der Apps könnten hier beispielsweise Software Development Kits (SDKs) gesehen werden, die man bei den jeweiligen Betriebssystemherstellern downloaden und darauf die gewünschte App entwickeln kann.

*Hauptaktivitäten* beinhalten diejenigen Aktivitäten, die besonders wichtig sind, um das Wertversprechen zu erfüllen, die Beziehung aufrecht zu erhalten und die Kanäle benützen zu können. Für den Einsatz von Apps ist auch hier ein aktiv gesuchter Dialog mit den Usern ein bedeutendes Element, um weitere Verbesserungen aufzunehmen. Ein gutes Beispiel hier ist die App „Viber“, die Voice-over-IP (VoIP)-Dienste für das iPhone und Android-Telefone anbietet. Die Betreiber stellen wöchentlich Fragen an ihre Facebook-Community, um neue gewünschte Funktionalitäten oder bestehende Schwachstellen der App zu identifizieren. Gleichzeitig dient ihnen Facebook als Servicekanal, über den User Störungen melden können.

Daneben muss im Geschäftsmodell die Ebene der Partner betrachtet werden. Wie kann beispielsweise das *Partnernetzwerk* genutzt werden, um Leistungen und Inhalte von Partnern zu integrieren, diverse Aktivitäten und Support extern zu vergeben oder enger mit Lieferanten zusammenzuarbeiten? So wurde beispielsweise die App „Gesund & Fit“ zur Lokalisierung von Beschwerden und deren Behandlungsmöglichkeiten in Kooperation von Onmeda und der Süddeutschen Krankenversicherung (SDK) entwickelt, um Kosten zu sparen und Erfahrungen aus der bisherigen App-Entwicklung zu teilen.

*Umsätze* ergeben sich im Geschäftsmodell aus den Bestandteilen des Wertversprechens, der Kundenbeziehung und Segmente bzw. der Anzahl der Kunden und der jeweiligen Preisgestaltung. Wichtige Fragen könnten sein: Für was zahlen die Kunden heute und für was sind sie bereit, darüber hinaus zu bezahlen? Wie zahlen

sie heute, und wie würden sie gerne zahlen? Speziell zu der zweiten Frage lassen sich mobile Bezahlsysteme mit Apps und Near-Field-Communication (NFC)-Technik bestens verbinden. Leser einer digitalen Zeitung auf dem iPad könnten so eventuell in Zukunft bereits gelesene Exemplare ihren Kollegen zu deutlich reduzierten Preisen weiterverkaufen, indem sie die Geräte ad hoc drahtlos verbinden. Wir gehen davon aus, dass sich in Zukunft mobile „Tauschbörsen“ entwickeln werden, über die virtuelle Güter gehandelt werden können. Der Einsatz digitaler personalisierter Werbung könnte den Verlagen zusätzlichen Umsatz generieren. Allerdings brauchen Marketingagenturen oder die Verlage hierfür gewisse Nutzerdaten, um festlegen zu können, welcher Inhalt zu welcher Zeit an welchem Ort erfolgreich wäre. Diese Informationen würden über die Hardware erfassbar sein, müssten dann aber in Echtzeit mit dem „Werbungspool“ abgeglichen werden (falls der Nutzer der Weiterleitung dieser Daten zustimmt).

All die genannten Elemente der Leistungserbringung und deren Generierung stellen den Kern der *Kostenstruktur* im Geschäftsmodell dar. Hierzu zählen die Kernaktivitäten, Ressourcen sowie Businesspartner. An diesen Elementen kann analysiert werden, welche Bestandteile die höchsten Kosten in einem Geschäftsmodell erzeugen. Verlage könnten sich durch den Einsatz einer App Teile ihrer Aufwände für Druck und Vertrieb sparen, müssten allerdings, wenn sie beispielsweise über etablierte App-Stores gehen, bis zu 30 Prozent des Umsatzes an den App-Store-Betreiber abführen (vgl. [6]). Ferner kann eine multimediale Aufbereitung der digitalen Inhalte in der Kostenstruktur leicht explodieren, wenn es darum geht, mehrere Smartphone- bzw. Tablet-Betriebssysteme (Plattformen) nativ zu unterstützen. Hier könnten Lösungen mit Entwicklungspartnern oder auf HTML5-Basis eine Alternative darstellen. Bei der Auswahl einer bestimmten Plattform sollte neben der Nutzer-Reichweite vor allem auch deren Innovativität bzw. Möglichkeiten geprüft werden. Falls die Wahl auf eine HTML5-basierte Lösung fällt, würde der im System integrierte Browser als App-Plattform dienen. Somit könnten Apps den Nutzern verschiedener Plattformen zugänglich gemacht werden. Kosten für einen App-Store würden damit umgangen werden. Dennoch fallen beispielsweise laufende Kosten in der Redaktion für die Pflege der Inhalte wie auch Updates der App an.

Zusammengefasst ergibt sich aus der differenzierten Betrachtung der Geschäftsmodellstruktur im App-Kontext ein völlig neues Spektrum von Möglichkeiten. Ein ganz wesentlicher Aspekt der Potentiale neuer Apps liegt in der neuartigen Interpretation des sog. „Kunden-Lebenszyklus“ und der Gestaltung der Customer Experience. Erstmals kann ein Konsument und Kunde einen „eigenen“, persönlichen Anbieter, Provider, Lieferanten, Partner etc. „mit sich tragen“ und eine nachhaltig dauerhafte Beziehung zu einer Marke aufbauen – das Smartphone als alltäglicher Begleiter in der Telefonie hat sich endgültig gewandelt zum omnipräsenten Begleiter im Leben, welcher die Leistung einer Vielzahl von Unternehmen im täglichen Gebrauch immer wieder in kurzen Abständen neu erlebbar macht. Die werthaltige und verantwortungsvolle Gestaltung dieser neuen Möglichkeiten dürfte eine der herausfordernden Aufgaben des modernen Marketings sein.

### 33.5 Beispiel einer Anwendung von Mobile Commerce (mCommerce) in der Bekleidungsindustrie

Mobilität und die persönliche Erreichbarkeit sind, wie oben beschrieben, menschliche Grundbedürfnisse. Sobald Unternehmen diese Bedürfnisse mit den Möglichkeiten moderner Informations- und Kommunikationstechnologien speziell in Form von Apps bedienen, können neue Märkte entstehen. Einen Weg kann „mCommerce“ (mobile Commerce) darstellen, welcher die Initiierung und Durchführung von Informations- und Transaktionsprozessen über mobile Endgeräte erfasst. Diese Funktionen sollten unmittelbar und ortsunabhängig erfolgen können. Eine zusätzliche Integration von Personalisierungs- und Lokalisierungsaspekten erhöht dabei die Möglichkeiten von mCommerce. Somit erweitert mCommerce das klassische eCommerce um die Aspekte der Mobilität, der Geo-Lokalisierung und der unmittelbaren Anwendung. Unter Nutzung des neuen Netzstandards LTE und der damit verbundenen höheren mobilen Durchsatzraten werden sich für mCommerce in naher Zukunft nochmals zusätzliche Möglichkeiten ergeben. Schauen wir uns nun ein konkretes fiktives Beispiel für den Bereich Retail und im Speziellen in der Bekleidungsindustrie an:

*Eine Kundin sucht die Filiale eines Bekleidungshauses auf, um sich über die Trends der aktuellen Saison zu informieren. Im Shop angekommen, probiert sie die ersten Kleidungsstücke an. Nun sind leider die gewünschten Größen der Artikel bereits vergriffen. Die Kundin greift zu ihrem Smartphone und startet das App des Bekleidungshauses. Sie scannt den Barcode des Artikels und bekommt ihn unmittelbar in ihrer App zur Bestellung angezeigt. Sie wählt die gewünschte Größe und lässt ihn sich entweder in den Laden oder direkt nach Hause liefern. Die Bankdaten der Kundin können entweder über ein Profil in der App hinterlegt sein oder, falls der Kundin diese Daten für eine Übermittlung zu sensibel erscheinen, kann sie die Bestellung direkt vor Ort bezahlen.*

Im Rahmen einer umfangreichen Multikanalstrategie des Bekleidungshauses ist im Nachhinein auch ein eventueller Umtausch der nach Hause gelieferten Produkte in jeder Filiale möglich. Dank Geo-Lokalisierungsfunktion wird über die App die jeweilige Bestellung / Provision der jeweiligen Filiale direkt zugeschlüsselt. Bei Bestellungen von anderen Orten aus wird nach der jeweiligen Filiale, in der der Artikel gesehen wurde, gefragt. Würden solche Funktionen von den Kunden genutzt werden? Der Einsatz von Touchscreenterminals mit ähnlichen Funktionen wurde bisher zu wenig beworben oder noch nicht optimal im Laden platziert. Hier könnten diverse Marketinginitiativen im Shop nochmals auf die jeweilige App verweisen, die dank kostenfreiem WLAN-Zugang auch direkt im Laden heruntergeladen und bedient werden könnte. Vorteile für den Kunden würden darin bestehen, alle Produkte in allen Größen direkt vor Ort bestellen zu können, nachdem diese „erlebt wurden“. Vor allem größere Bestellungen müssten nicht transportiert werden und könnten bequem nach Hause geliefert werden. Eine Integration von Kundenbindungsaspekten in der jeweiligen App (z. B. Wertgutschein bei mehr als drei Bestellungen pro Filiale oder eines bestimmten Sortiments bzw. Add-On-Zufallsprodukt via Schüttel-

funktion bei einer Vor-Ort-, „In-Shopbestellung“ über 50 Euro etc.) stellt eine häufige Verwendung der App sicher. Produkte/Sortimente könnten vor Ort mit Freunden via Facebook geteilt werden und Meinungen zum „Style-Faktor“ und der persönlichen „Coolness“ eingeholt werden. Vorteile für das Unternehmen beständen in potentiell geringeren Lagerkosten, da nicht alle Produkte in allen Größen permanent vorrätig sein müssten. Hierdurch könnten Ausstellungsflächen verkleinert bzw. effektiver genutzt werden, z. B. für eine Ausweitung des Sortiments. Durch die oben genannten In-App-Funktionalitäten könnten Kunden zusätzlich an die eigene Marke und die jeweilige Filiale gebunden werden. Durch die Bestellung des jeweiligen Produktes in der aktuell vor Ort nicht vorhandenen Größe könnten weitere Umsätze generiert werden, die ansonsten nicht realisierbar wären. Die Nutzung von Social-Media-Elementen seitens der Kunden könnte dazu beitragen, das eigene Sortiment einer größeren Zielgruppe vorzustellen. Diese persönlichen Empfehlungen hätten darüber hinaus einen größeren positiven Einfluss auf Peer Groups als „Standard“-Marketinginitiativen des Unternehmens via Print, TV und digital. Daneben könnten dabei die eigenen Marketingausgaben gesenkt werden.

### **33.6 Die Bedeutung von Mobile Apps für die Entwicklung hin zu neuen Plattformen**

Heute steht eine Vielzahl an Unternehmen vor der Entscheidung, ob und in welcher Art und Weise eine Mobile App im jeweiligen Fall Sinn macht. Die Anzahl der Smartphone-Nutzer steigt permanent, die technischen Möglichkeiten der diversen mobilen Endgeräte wachsen, und es werden sich mit dem neuen Netzstandard LTE und den damit verbundenen höheren Durchsatzraten weitere Optionen im mobilen Internetumfeld ergeben. Apps agieren hierbei als Technologietreiber und gestalten die Evolution von Endgeräten und deren Leistungsfähigkeit bedeutend mit. Ständig wachsende Anforderungen an die Endgeräte (Prozessorleistung, Akkulaufzeit, Displaygröße, Bedienbarkeit etc.) bringen Hersteller wie schon im Bereich der Computerspiele dazu, stets neue Geräte zu entwickeln. Speziell die vier Kernfaktoren, die in Abschn. 33.2 beschrieben wurden sind hier als Katalysatoren hervorzuheben. So kann in Zukunft die Entwicklung einer Mobile App nochmals einfacher und schneller realisierbar sein, bzw. der Kunde kann bestehende Apps selbstständig anpassen und individualisieren. Aber auch eine Kombination oder Zusammenführung von diversen Apps könnte dem Kunden einen weiteren Mehrwert bringen (auch als „App in App“).

Es ist nur eine Frage der Zeit, bis wir all diese Funktionen der mobilen Geräte auch in Alltagsgegenständen wiederfinden werden und somit die Informationslücke zwischen der realen und virtuellen Welt minimieren werden. Man wird sehen, welche Gegenstände sich demnächst alle mit dem Internet verbinden lassen, sobald die Größe und Beschaffenheit der Chips, Energiezellen, Funkmodule und damit die Anbindung an das Internet dies zulässt. So rückt heute schon „intelligente Kleidung“ in den Raum des Möglichen, indem Technik bspw. in die Nähte von T-Shirts integriert

oder in aufgesetzte Taschen von Designer-Jeans eingenäht wird. Durch Nutzung der Optionen aus dem „Internet der Dinge“ können wiederum Prozesse werden (Verkehrsfluss, Stromverteilung etc.) effizienter gestaltet oder der Kundenservice verbessert werden. Bei all den Anwendungen muss jedoch die User Experience und damit der „Joy of Use“ für den Nutzer stets im Vordergrund stehen, damit die jeweilige App, wo auch immer sie abläuft, genutzt wird.

## Literaturverzeichnis

1. FTD (2010) Welchen Einfluss Smartphones haben. <http://www.ftd.de/karriere-management/management/:mediennutzung-in-deutschland-welchen-einfluss-smartphones-haben/50145166.html> Zugriffen 28. Mai 2011
2. Mobilemarketingwatch (2011) WWDC: iOS 5 to Feature Twitter Integration for Advanced Smartphone-Based Social Networking. <http://www.mobilemarketingwatch.com/wwdc-ios-5-to-feature-twitter-integration-for-advanced-smartphone-based-social-networking-15994/> Zugriffen 10. Juni 2011
3. Facebook (2011) Statistik. <http://www.facebook.com/press/info.php?statistics> Zugriffen 28. Mai 2011
4. ZDNet (2011) Geld verdienen mit Mobile Apps: Realität oder Mythos? <http://www.zdnet.de/magazin/41554175/geld-verdienen-mit-mobile-apps-realitaet-oder-mythos.htm> Zugriffen 11. Juni 2011
5. Internetworld (2011) Strategien für In-App-Sales – Mobile Geschäftsmodelle mit Zukunft. <http://www.internetworld.de/Nachrichten/Mobile/Praxistipps/Strategien-fuer-In-App-Sales-Mobile-Geschaeftsmodelle-mit-Zukunft-53369.html> Zugriffen 08. Juni 2011
6. Spiegel Online (2011) Apple lockert Regelungen für Verlage. <http://www.spiegel.de/netzwelt/gadgets/0,1518,767953,00.html> Zugriffen 12. Juni 2011
7. Osterwalder A, Yves P (2009) Business Model Generation. [http://www.businessmodelgeneration.com/downloads/businessmodelgeneration\\_preview.pdf](http://www.businessmodelgeneration.com/downloads/businessmodelgeneration_preview.pdf) Zugriffen 30. Mai 2011

**Teil X**

## **Marktwirtschaftliche Sicht: Wandel in der Gesellschaft**

---

# Kapitel 34

## Mobile Kommunikation der nächsten Dekade

### Erkenntnisse aus den Zukunftsstudien des Münchener Kreises

Arnold Picot und Christoph Janello

**Zusammenfassung** Dieser Beitrag gibt, über die enge Betrachtung von Apps und deren Technologien hinausgehend, einen Ausblick auf die bevorstehenden, teilweise auch schon stattfindenden Veränderungen auf dem Gebiet der mobilen Telekommunikation. Er weist somit auf künftige Entwicklungen und Bedingungen hin, die den Rahmen für mobile Datenanwendungen und Apps bilden.

Eine hilfreiche Basis für dieses Vorhaben stellt die vom Münchener Kreis und Partnerunternehmen durchgeführte Delphi-Studie „Zukunft und Zukunftsfähigkeit der Informations- und Kommunikationstechnologien und der Medien“ dar, welche verschiedenen für Telekommunikationsmärkte relevante Veränderungen bis ins Jahr 2030 zu prognostizieren versucht.

So verändert die Verfügbarkeit von breitbandigem, mobilem Internetzugang – eine Voraussetzung, die zunächst realisiert sein muss – in Kombination mit leistungsfähigen Smartphones und entsprechenden Services das menschliche Kommunikations-, Mediennutzungs- und Arbeitsverhalten. Allerdings wird dadurch auch das Problem der „digitalen Spaltung“ eher größer als kleiner.

Aufgrund der über „Apps“ flexiblen Erweiterbarkeit des Funktionsumfangs der Smartphones entstehen darüber hinaus aber auch neue Anwendungen, die oftmals „lokale“ Funktionen des Endgerätes mit internetbasierten Services verbinden und somit neue Nutzenpotenziale erschließen. Diese technischen Innovationen werfen eine Reihe ökonomischer Fragestellungen (bspw. Marktplatzorganisation und Revenue-Sharing) auf und stellen traditionelle Geschäftsmodelle infrage.

---

Arnold Picot

Ludwig-Maximilians-Universität München, Institut für Information, Organisation und Management,  
E-mail: picot@lmu.de

Christoph Janello

Ludwig-Maximilians-Universität München, Institut für Information, Organisation und Management,  
E-mail: janello@lmu.de

Auf der Grundlage der umfangreichen prognostischen Erkenntnisse der Zukunftsstudien des Münchener Kreises werden o. g. Entwicklungen genauer erläutert und hinsichtlich Eintrittswahrscheinlichkeit und -zeitpunkt verortet. Darauf aufbauend werden deren Implikationen hergeleitet und aus ökonomischer wie gesellschaftlicher Sicht reflektiert.

### 34.1 Einleitung

In den vorangegangenen Kapiteln des vorliegenden Buches wurde das Themengebiet „Mobile Apps“ aus anwendungsorientierter wie technologischer Sicht umfassend betrachtet. Dieser Beitrag soll nun über die isolierte Betrachtung der Apps oder Technologien hinausgehen. Dazu geben wir einen Ausblick auf die bevorstehenden, teilweise auch schon stattfindenden Veränderungen auf dem Gebiet der mobilen Telekommunikation und zeigen somit einen Zukunftspfad für die in diesem Band beschriebenen Teilaufgaben auf.

Eine hilfreiche Basis für dieses Vorhaben stellt die vom Münchener Kreis und Partnerunternehmen durchgeführte Delphi-Studie „Zukunft und Zukunftsfähigkeit der Informations- und Kommunikationstechnologien und der Medien“ dar, welche verschiedenste für Telekommunikationsmärkte relevante Veränderungen bis ins Jahr 2030 zu prognostizieren versucht [1, 2]. Um die Grundlagen dieser Aussagen besser einschätzen zu können, werden Ansatz und Methodik im Folgenden kurz vorgestellt.

Delphi-Studien stellen eine Methode dar, Zukunftsprognosen für nur schlecht zu strukturierende Fragestellungen zu erstellen, die sich modellbasierten statistischen Methoden entziehen [3–5]. Dabei wird auf das Wissen einer heterogenen Gruppe von Experten zurückgegriffen, die sich in einem strukturierten, anonymisierten Kommunikationsprozess austauschen und so als Gruppe ein komplexes Problem bewerten können [3].

Konkret wurden in der zweiten Phase der Delphi-Studie (die erste Phase war als Vorstudie angelegt) in der ersten Welle 551 und in der zweiten Welle 439 Experten internetgestützt befragt. Die Fachleute vertraten die Domänen der Wirtschaft, Wissenschaft und Politik und stammten aus Deutschland, der EU, den USA und weiteren Ländern (insbesondere aus Asien). Abgefragt wurden 144 Thesen aus verschiedenen Themenfeldern zur Zukunft und Zukunftsfähigkeit der IKT und Medien sowie 61 weitere Fragen. In der dritten Phase wurde neben der eher quantitativen Befragung der Fachleute zwischen diesen eine moderierte Online-Expertendiskussion durchgeführt, wodurch die Erkenntnisse weiter vertieft werden konnten.

Dieser Beitrag enthält, dem inhaltlichen Schwerpunkt des vorliegenden Sammelbandes entsprechend, ausgewählte Erkenntnisse dieser Studie, die sich auf die Zukunft der Mobilkommunikation beziehen, und verknüpft sie mit ergänzenden und vertiefenden Analysen und Reflexionen. Dies geschieht auf der Basis stark verdichteter Kernaussagen. Die detaillierten Erkenntnisse sind, soweit nicht anders angegeben, der Studie des Münchener Kreises zu entnehmen [1, 2].

## 34.2 Veränderungstrends und Potenziale der mobilen Kommunikation

Für den Erfolg App-basierter Geschäftsmodelle, gerade im Bereich professioneller Anwendungen, stellt die flächendeckende Verfügbarkeit einer schnellen und zuverlässigen Internetzugangsinfrastruktur eine notwendige Grundvoraussetzung dar. Deren prognostizierte Entwicklung wird im folgenden Abschnitt näher beleuchtet.

### 34.2.1 Verfügbarkeit und Nutzung von mobilem und stationärem Breitbandinternetzugang

Die technische Verfügbarkeit von Internet stiftet nur Nutzen, wenn dieses von der Bevölkerung auch angenommen wird. Aus diesem Grunde wird zunächst die prognostizierte Situation der Nutzerbasis dargestellt, bevor auf die technische Verfügbarkeit des Internetzugangs eingegangen wird:

**Das Internet erreicht in den nächsten Jahrzehnten fast die gesamte Bevölkerung, aber die digitale Spaltung verschwindet womöglich nie.**

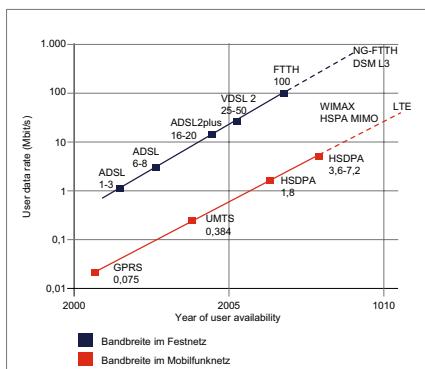
Die Mehrheit der Experten (66 %) prognostiziert, dass spätestens im Jahr 2024 95 % der deutschen Bevölkerung das Internet und seine Dienste aktiv und regelmäßig nutzen werden. Allerdings gehen immerhin 10 % der Befragten davon aus, dass der Wert von 95 % Internetnutzung in der Bevölkerung nie erreicht werden wird. Auch sind 44 % der Delphi-Experten der Auffassung, dass das Problem der digitalen Spaltung in Deutschland nie überwunden werden wird. Hinter dem Begriff der digitalen Spaltung verbergen sich vielschichtige Probleme: Zunächst wird technisch auf die Verfügbarkeit des breitbandigen Internets abgestellt, konkret ist die Unterversorgung in ländlichen Gebieten gemeint. Neben diesem mit entsprechendem finanziellem Einsatz technisch lösbarer Problem verbergen sich mit Nutzung vs. Nichtnutzung und ferner der medienkompetenten vs. inkompetenten Nutzung des Internets hinter dem Begriff der digitalen Spaltung gesellschaftliche Herausforderungen, die einer umfassenderen Herangehensweise bedürfen.

Die digitale Spaltung ist für Smart Mobile Apps folglich aus zwei Perspektiven von Bedeutung: Einerseits reduziert sich die adressierbare Zielgruppe für Apps, Endgeräte und Dienste auf die „glückliche“ Seite der Spaltung, zuerst in technischer und, die Zielgruppe weiter einschränkend, darüber hinaus auch in gesellschaftlicher Lesart. Andererseits werden Personen auf der „benachteiligten“ Seite von einer weiteren Entwicklung ausgeschlossen, wodurch die Spaltung eher zu- als abnimmt.

Hinsichtlich der technischen Verfügbarkeit des breitbandigen Internetzugangs wird im Folgenden entsprechend der Fokussierung auf mobile Nutzung besonders die mobile Verfügbarkeit in den Vordergrund gestellt und mit der Situation im Festnetz verglichen.

**Mobiles Breitband (> 50 MBit/s) ist in 8 bis 13 Jahren flächendeckend verfügbar.**

### **Historische Entwicklung der Bandbreiten im Festnetz und Mobilfunknetz**



### **Von den Delphi-Experten prognostizierte Bandbreitenentwicklung**

Jahr	Verfügbare Bandbreite (Mbit/s)	
	Stationär	Mobil
2010	8	3
2015	36	7
2020	101	20
2025	195	47
2030	406	84

(Mittelwerte aus den Antworten der Experten für Deutschland)

**Abb. 34.1** Historische und prognostizierte Bandbreitenentwicklung [1, 6]

Der prognostizierte Zeitraum von 8–13 Jahren für die flächendeckende Verfügbarkeit des mobilen Breitband-Internetzugangs erscheint zunächst als recht hoch gegeben, verdeutlicht jedoch die noch immer bestehenden Herausforderungen für den Ausbau des Mobilfunknetzes: die Erhöhung der Übertragungskapazitäten (hier: mindestens 50 MBit/s im Up- und Download) und der flächendeckende Netzausbau. In Bezug auf Deutschland gehen 34 % der Experten davon aus, dass dieses Ziel spätestens 2019 erreicht sein wird, 29 % erwarten dies erst ein bis fünf Jahre später.

Aber auch über diesen Zeitraum hinaus kann der drahtlose Internetzugang das Festnetz nicht vollständig substituieren:

**Der Festnetzzugang weist auch in Zukunft eine wesentlich höhere Bandbreite als das Mobilfunknetz auf.**

Sowohl der historische Vergleich als auch die Prognose der Delphi-Experten zeigt das deutliche Kapazitätsdefizit des mobilen gegenüber dem stationären Internetzugang (vgl. Abb. 34.1).

Auch die zusätzlich verfügbaren Frequenzen aus der digitalen Dividende können diesen Abstand allenfalls verringern, aber nicht auflösen, da die Luftschnittstelle physikalische Limitationen mit sich bringt, die das Festnetz nicht aufweist: Verwendet ein Anwender den mobilen Internetzugang, so verringert sich die verbleibende Kapazität für andere Nutzer in dieser Zelle. Zudem gibt es im verfügbaren Frequenzspektrum physikalische Grenzen und es besteht ein Tradeoff zwischen Zellgröße und Übertragungskapazität. Ferner ist die Zuverlässigkeit geringer als im Festnetz und es bestehen Akzeptanzprobleme durch die Angst vor gesundheitsschädlicher Strahlung [7].

Folglich ist bei der Konzeption und Implementierung von datenintensiven Apps auf ein hybrides Internetzugangskonzept abzustellen, das bei Prozessen mit hohem Bandbreitenbedarf bevorzugt auf festnetzbasierte Infrastrukturen (bspw. auch WLAN) zurückgreift.

### **34.2.2 Trends in Technologie und Nutzung mobiler Endgeräte**

Auf Seite der Endgeräte ist ein klarer Trend in Richtung Leistungssteigerung und Mobilität zu erkennen – so werden bereits 2011 mehr Smartphones und Tablet-PCs verkauft als Notebooks und Personal Computer. Bei Smartphones stellt die Sprachtelefonie, ursprünglich das einzige Nutzingsfeld, nur noch eine Applikation unter vielen dar.

**Telefonie und SMS machen in den USA nur noch ein Drittel der Nutzungszeit von Smartphones aus.**

So kommt die Telefonie in den USA nur noch auf 32 % der Nutzungszeit von Smartphones, während 12 % auf die Nutzung des Internets, 9 % auf E-Mails und 47 % auf andere Apps entfallen [8]. Smartphones können also mit Fug und Recht als mobile Taschencomputer bezeichnet werden und es ist davon auszugehen, dass sich dieser Trend erhält und voraussichtlich weiter verstärkt.

Diese Tendenz zu kleinen, multifunktionalen Endgeräten hat auch Auswirkungen auf die Nutzung traditioneller Medien:

**Klassische Medien werden in Zukunft auf multimedialen, mobilen Endgeräten rezipiert.**

Der Wandel im dominanten Mediennutzungsverhalten braucht jedoch Zeit. Erst im Zeitraum von 2020 bis 2024 wird nach mehrheitlicher Einschätzung der Experten von mindestens 75 % der Bevölkerung ein multimediales, mobiles Endgerät als verbindendes Element zur Rezeption der klassischen Medien (Buch, Zeitung, Zeitschrift, Fernsehen und Internet) genutzt.

Hinsichtlich der vollständigen Umstellung von Printmedien auf elektronische Erscheinungsformen sind die Experten eher skeptisch: Bei Zeitungen sind 54 % der Experten aus Deutschland der Meinung, dass dies nie eintreten wird, bei Zeitschriften sogar 63 %. Etwas optimistischer sind die Prognosen für digitale Bücher: „Nur“ 38 % der Befragten sind der Meinung, dass das E-Book sich nie als das Standardformat des Buches etablieren wird. Hinsichtlich des wahrscheinlichsten Eintrittszeitraums ist jedoch kein Konsens zu erkennen. Dies ist angesichts der großen inhaltlichen Bandbreite von Büchern und der damit einhergehenden heterogenen Nutzungsformen und -situationen wenig überraschend: Die Digitalisierung ist in manchen Bereichen wie etwa den Nachschlagewerken bereits weitgehend abgeschlossen, während bspw. Kinderbücher und hochwertige Kunstbücher noch einige Zeit primär in gedruckter Form erscheinen werden [9].

Klassische Medien behalten nach Meinung von 51 % der Experten aus Deutschland ihre Rolle als Leitmedien auf Dauer. Noch deutlicher fällt die Meinung in Bezug auf den öffentlich-rechtlichen Rundfunk aus: 72 % der Befragten sind der Auffassung, dass dieser für die öffentliche, demokratische Meinungsbildung auch langfristig erforderlich sein wird.

Auf dem Gebiet der Applikationen bringt das Zusammenspiel aus mobilem – daher für den Nutzer immer verfügbaren und leistungskraftig ausgestatteten – Endgerät

und internetbasierten Diensten neue Potenziale, die in den kommenden Jahren und Jahrzehnten ausgeschöpft werden.

**Location-based Services werden täglich genutzt.**

Als treffendes Beispiel für solche Dienste sind Location-based Services zu nennen, die als Applikation auf dem Endgerät in Kombination mit der genau ermittelten Position und Internetdiensten den Nutzer situations- und bedarfsgerechte Services anbieten können. 35 % der Delphi-Experten aus Deutschland schätzen das Potenzial dieser Technologie so hoch ein, dass aus ihrer Sicht bereits 2015 bis 2019 75 % der Mobilfunknutzer täglich Location-based Services über ihr mobiles Endgerät verwenden. Weitere 24 % der Befragten prognostizieren dies für den Zeitraum zwischen 2020 und 2024.

**Die Nutzung von sozialen Netzwerken (z. B. Facebook) wird zum Standard.**

Das „soziale Internet“ wird zum dominanten Ort der Kontaktpflege. Mehr als die Hälfte der deutschen Bevölkerung pflegt seine sozialen Kontakte regelmäßig über Anwendungen und Dienstleistungen des „Social web“ (Web 2.0) – dies prognostizieren 30 % der befragten Fachleute für den Zeitraum von 2015 bis 2019 und 26 % für die fünf darauf folgenden Jahre. Bereits heute verfügen 39 % der deutschen Onlinebevölkerung über ein Profil in einem sozialen Netzwerk zur privaten Kontaktpflege [10]. Dass soziale Netzwerke durch die Nutzung auf mobilen Endgeräten, etwa durch die Integration von Location-based Services, neue Nutzenpotenziale erschließen und somit weiter an Attraktivität gewinnen werden, steht außer Frage. Dieser Aspekt konnte jedoch in der Delphi-Studie nicht vertieft werden.

Im Folgenden werden mit Mobile Commerce und Cloud Computing zwei weitere Themenfelder adressiert, denen in Verbindung mit Apps ein hohes Potenzial attestiert wird.

Während der derzeitige Anteil der im Mobile Commerce erzielten Umsätze im Vergleich zum gesamten Marktvolumen noch vernachlässigbar erscheint, so ist doch ein umso größeres Wachstum dieses Segmentes zu vermuten. In der Zukunftsstudie wurde mit der Standardisierung der Bezahlverfahren ein Aspekt des M-Commerce abgedeckt, der für die Nutzerakzeptanz und in Folge für die Adoption dieser Anwendungen eine wesentliche Rolle spielt.

**Im Mobile Commerce kommt es zur Standardisierung der Bezahlverfahren.**

Das Expertengremium ist hinsichtlich der Etablierung eines Standards verhältnismäßig optimistisch: 62 % der Fachleute aus Deutschland rechnen damit, dass spätestens 2024 ein weltweiter Technologiestandard für die „elektronische Geldbörse“ etabliert ist. Near Field Communication (NFC), bei dem durch eine bewusst geringe Reichweite die physische Nähe des Endgeräts zum Bezahlterminal als Identifizierungsmerkmal eingesetzt wird, könnte hierfür eine technologische Basis darstellen. Auch hier ist die ausreichende Verfügbarkeit von Endgeräten und Diensten als Erfolgsdeterminante anzusehen. Die Tatsache, dass bereits im Jahre 2011 in einzelnen Ländern verschiedene mobile Bezahllösungen eingeführt werden, könnte darauf hinweisen, dass diese Entwicklung deutlich schneller verläuft.

**These:** Mehr als 75 Prozent der Daten (z.B. Dokumente, Bilder, Musik, Firmendatenbanken) liegen im Internet (Net centric approach).

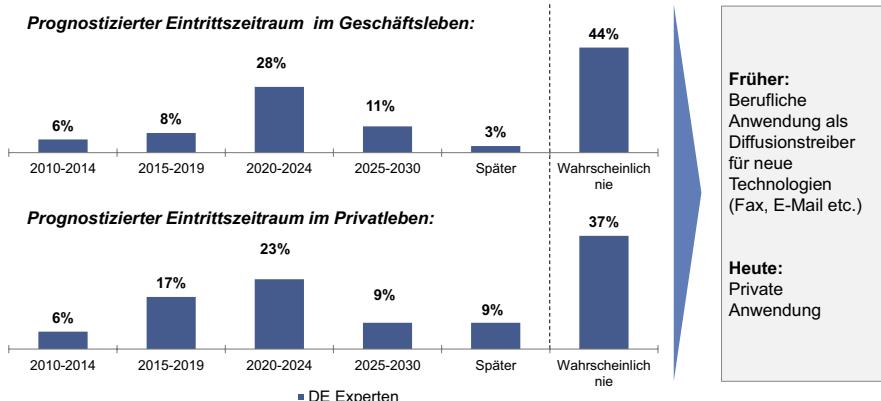


Abb. 34.2 Nutzung von internetbasierter Datenhaltung im beruflichen und privaten Kontext [1]

Die Interaktion zwischen mobilen Endgeräten und über das Internet angebundnen serverseitigen Diensten führt in der vollen Ausprägung zum Cloud Computing. Hierbei spielt die zentrale serverseitige Datenhaltung mit universeller Zugreifbarkeit über das Internet eine wichtige Rolle.

**Die zentrale Datenhaltung im Internet setzt sich nicht vollends durch – im privaten jedoch mehr als im beruflichen Kontext.**

Wie in der nachstehenden Abb. 34.2 ersichtlich, sind die Delphi-Experten hinsichtlich der Akzeptanz der zentralen Datenhaltung im Internet eher skeptisch. Interessant ist allerdings, dass der privaten Anwendung höhere Chancen eingeräumt werden als der beruflichen. Die private Nutzung löst demnach die berufliche als Innovationstreiber zusehends ab – allerdings unterscheiden sich selbstverständlich auch die Sicherheitsanforderungen in den beiden Bereichen.

#### Software as a Service wird zur Normalität.

Als wesentlich aussichtsreicher beurteilen die Studienteilnehmer die Perspektive von über das Web bedarfsweise genutzter, serverseitig ausgeführter Software (Software as a Service). Den Zeitpunkt für die vordringliche Präsenz dieses Nutzungsszenarios in Deutschland sehen 9 % der Befragten im Zeitraum zwischen 2010 und 2014, 36 % zwischen 2015 und 2019 und 27 % zwischen 2020 und 2024.

Ferner können durch den Einsatz mobiler Systeme für Unternehmensanwendungen deren Wertschöpfungsprozesse, gar die Unternehmensstruktur an sich, mobilisiert werden, wodurch deren Effizienz gesteigert werden kann [11]. Beispiele hierfür finden sich u. a. im Simobit-Programm des BMWi ([www.simobit.de](http://www.simobit.de)), das zwölf entsprechende Forschungs- und Entwicklungsprojekte fördert. Es ist zu erwarten, dass die Mobilisierung von Teilen von oder gar ganzen Wertschöpfungsketten in

den nächsten Jahren erheblich zunimmt und sich insofern eine Vielzahl von App-Anwendungsmöglichkeiten im geschäftlichen Sektor eröffnen wird.

### 34.3 Ökonomische Implikationen

Die im vorliegenden Band und in diesem Beitrag skizzierten, mit Veränderungen im Nutzungsverhalten einhergehenden technologischen Entwicklungen ziehen auch Anpassungen der ökonomischen Konzepte und Strukturen nach sich.

Herkömmliche Mobiltelefone stellen die Basis für zwei Geschäftsmodelle dar: Hardwareverkauf und die Bereitstellung von Netzzugang für Datendienste und Sprachtelefonie. Dabei blieb der Funktionsumfang des Endgerätes im Wesentlichen über dessen Nutzungsdauer unverändert. Die Öffnung der Smartphonebetriebssysteme für nachträgliche Erweiterungen durch Apps verändert diese beiden Sachverhalte: Durch die Installation von Apps lassen sich durch den Nutzer seine jeweils benötigten Funktionen nachrüsten. Es entsteht eine „Spielwiese“ für neue Geschäftsmodelle, die durch die Endgerätehersteller, Mobilfunknetzbetreiber, Dritte oder eine beliebige Kombination aus diesen realisiert werden können. Erlöse werden sowohl direkt (einmalig oder periodisch bezahlte Apps) als auch indirekt über Werbung erzielt, der Erlösgenerierungszeitraum eines verkauften Endgerätes wird somit gegenüber der einmaligen Zahlung erweitert. Immerhin ist derzeit etwa jede vierte installierte App kostenpflichtig (siehe Abb. 34.3).

In der Abbildung zu Grunde liegenden Statistik werden Mobile Business Apps (noch) nicht als eigenständige Kategorie aufgeführt. Dies ist aber nicht als Indiz für deren geringe Bedeutung zu werten, vielmehr werden die Potenziale im unternehmerischen Kontext erst jetzt erschlossen.

Die Zukunft wird zeigen, welche Varianten der Offenheit im Plattformwettbewerb (geschlossene vs. offene Plattform) und der Erlösaufteilung (Marktplatzbe-

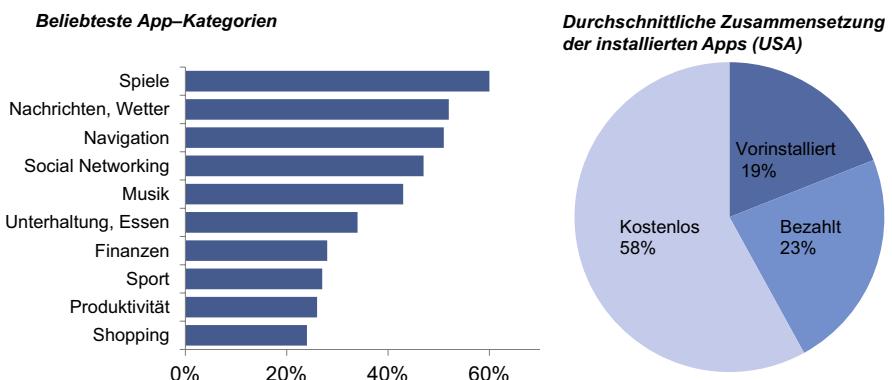


Abb. 34.3 Zusammensetzung derzeit installierter Apps [8, 12]

treiber vs. App-Entwickler) den größten Erfolg nach sich ziehen. Ziel muss es sein, den Nutzen und somit die Zahlungsbereitschaft des Kunden für sein individuell konfiguriertes Produkt zu maximieren.

## 34.4 Reflexionen

Abschließend werden nachstehend die im vorliegenden Beitrag behandelten Themenbereiche zusammengefasst und reflektiert.

Kein Zweifel besteht daran, dass die Nutzung mobiler Kommunikation weiter zunehmen wird. Dabei kommt es durch Voice Over IP zu einer Konvergenz von Sprach- und Datendiensten. Vor diesem Hintergrund erscheint auch die derzeit dominante Abrechnung von Mobilfunkleistungen im Sprachbereich nach Gesprächsdauer fraglich. Folglich werden Volumenpakete und Flatrates bzw. Kombinationen aus beiden weiter zunehmen.

Um mit der voranschreitenden technischen Entwicklung auf der Endgeräteseite (Software und Hardware), aber auch mit der zunehmenden Mobilität der Nutzer Schritt halten zu können, muss die Infrastruktur für Mobilkommunikation weiter ausgebaut werden. Aufgrund der gegebenen physikalischen Restriktionen der Luftschnittstelle wird diese aber nie vollends in der Lage sein, das Festnetz zu ersetzen, sondern dieses weiterhin in sinnvoller Weise ergänzen.

Die Mobilkommunikation entwickelt sich immer mehr zum Experimentierfeld für neue technologische wie ökonomische Konzepte. Dies wird am Beispiel der Apps und der damit verbundenen Plattformen, Marktplätze und Erlösmodelle deutlich. Der Einsatz von Apps innerhalb der Anwendungssysteme eines Unternehmens verspricht großes Potenzial, das es zu heben gilt. Zugleich sind die spezifischen Anwendungsvorteile von mobilen Apps im geschäftlichen Bereich (etwa im Bereich von Vertriebs-, Service und anderen mobilen Fachtätigkeiten) näher zu ergründen und mit anderen geschäftlichen IT-Anwendungen zu vergleichen. Auch die Übertragung der in diesem dynamischen Umfeld gemachten ökonomischen Erfahrungen in andere Bereiche erscheint vielversprechend.

Darüber hinaus schaffen moderne Kommunikationstechnologien, insbesondere in Verbindung mit der orts- und zeitunabhängigen Zugriffsmöglichkeit auf öffentliche wie eigene Informationen im Netz eine nahezu vollständige Informationstransparenz. Dies hat zur Folge, dass traditionelle Informationsvorsprünge ihren ökonomischen Wert verlieren. Zusehends bedeutsamer wird im Gegenzug der Wissensvorsprung, der aus der Verknüpfung und kognitiven Verarbeitung von Informationen hervorgeht. Wissen kann in der Generierung durch moderne Technologien unterstützt, letztlich aber nur durch die menschliche kognitive Verknüpfung und Verarbeitung gesteigert und angewandt werden. Dazu bedarf es breit angelegter, umfassender allgemeiner und professioneller Bildung, deren strategische Bedeutung für Wirtschaft und Gesellschaft weiter zunimmt.

## Literaturverzeichnis

1. Münchener Kreis et al. (2009) Zukunft und Zukunftsähigkeit der Informations- und Kommunikationstechnologien und der Medien: Internationale Delphi-Studie 2030. [http://www.muenchener-kreis.de/pdfs/Delphi/Zukunft\\_und\\_Zukunftsfaehigkeit\\_der\\_IKT\\_2009.pdf](http://www.muenchener-kreis.de/pdfs/Delphi/Zukunft_und_Zukunftsfaehigkeit_der_IKT_2009.pdf). Zugriffen am 10.5.2011
2. Münchener Kreis et al. (2010) Offen für die Zukunft - Offen in die Zukunft: Kompetenz, Sicherheit und neue Geschäftsfelder, Ergebniss der dritten Phase der internationalen Zukunftsstudie. [http://www.muenchener-kreis.de/pdfs/Delphi/Offen\\_fuer\\_die\\_Zukunft\\_Offen\\_in\\_die\\_Zukunft.pdf](http://www.muenchener-kreis.de/pdfs/Delphi/Offen_fuer_die_Zukunft_Offen_in_die_Zukunft.pdf). Zugriffen am 10.5.2011
3. Linstone HA, Turoff M (1975) The Delphi Method. Addison-Wesley, Reading, Mass.
4. Rowe G, Wright G (1999) The Delphi technique as a forecasting tool: issues and analysis. Int J Forecasting 15(4):353–375
5. Rowe G, Wright G (2001) Expert opinions in forecasting: the role of the delphi technique. Principles of forecasting. SJ Armstrong. Kluwer, Boston, pp 125–143
6. BMWI (2008) Breitband der Zukunft – Strategiepapier Breitband der Zukunft für Deutschland: Gemeinsame Strategie von BMWi und IKT-Wirtschaft zur Zukunft der Breitbandnetze. <http://www.zukunft-breitband.de/BBA/Redaktion/PDF/Publikationen/it-gipfel-strategiepapier-breitband-der-zukunft.property=pdf,bereich=bba,sprache=de,rwb=true.pdf>. Zugriffen am 10.6.2011
7. Grove N, Picot A, Jondral F J, Elsner J (2011) Why the Digital Dividend will not close the Digital Divide. InterMedia 39(2):32–37
8. Appsfire (2011) iOS Apps vs. Web apps. [http://appsfire.com/infographics/apps\\_vs\\_webapps\\_150dpi.png](http://appsfire.com/infographics/apps_vs_webapps_150dpi.png). Zugriffen am 15.5.2011
9. Janello C (2010) Wertschöpfung im digitalisierten Buchmarkt. Gabler, Wiesbaden
10. Busemann K, Gscheidle C (2010) Web 2.0: Nutzung steigt – Interesse an aktiver Teilhaben sinkt: Ergebnisse der ARD/ZDF-Onlinestudie 2010. Media Perspektiven 14(7–8):359–368
11. Picot A, Schmid M S (2009) Mobilisierung von Wertschöpfungsprozessen durch innovative und sichere Informationstechnologie, Studie im Rahmen der Begleitforschung des SimoBIT-Projektes. [http://www.simobit.de/de/intern-pdf/Picot\\_Schmid\\_2009\\_Mobilisierung\\_von\\_Wertschoepfungsprozessen.pdf](http://www.simobit.de/de/intern-pdf/Picot_Schmid_2009_Mobilisierung_von_Wertschoepfungsprozessen.pdf). Zugriffen am 10.5.2011
12. Purcell K, Entner R, Henderson N (2010) The Rise of Apps Culture. <http://pewinternet.org/Reports/2010/The-Rise-of-Apps-Culture.aspx>. Zugriffen am 14.6.2011

---

# **Kapitel 35**

## **Mobil – interaktiv – sozial: Der digitale Mensch von morgen – „always in touch“?**

### **Insights der Marktforschung zum digitalen Konsumenten und Nutzer**

Malthe Wolf, Kerstin Dirtheuer, Stefanie Sagl und Gabriele Herrmann

**Zusammenfassung** Der mobile digitale Mensch von morgen wird sich nicht mehr bewusst mit dem Internet verbinden, sondern er wird wie selbstverständlich immer und überall im Internet sein. Situationen und Nutzungskontexte werden die Art und Weise der jeweiligen Nutzung bestimmen und das heutige „always connected“ in ein „always in touch“ wandeln. Soziale Netze, Ortungsdienste, Location-based-Services oder Mobile Commerce sind dabei nur einige Beispiele dieser Entwicklung. Innovationen in diesem dynamischen Umfeld werden zukünftig nicht mehr alleine darauf beruhen, bestehende Dienste und Services der stationären Welt in die mobile Welt zu übersetzen. Vielmehr werden neue Industrien entstehen, die originär mobile Anwendungen und Dienste entwickeln. Gleichzeitig werden bei den Nutzern neue Motive und Bedürfnisse geweckt und in Kombination mit vollkommen neuen Nutzungssituationen zu einem bis dato noch nicht bekannten Nutzungsverhalten führen. Kommunikations- und Informationsverhalten, Unterhaltung und Infotainment, E-Commerce und Soziale Medien werden durch parallele, teilweise disruptive, häufig konvergente Entwicklungen in völlig neue Zusammenhänge gebracht, die von Wirtschaft und Gesellschaft verarbeitet und entsprechend umgesetzt werden müssen.

---

Malthe Wolf  
TNS Infratest GmbH, München,  
E-mail: malthe.wolf@tns-infratest.com

Kerstin Dirtheuer  
TNS Infratest GmbH, München,  
E-mail: kerstin.dirtheuer@tns-infratest.com

Stefanie Sagl  
TNS Infratest GmbH, München,  
E-mail: stefanie.sagl@tns-infratest.com

Gabriele Herrmann  
TNS Infratest GmbH, Hamburg,  
E-mail: gabriele.herrmann@tns-infratest.com

## 35.1 Einleitung

Spätestens im Jahr 2019 werden das Internet und seine Dienste überwiegend mobil genutzt, so eines der zentralen Ergebnisse der Delphi-Studie „Zukunft und Zukunftsfähigkeit der Informations- und Kommunikationstechnologien und Medien“ aus dem Jahr 2009, die der Münchener Kreis zusammen mit namhaften Partnern aus Wirtschaft und Wissenschaft erarbeitet hat. Diese Entwicklung, so die Studie, deren Basis eine Befragung von 551 Entscheidern aus der Informations- und Kommunikationstechnologie (IKT) und Medien-Branche ist, wird in den kommenden Jahren eine entscheidende Rolle bei der Digitalisierung von Gesellschaft und Wirtschaft darstellen. Und, laut den Experten, zu einer erheblichen Dynamik des digitalen Lebens führen (vgl. [1]).

Soziale Netze, Ortungsdienste, Location-based-Services oder Mobile Commerce sind dabei nur einige Beispiele dieser Entwicklung. Innovationen in diesem dynamischen Umfeld werden zukünftig nicht mehr alleine darauf beruhen, bestehende Dienste und Services der stationären Welt in die mobile Welt zu übersetzen. Vielmehr werden neue Industrien entstehen, die originär mobile Anwendungen und Dienste entwickeln. Gleichzeitig werden bei den Nutzern neue Motive und Bedürfnisse geweckt und in Kombination mit vollkommen neuen Nutzungssituationen zu einem bis dato noch nicht bekannten Nutzungsverhalten führen. Kommunikations- und Informationsverhalten, Unterhaltung und Infotainment, E-Commerce und Soziale Medien werden durch parallele, teilweise disruptive, häufig konvergente Entwicklungen in völlig neue Zusammenhänge gebracht, die von Wirtschaft und Gesellschaft verarbeitet und entsprechend umgesetzt werden müssen.

Schon heute ist diese Veränderung der Mobilkommunikation, die durch mobile Breitbandtechnologie ermöglicht wird, ersichtlich: Märkte verändern sich sichtbar, neue Bedürfnisse werden geweckt und neues Nutzungsverhalten zeichnet sich messbar ab. Dies führt zu Anpassungen der bestehenden Ecosysteme mit der Folge, dass sich für viele Bereiche, wie z. B. Medien, Mobilität oder auch Gesundheit gänzlich neue Geschäftsfelder auftun können, wenn die jetzt eröffneten Chancen genutzt werden.

## 35.2 Neue Marktforschungsmethoden liefern neue Erkenntnisse und sind Leitlinie strategischer Marktentscheidungen

Was bedeutet die zunehmende mobile Nutzung von Smartphone und Tablet-Computer für Unternehmen? Es bedeutet, dass Kunden immer stärker virtuell und online leben und das Medium (mobiles) Internet immer und überall nutzen.

Typische Nutzungsmuster sind heutzutage situativ ausgelöst, spontan, kurzfristig, und für den Nutzer und auch den Anbieter ist es zunehmend ein Multi-Channel-Erlebnis über mehrere Kanäle und Devices hinweg. Grundsätzlich bedeutet dies eine fundamentale Veränderung, denn dem Nutzer ist die eigene Nutzung selbst immer weniger bewusst. Ein Informationsbedürfnis wird meist durch einen Impuls

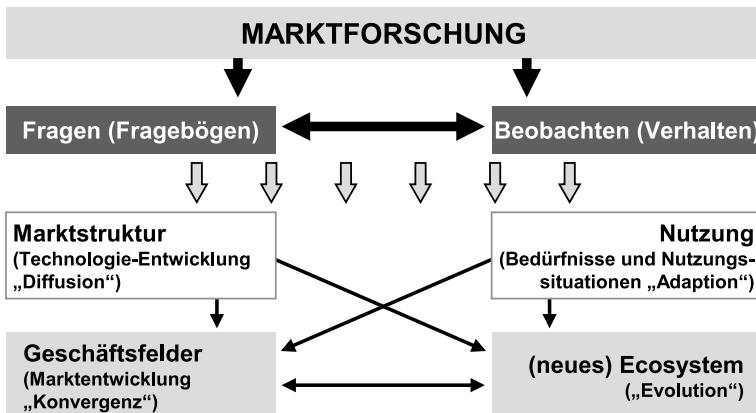
ausgelöst und das Nutzungserlebnis selbst ist oft kurz und sprunghaft. Des Weiteren bedeutet es, dass die Nutzung fragmentiert ist: Viele verschiedene Handy-Applikationen und Internetseiten bieten den Kunden die Möglichkeit, ihren individuellen und situativen Bedürfnissen jederzeit, an jedem Ort nachzugehen.

Gleichzeitig verschieben sich Geschäftsmodelle zunehmend in die digitale Welt und vernetzen Marken- und Produkterlebnisse. Für Unternehmen bedeutet dies, dass detailliertes und präzises Wissen über die Nutzung verschiedener Angebote zunehmend relevant wird. In der klassischen Marktforschung werden die Personen, die für Unternehmen relevant sind, rekrutiert und zu ihren Einstellungen, Nutzungsgewohnheiten, genutzten Angeboten und Motivationen für diese Nutzung befragt. Dies wird auch weiterhin wichtig und relevant bleiben, denn das Wissen um Gründe und Motivationen für die Präferenzbildung, z. B. für bestimmte Angebote, ist essentiell und für eine zielgerichtete Steuerung des Geschäftserfolges notwendig.

Allerdings können und sollten Teilbereiche aus der klassischen Befragung auf den Prüfstein gestellt werden. Mit zunehmend sprunghaft-vernetzter Nutzung verschwimmt die Erinnerung an den Nutzungsakt selbst. Befragte können sich zwar daran erinnern, dass sie ein bestimmtes Angebot im letzten Monat genutzt haben, wann, wie oft, wie lange, wo etc. ist jedoch nicht in der Erinnerung des Nutzers verankert. Genau hier verfügt die Marktforschung über zunehmend bessere technische Möglichkeiten und setzt diese auch ein. Kleine Add-ons, die die Aktivitäten detailliert und genau auf dem mobilen Gerät oder auch dem stationären Browser messen, können für diese Fälle gezielt eingesetzt werden. Damit ist die Marktforschung nun auch in der Lage, Dinge aus dem digitalen Leben abzubilden, die nicht erfragbar sind.

Gleichzeitig bietet sich damit die Möglichkeit, Nutzern kürzere Fragebögen zu geben, in denen sie nicht detailliert Auskunft über das eigene Nutzungsverhalten geben müssen und somit in die Lage versetzt werden, sich stärker auf den Punkt der Motivationen und Intentionen in der Befragung konzentrieren zu können. Aber auch die Beobachtung bzw. die Messung hat Grenzen. Sie zeigt zwar präzise und genau, wann, was, wo und wie lange genutzt wird – aber meistens eben nicht warum. Die Kunst besteht also darin, die verschiedenen Methoden von Messung und Befragung so miteinander zu verknüpfen, dass ein möglichst detailliertes und ganzheitliches Bild entsteht. Ebenso wichtig ist es zu wissen, was gemessen werden kann und soll und vor allem was durch eine Befragung erforscht werden soll (vgl. Abb. 35.1).

Die Entwicklung und der Erfolg von Tablet-Computern und Smartphones bedingen zunehmend diesen beschriebenen neuen Methodenmix. In der (nahen) Zukunft werden immer höhere Umsätze über mobile Endgeräte generiert werden. Daher ist das Bemühen um ein tiefgehendes, umfängliches Verständnis der Nutzer sowie ihrer Nutzungspräferenzen essentiell, um Angebote zu entwickeln, zu optimieren und einzuführen, die erfolgreich sind und auf die langfristig gebaut werden kann. Der moderne Konsument ist mannigfaltig vernetzt. Unternehmen werden den informierten und vernetzten Kunden akzeptieren lernen (müssen). Feedbacks in App-Stores oder Online-Kundenbewertungen werden eine zunehmend relevan-



**Abb. 35.1** Strategische Relevanz von Marktforschungsdaten. Quelle: Eigene Darstellung

te Entscheidungshilfe für den Konsumenten. Auch vor diesem Hintergrund ist es hochrelevant, den Kunden zu kennen, seine Vorlieben zu verstehen und in kreative und stimmige Geschäftsmodelle umzusetzen. Die derzeit verfügbaren Geräte werden in unterschiedlichen Nutzungssituationen vom Konsumenten genutzt. Das kleine Smartphone ist immer und überall dabei, hat aber einen kleineren Bildschirm. Der Tablet-Computer kann mit seinem hochwertigen Bildschirm komplexeren Content zeigen, ist aber nicht immer dabei. Erfolgreiche Geschäftsmodelle der Zukunft werden auf die jeweiligen Geräte zugeschnitten sein und dem Nutzer ein Multi-Channel-Produkterlebnis bieten, unter geschickter Einbeziehung der Geräte. Auch und ganz besonders vor diesem Hintergrund ist ein möglichst holistisches und ganzheitliches Verständnis der neuen Generation von Nutzern und der genutzten mobilen Endgeräte maßgeblich, um den Erfolg neuer Geschäftsideen positiv zu beeinflussen.

### 35.3 Einblicke in zentrale Studien zur Zukunft des Mobilfunkmarkts

Die Bedeutung der Informations- und Kommunikationstechnologien (IKT) und Medien für die Informationsgesellschaft des 21. Jahrhunderts ist größer als je zuvor. Um Einblicke in die langfristigen Entwicklungen dieser Branchen zu erhalten, hat der Münchener Kreis, gemeinsam mit den Herausgebern EIIC, Deutsche Telekom, TNS Infratest, Siemens, Vodafone, SAP, Telefónica O2, ZDF sowie den Förderern Focus, VDE, Sony, Deutsche Bank, Opera und Daimler sowie dem Bundesministerium für Wirtschaft und Technologie, die Studien „Zukunft und Zukunftsfähigkeit der deutschen IKT“, „Zukunft und Zukunftsfähigkeit der Informations- und Kommunikationstechnologien und Medien“ sowie „Offen für die Zukunft – Offen in die

Zukunft“ durchgeführt. Eine weitere Studie, „Mobile Life 2011“, gibt Einblicke zu Mobilfunknutzern weltweit.

In der Studie „Zukunft und Zukunftsfähigkeit der deutschen IKT“ wurde eine Standortbestimmung der deutschen Informations- und Kommunikationstechnologie vorgenommen. Zusätzlich wurden zentrale Handlungsfelder für die Entwicklung und den Einsatz der Informations- und Kommunikationstechnologien zur Sicherung der Wettbewerbsfähigkeit des Standorts und der gesellschaftlichen Zukunft Deutschlands abgeleitet.

Die Studie „Zukunft und Zukunftsfähigkeit der Informations- und Kommunikationstechnologien und Medien“ hat diesen Faden aufgegriffen und verfolgt das Ziel, Einblicke in die langfristigen Entwicklungen der IKT und Medien zu erhalten. Hierfür wurden Experten zu ihrer Einschätzung der Trends in diesen Branchen bis zum Jahr 2030 im Rahmen einer wissenschaftlich fundierten, international angelegten Delphi-Studie befragt.

Eine weitere Studie, „Mobile Life 2011“, ist eine der bislang umfangreichsten Studien zu Nutzern mobiler Endgeräte weltweit. Sie zeigt auf, wie Konsumenten mobile Technologien nutzen und mit ihrer Hilfe interagieren, und berücksichtigt dabei die internationale Perspektive. „Mobile Life“ vermittelt ein holistisches Bild der aktuellen Nutzer mobiler Endgeräte und veranschaulicht, welchen Einfluss Mobilfunk auf unsere zukünftige digitale Welt haben wird. Im Rahmen der Studie wurden mehr als 34.000 Mobilfunknutzer in über 43 Ländern befragt. Ziel war es, Informationen zur Nutzung von mobilen Anbietern und Services zu erhalten: Welche Dienste nutzt der Konsument wann und wie häufig, auf welcher Grundlage trifft er künftige Konsumententscheidungen und was müssen Anbieter tun, um auf diesem schnelllebigen Markt zukünftig zu bestehen. Für Unternehmen besteht hier die Chance, Konsumenten im eigenen Land oder Einzugsgebiet zu verstehen, sie aber auch in Relation zu Nutzern anderer Regionen zu setzen. Kulturelles Verständnis und damit das Verständnis, welche Geschäftsmodelle in der eigenen Region Erfolg haben werden und welche nicht, obwohl sie woanders gut funktionieren können, bietet eine wertvolle Entscheidungshilfe für Unternehmen.

Die Studie ermöglicht die Analyse aktueller Trends und Entwicklungen weltweit. Sie ist damit eine unerlässliche Basis für strategische Planungen über Länder hinweg und unterstützt bei wichtigen Entscheidungsprozessen.

Das Thema „Mobile“ ist sehr vielschichtig und komplex und bedarf einer Reihe moderner Forschungsmethoden. Die Firma Zokem hat für den deutschen Markt eine Pilotstudie durchgeführt, die gemeinsam mit TNS analysiert und vermarktet wird. In dieser wurden Nutzer von Smartphones mit einer kleinen App ausgestattet, die das Nutzungsverhalten und die Interaktion mit dem gesamten Gerät sekundengenau mitschreibt. Damit wird die Auswertung sehr detaillierter Informationen möglich. Nutzungsmuster können bis auf einzelne Apps oder Websites aufgezeigt werden: wo, wann, wie oft, wie lange wird welcher Content genutzt. Wertvolle Informationen, die durch klassische Befragungen nicht lieferbar sind.

## 35.4 Marktstruktur und Marktentwicklung im Mobilfunk

Die Geburtsstunde des Mobilfunkmarkts lässt sich mit der Einführung des Motorola DynaTac 8000X am 21. September 1983 datieren (vgl. [2]). Seither hat sich der Mobilfunkmarkt mit einer sehr hohen Dynamik weiterentwickelt.

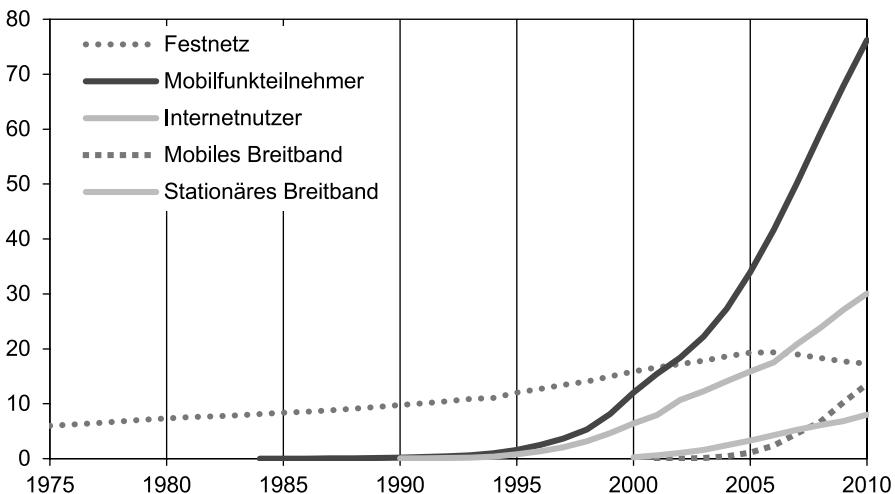
Schon seit Beginn der kommerziellen Mobilfunknutzung stehen die Endgeräte als treibende Kräfte der Nutzung und damit der Entwicklung im Mittelpunkt. Diese dynamische Entwicklung lässt sich anhand einiger Fakten des mobilen Endgeräts anschaulich darstellen: Als das Motorola DynaTac 8000X im Jahr 1983 seine FCC-Zulassung erhielt, wog es knapp ein Kilogramm und hatte eine Entwicklungszeit von mehr als zehn Jahren hinter sich gelassen (vgl. [2]). Außer Telefonie gab es keinen Nutzungskontext und bereits nach einer Stunde war die Akkuleistung des Geräts am Ende.

Diese aus heutiger Sicht extrem eingeschränkte Nutzung blieb zunächst einer sehr kleinen Nutzergruppe vorbehalten, nicht zuletzt aufgrund der Tatsache, dass ein solches Endgerät auf das heutige Preisniveau umgerechnet an die 9000 Euro kostete. Dennoch: Bereits im Jahr 1984 nutzten weltweit 300.000 Menschen diese neue Möglichkeit der Kommunikation.

Diese Zahlen machen deutlich, dass es schon sehr früh Anzeichen dafür gab, dass mobile Kommunikationstechnologien in den folgenden Jahren eine hohe Durchdringung in den modernen Industrienationen erreichen würden. Heute – 27 Jahre später – nutzen weltweit bereits 3,5 Milliarden Menschen weltweit Mobiltelefone (vgl. [3]). Allein in Deutschland werden nach Angaben der Bundesnetzagentur bereits ca. 109 Millionen Mobiltelefone genutzt, was einer Penetrationsrate von 133 Prozent in der Bevölkerung entspricht – der Trend geht zunehmend zum Zweit- oder gar Drittgerät (vgl. [4]).

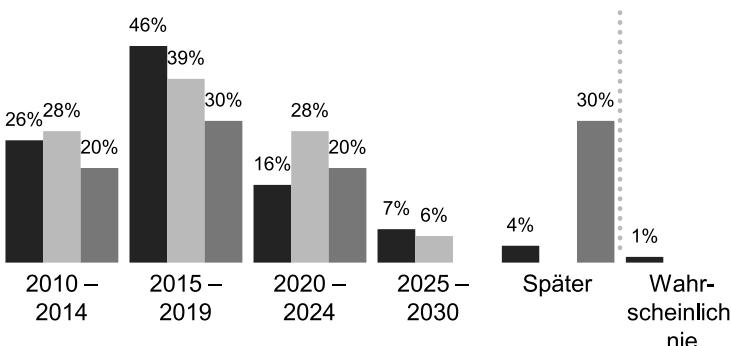
Besonders deutlich wird die außergewöhnliche Entwicklung der Mobilfunktechnologie, wenn man diese im Vergleich zur Entwicklung der klassischen Festnetztelefonie, des Internets, des stationären Breitbands und des mobilen Breitbands setzt (Abb. 35.2).

Der Vergleich der Diffusionskurven zeigt, dass die Zahlen der Mobilfunknutzer und der Internetnutzer ab Anfang der 1990er Jahre mehr oder weniger parallel ansteigen. Dabei ist die Dynamik der Entwicklung des Mobilfunks bis heute auf deutlich höherem Niveau als die des Internets: Ende 2010 nutzten weltweit über 75 Prozent der Menschen weltweit ein Mobilfunkgerät, während zeitgleich nur 30 Prozent der Weltbevölkerung Zugang zum Internet und seinen Diensten hatten. Neben der parallelen Entwicklung der beiden Technologien kann man heute die Fusion dieser beiden „Megatrends“ beobachten. Diese Entwicklung wird in den kommenden Jahren zu einer heute noch ungeahnten Explosion der Nutzungsmöglichkeiten führen. Erste Anzeichen sind am starken Anstieg der Nutzer mobiler Breitbandinternetverbindungen seit Anfang des Jahres 2001 zu erkennen. Diese Dynamik spiegelt auch die Befragung der internationalen Experten im Rahmen der Internationalen Delphi-Studie 2030 des Münchner Kreis wider (vgl. [1]): Wenn die Entwicklung wie erwartet fortschreitet, so die Ergebnisse, werden spätestens im Jahr 2019, aller Voraussicht jedoch schon deutlich früher, drei Viertel der deutschen Mobilfunknutzer das



**Abb. 35.2** Entwicklung zentraler Informations- und Kommunikationstechnologien von 1975 bis heute, weltweit, in Prozent. Quelle: International Telecommunication Union 2011

■ Deutschland-Experten ■ EU-Experten ■ USA-Experten



**Abb. 35.3** Zukünftige tägliche mobile Internetnutzung. These: 75 Prozent der Mobilfunknutzer in Deutschland nutzen über ihr mobiles Endgerät täglich das Internet, in Prozent. Quelle: Münchener Kreis et al. 2009; Basis: Experten für Deutschland ( $n = 85$ )

Internet über ihr mobiles Endgerät nutzen. Die überwiegende Mehrheit der befragten Experten für Deutschland geht davon aus, dass sich die mobile Internetnutzung dabei (sehr) positiv auf Wirtschaft und Gesellschaft auswirken wird. Treiber für diese Entwicklung sind nach Meinung der Befragten vor allem der technische Fortschritt, das Sinken der Kosten sowie Investitionen in die notwendige Infrastruktur.

Der dargestellte Trend resultiert aus einem Wechselspiel der Weiterentwicklung der Endgeräte auf der einen und der Nutzungsmöglichkeiten auf der anderen Seite. War das Nutzungsspektrum des Motorola DynaTAC 8000X nur auf reine Telefonie beschränkt, so ist die Vielfalt heutiger Smartphones, die über einen breitbandigen mobilen Internetanschluss und ein modernes Betriebssystem verfügen, mit allen

Anwendungen (Apps) nahezu unbegrenzt – und wir stehen hier erst am Anfang der Entwicklung.

Zukünftig wird das Internet selbst zunehmend durch mobile Trends geprägt. So kann man davon ausgehen, dass dies erneut zu gravierenden Umwälzungen der Informations- und Kommunikationstechnologien und ihrer Branche sowie der Medienbranche führen wird. Entscheidend hierfür ist, dass ein Großteil der noch bis vor wenigen Jahren vorwiegend stationären Internet- und Medienangebote die Schwelle zur digitalen mobilen Nutzung bereits überschritten haben oder diesen Schritt in naher Zukunft vollziehen werden. Die Nutzung des Internets verlagert sich – auch global betrachtet – immer mehr vom stationären Computer hin zum Surfen über ein mobiles Endgerät. Dabei wählen sich Nutzer über Smartphones oder Tablet-Computer nicht nur unterwegs ins Internet ein – auch zuhause sind sie immer beliebter, um schnell mal E-Mails zu checken oder etwas im Internet zu suchen (Abb. 35.3).

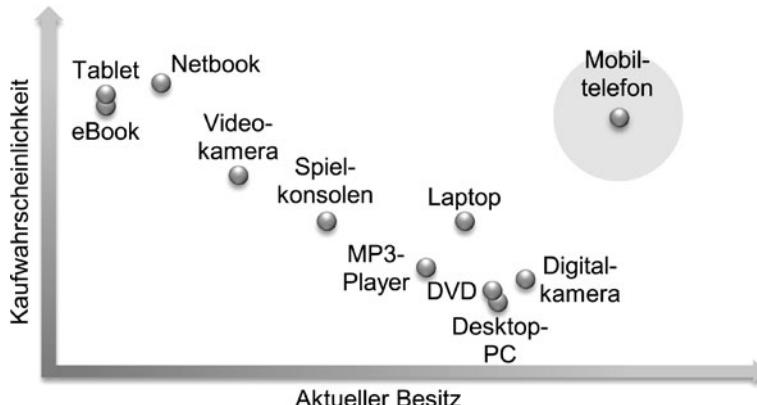
## 35.5 Der Einfluss mobiler Endgeräte auf unser Leben

Die zentrale Bedeutung des mobilen Endgeräts, heute und zukünftig, wird in der Studie „Mobile Life 2011“ deutlich: Die „mobile Revolution“ ist längst Realität – so das zentrale Ergebnis der Studie.

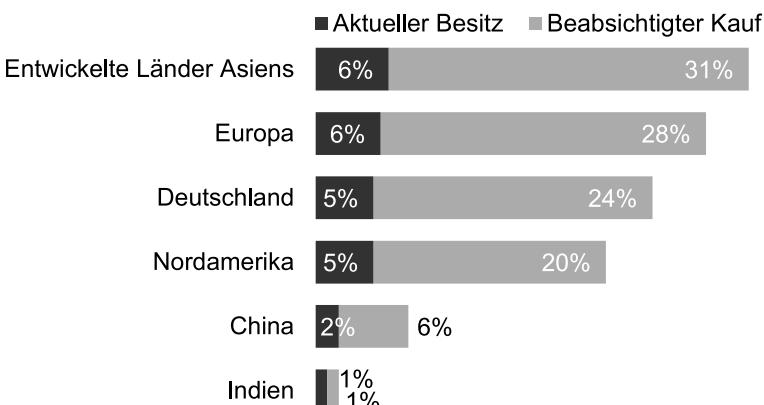
Mobiltelefone sind heute das meistverbreitete mobile elektronische Gerät in Deutschland. Sie werden häufig, obwohl noch funktionstüchtig, ersetzt. Neben den vielfältigen Nutzungsmöglichkeiten wie Telefonie, SMS, mobiles Internet, GPS und Apps wird die Entscheidung für diese Geräte oftmals aus Imagegründen getroffen, sie haben „Accessoire-Charakter“ (Abb. 35.4).

Ein immer größerer Teil der heute genutzten Handys sind Smartphones. Diese stellen derzeit die Spitze der heutigen Entwicklung auf dem Markt der Mobiltelefone dar und führen mit ihren großen, berührungsempfindlichen Bildschirmen, in Kombination mit breitbandigen Internetverbindungen, zu hohen Abverkäufen. Deloitte prognostiziert im Bericht „Technology, Media & Telecommunications Predictions 2011“ einen Abverkauf von 375 Millionen Smartphones weltweit für das Jahr 2011 (vgl. [5]). Mit den so genannten „Tablets“, einer Gerätekategorie, die Apple mit dem iPad im letzten Jahr erfolgreich etablierte, kommen weitere mobile Endgeräte auf den Markt. In Europa geben bereits 28 Prozent der Befragten an, sich in den nächsten sechs Monaten einen Tablet-Computer kaufen zu wollen (vgl. [6]), (Abb. 35.5).

Dabei geht von Tablets für Mobiltelefon- und Computer-Hersteller zumindest bisher weniger Gefahr aus als zunächst erwartet: Nutzer sehen das Tablet momentan eher als zusätzliches Gerät denn als Ersatz für Mobiltelefon oder Computer an. Im Zweifel werden Tablets von Mobilfunknutzern jedoch eher als Ersatz für den klassischen PC gesehen. Dies wird auch deutlich, wenn man die aktuellen Absatzprognosen für PCs betrachtet. Bislang ist für die Mehrheit der Nutzer der Tablet-Computer jedoch noch ein Zusatzgerät in Ergänzung zu den bereits vorhandenen mobilen Endgeräten (vgl. [6]), (Abb. 35.6).



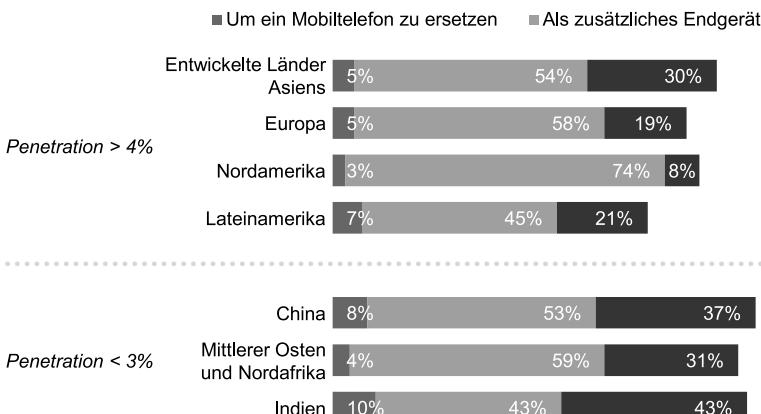
**Abb. 35.4** Mobiltelefone sind ein Must-have in Deutschland. Quelle: TNS 2011; Basis: Alle Befragten



**Abb. 35.5** Besitz und Kaufabsicht von Tablet-Computern. Quelle: TNS 2011; Basis: Alle Befragten

Mit der zunehmenden Verlagerung der Internetnutzung in das mobile Leben wird es deshalb für Unternehmen immer wichtiger, die mobile Zielgruppe zu verstehen, um sich erfolgreich in diesem Umfeld zu positionieren.

Bei näherer Betrachtung der Befragungsdaten von Mobiltelefonnutzern wird schnell deutlich, dass diese heute bereits mehr oder weniger die gesamte Bandbreite der Anwendungsmöglichkeiten des „klassischen“ Internets über ihre mobilen Endgeräte nutzen: Soziale Medien, Kontaktpflege, Navigieren, Recherchieren und die Nutzung „klassischer“ Medien über Internet – schon heute heißt dies für viele Nutzer: immer und überall. Einige Zahlen der Studie „Mobile Life 2011“ zum Thema Social Media sollen diese Entwicklung veranschaulichen: So nutzen im Jahr 2011 bereits 19 Prozent aller Mobiltelefonnutzer weltweit soziale Netzwerke mobil – in Deutschland sind es im gleichen Zeitraum nur 14 Prozent. Die Nutzung von Social-Media-Angeboten und das zunehmende Bedürfnis, mehrmals täglich seinen



**Abb. 35.6** Tablet-Computer schaffen neue Nutzungskontexte und -situationen. Quelle: TNS 2011; Basis: Tablet-Besitzer bzw. Befragte mit Kaufabsicht

Freunden mitzuteilen, wo man ist und was man macht, werden auch in Zukunft die Mobilkommunikation weiter vorantreiben. Mit zunehmender Verbreitung von internetfähigen Handys werden die Nutzerzahlen dieser Services rasch ansteigen.

„Gerade die bisher mobil noch wenig genutzten Social-Media-Services werden zukünftig an Bedeutung gewinnen: 38 Prozent der Netzwerker laden bereits Fotos oder Videos mobil in ihr soziales Netzwerk hoch. Weitere 21 Prozent planen dies künftig zu tun. Weltweit geben 24 Prozent der Befragten an, dass die Möglichkeit, Fotos und Videos aufzunehmen und mit anderen zu teilen, beim Kauf ihres nächsten Mobiltelefons eine wichtige Rolle spielen wird. Auch Streaming ist ein begehrter Service, den heute erst 17 Prozent dieser Zielgruppe nutzen, aber weitere 20 Prozent gern nutzen würden. Immer interessanter wird für die mobilen Netzwerker die Verknüpfung von Social-Network-Diensten mit ihren gerade stattfindenden „Offline-Aktivitäten“. Der Online-Check-In ermöglicht ihnen, Freunde und Bekannte auf ihr Verweilen an einem bestimmten Ort, z. B. im Café an der Ecke, aufmerksam zu machen. Freunde, die gerade in der Nähe sind, können so spontan auf ein Treffen vorbeikommen. Auch wenn dies bisher nur neun Prozent der Netzwerker nutzen, wünschen es sich derzeit bereits weitere 17 Prozent. Für Unternehmen ist diese Entwicklung sehr interessant; denn auf diese Weise können dem Verbraucher perfekt abgestimmte Services an ihrem aktuellen Aufenthaltsort angeboten werden“ [7].

Um mit dieser immer komplexeren Welt der mobilen Internet- und Mediennutzung auch methodisch Schritt halten zu können, spielen Messdaten in der Marktforschung eine immer wichtigere Größe. Nur so sind die sich verändernden oder neuen Nutzungssituationen auf der einen Seite sowie die Entwicklung der Nutzung auf der anderen Seite besser zu verstehen und in strategierelevante Entscheidungen überführbar. Erste Erkenntnisse aus der von der Firma Zokem durchgeföhrten Studie „Deutschland Pilot. Messung der Nutzung von Smartphones“ veranschaulichen dieses tiefere Verständnis durch Messdaten: So nutzen Smartphonebesitzer das Internet im Monat durchschnittlich 326 Minuten über ihren mobilen Browser, 254 Minuten entfielen auf die Nutzung von Apps und „nur“ 226 Minuten wurde telefoniert.

## 35.6 Trends im Mobilfunk – Neues Geschäftsfeld M-Commerce

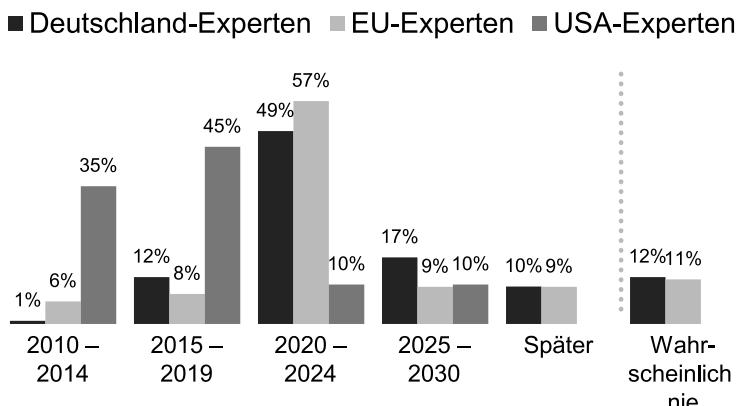
Neue Anwendungsfelder wie Mobile Commerce, Mobile Banking oder Cloud Computing sind nur einige Beispiele neuer Möglichkeiten, die mit der Entwicklung des Mobilfunks einhergehen: Eine Fülle von Innovationen wird durch das mobile breitbandige Internet gefördert. Herauszufinden, ob jede dieser Innovationen auch einen zukünftigen Trend abbildet, der seitens der Konsumenten und Nutzer ein spezifisches Bedürfnis adressiert, ist eine der zentralen Aufgaben der Markt- und Konsumentenforschung.

Beispielhaft wird im Folgenden an der Entwicklung des Mobile Commerce dargestellt, wie sich ein solcher, gerade im Entstehen befindlicher Trend entwickelt bzw. entwickeln kann.

In der Zukunftsstudie des Münchener Kreises wurde den befragten Experten unter anderem die These zur Bewertung vorgelegt, ab wann die Bezahlung im Einzelhandel oder Restaurant über das mobile Endgerät (Mobile Wallet) weltweit mit einheitlicher Technologie möglich ist. Das Ergebnis überrascht insofern, als das zwar die überwiegende Mehrheit der Befragten grundsätzlich davon ausgeht, dass dieses Zukunftsbild einmal Realität werden wird, allerdings insbesondere die Deutschland-Experten hier durchaus noch eine lange Wegstrecke erwarten: Erst im Zeitraum 2020 bis 2024, soll laut Branchenkennern dieses Ziel erreicht werden; die US-Experten schätzen den Realisationszeitpunkt deutlich früher ein und gehen von einer Realisation bis spätestens 2019 aus. Den wichtigsten Treiber (62 Prozent) und gleichzeitig die größte Barriere (61 Prozent) stellt dabei nach Einschätzung der Experten für Deutschland die Festlegung von (internationalen) Standards bzw. das Fehlen dieser dar. Ebenfalls ein noch zu lösendes Problem ist nach Ansicht von 55 Prozent der befragten Experten das Thema des Datenschutzes.

Diese Divergenz zwischen den internationalen Märkten spiegeln auch die Befragungsdaten der Mobile Life Studie 2011 wieder. Während weltweit bereits zehn Prozent der Mobilfunknutzer die Möglichkeiten eines „digitalen“ Portemonnaies nutzen, sind vergleichbare Dienste in Deutschland derzeit nur in kleineren Pilotprojekten nutzbar – eine flächendeckende Verfügbarkeit ist noch nicht realisiert. Dennoch ist der potentielle Bedarf nach mobilen Bezahldiensten in Deutschland heute schon absehbar. 16 Prozent der im Rahmen der Studie befragten deutschen Mobilfunknutzer wünschen sich bereits heute, mobil mit dem Handy in Läden oder im Restaurant bezahlen zu können (Abb. 35.7).

Insgesamt ist festzustellen, dass das Thema M-Commerce ein Trend ist, der sich mit dem mobilen Internet sehr gut verbindet und zukünftig zu einer erheblichen Nutzung führen dürfte. Betrachtet man nur die Gruppe der Smartphonenuutzer und interpretiert diese als eine Art Vorreitergruppe in der mobilen Internetnutzung, so zeigt sich, dass M-Commerce in dieser Zielgruppe bereits weit verbreitet ist. Erste Messdaten des Pilotprojekts „Deutschland Pilot Mobile Behave“ untermauern diese Vermutung, denn bereits die Mehrheit der Smartphonenuutzer dieser Studie nutzt regelmäßig M-Commerce-Angebote im mobilen Internet. Bereits ein Fünftel der Probanden hat schon entsprechende Applikationen auf seinem mobilen Endgerät installiert.

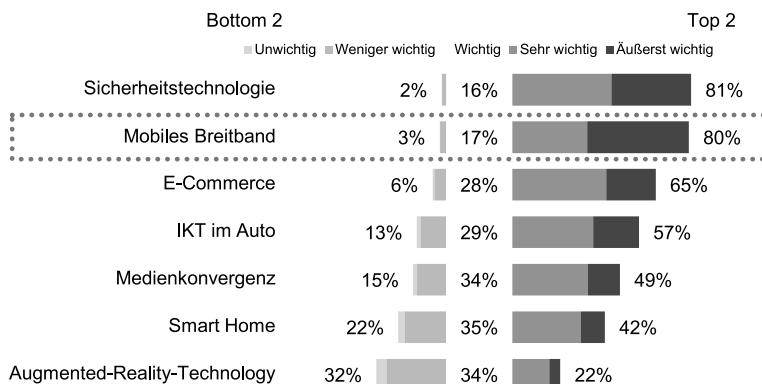


**Abb. 35.7** Mobile Commerce – Bezahlung im Einzelhandel oder Restaurant über das mobile Endgerät. These: Bezahlung im Einzelhandel oder Restaurant über das mobile Endgerät (Mobile Wallet) ist weltweit mit einheitlicher Technologie möglich, in Prozent. Quelle: Münchener Kreis et al. (2009); Basis: Experten für Deutschland/USA

Auch Dienste und Anwendungen rund um mobile Finanzen sind in diesem Kontext zu nennen und ergänzen das Thema M-Commerce. Sechs Prozent der befragten deutschen Mobilfunknutzer der Mobile Life Studie 2011 geben an, dass sie bereits heute über ihr mobiles Endgerät ihren Kontostand überprüfen. Weitere 14 Prozent der Befragten würden dies sehr gerne tun, sind aber zum jetzigen Zeitpunkt noch nicht in der Lage – ein mobiler Dienst, der schon heute großes Potential verspricht. Und auch hier verstärken die Messdaten die in der Befragung gefundenen Potentiale: Die definierte Vorreitergruppe der Smartphonenuutzter zeigt auch in der Nutzung mobiler Finanzseiten einen deutlich erhöhten Wert, der in diesem Fall eng mit dem mobilen Endgerät an sich korrespondiert. Erste Messdaten zeigen, dass die Mehrheit der gemessenen Probanden ihr Smartphone regelmäßig, im Durchschnitt mindestens an jedem zweiten Tag im Monat, dazu verwendet eine Finanzdienstleistung mobil zu nutzen. Interessant ist, dass klassische Apps derzeit nur bei einem kleinen Teil dieser das mobile Internet intensiv nutzenden Anwender installiert sind.

## 35.7 Neue Ökosysteme entstehen

Die bereits eingangs dargestellte Studie des Münchener Kreises und seiner Partner wurde auch im Jahr 2010 weitergeführt und nicht nur die im Beitrag dargestellte große gesellschaftliche Bedeutung des Themas „mobiles Breitband“ wurde in der Befragung von 513 Experten bestätigt, gerade auch die hohe wirtschaftliche Relevanz wurde unterstrichen: Über vier Fünftel der Befragten waren davon überzeugt, dass das Thema mobiles Breitband für die wirtschaftliche Entwicklung Deutschlands als „sehr wichtig“ oder sogar „äußerst wichtig“ einzustufen ist (Abb. 35.8).



**Abb. 35.8** Bedeutung von Technologietrends für die wirtschaftliche Entwicklung in Deutschland, in Prozent. Quelle: Münchener Kreis e.V. et al. (2010); Basis: befragte Fachexperten

Diese Einschätzung der wirtschaftlichen Bedeutung der zukünftigen Mobilfunktechnologien zeigt sich auch in dem sich verändernden „Ökosystem“ mobiler Marken und spiegelt die dargestellten Entwicklungen der Marktstrukturen und der Nutzung durch die Anwender wider: Etwas überspitzt beschrieben, dominierten noch vor wenigen Jahren einige große Telekommunikationsanbieter mit ihren Mobilfunknetzen auf der einen Seite und eine Handvoll Hardware-Hersteller mit ihren Mobiltelefonen auf der anderen Seite den Markt. Dieses Bild hat sich in vergleichsweise kurzer Zeit massiv gewandelt. Als eine Art Branchentreiber fungierte dabei der viel zitierte Eintritt der Firma Apple, der gleich in zweifacher Hinsicht zu den heute offensichtlichen, massiven Veränderungen führte. Zum einen hat Apple mit der Einführung des iPhones und später des iPads angestammte Hardware-Hersteller massiv in Bedrängnis gebracht. Zum anderen, und das ist aus heutiger Sicht die noch bedeutendere Entwicklung, wurde durch die Nutzerfreundlichkeit von iOS und der auf diesem Betriebssystem laufenden Apps, des App-Stores und der Angebote von iTunes in kürzester Zeit auch das Thema „Content“ massiv in den Vordergrund der Nutzung gestellt. Diese Entwicklung macht sich auch im Erfolg weiterer Content- und Serviceanbieter bemerkbar, wie sich auch in den Angeboten und Diensten der Betreiber sozialer Netzwerke zeigt. Ergebnisse der „Mobile Life Studie 2011“ zeigen eindeutig, dass die Markenbindung und damit die Bedeutung von Marken der Telekommunikations- und Hardwarehersteller beim Kunden zunehmend an Bedeutung verlieren – mit Ausnahme von Apple. Gleichzeitig sind Content-Marken die klaren Gewinner im mobilen Ökosystem, ihre Bedeutung für Nutzer steigt zunehmend. Die Ausnahmestellung von Apple lässt sich dabei sicher auch damit erklären, dass es der Marke gelungen ist, ein eigenes – teilweise unabhängiges – Ökosystem aufzubauen, das – obgleich stark abgeschottet – durch seine hohe Anwenderfreundlichkeit für den Kunden derzeit eine hohe Akzeptanz und ein hohes Maß an Vertrauen genießt. Eine ähnliche Entwicklung zeichnet sich auch beim deutlich offeneren System Android von Google ab.

## 35.8 Fazit

Abschließend lässt sich die zukünftige Nutzung des mobilen Internets und seiner Dienste mit all seinen Applikationen, Medien und sonstigen Inhalten im Unterschied zur klassischen Internetnutzung wie folgt zusammenfassen: Der mobile digitale Mensch von morgen wird sich nicht mehr bewusst mit dem Internet verbinden, sondern er wird wie selbstverständlich immer und überall im Internet sein. Situationen und Nutzungskontexte werden die Art und Weise der jeweiligen Nutzung bestimmen und das heutige „always connected“ in ein „always in touch“ wandeln.

Die Herausforderung für die Wirtschaft wird hierbei sein, ein echtes Verständnis dafür zu entwickeln, was „mobil“ wirklich ist und was es für den Nutzer wirklich bedeutet, um dies in entsprechende Geschäftsmodelle umzusetzen.

## Literaturverzeichnis

1. Münchner Kreis e.V., European Center for Information and Communication Technologies (EICT) GmbH, Deutsche Telekom AG, TNS Infratest GmbH (2009) Zukunft und Zukunfts-fähigkeit der Informations- und Kommunikationstechnologien und Medien – Internationale Delphi-Studie 2030. München Berlin
2. Motorola (2011) [http://www.motorola.com/Consumers/US-EN/About\\_Motorola/History/Explore\\_Motorola\\_Heritage/Cell\\_Phone\\_Development](http://www.motorola.com/Consumers/US-EN/About_Motorola/History/Explore_Motorola_Heritage/Cell_Phone_Development), abgerufen am 16. Juni 2011
3. Young and Rubicam (2010) Mobile Mania. London
4. Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen (2011) Jahresbericht 2010. Bonn
5. Deloitte Global Services Limited (2011) Technology, Media & Telecommunications Predictions 2011. London
6. TNS (2011). Mobile Life: Global Telecoms Insights 2011
7. Peterhans, Matthias & Sagl, Stefanie (2011). Digital heißt zukünftig immer mehr: mobil! Veröffentlicht in: media spectrum 04/2011. Kliemo, Eupen (Belgien)
8. Münchner Kreis e.V. European Center for Information and Communication Technologies (EICT) GmbH, Deutsche Telekom AG, TNS Infratest GmbH, Siemens AG, Vodafone D2 GmbH, SAP AG, Telefónica O2 Germany GmbH & Co. OHG, ZDF (2010) Offen für die Zukunft – Offen in die Zukunft. München Berlin
9. Zokem (2011) Deutschland Pilot. Messung der Nutzung von Smartphones

---

# Kapitel 36

## Die mobile „Generation plus“ – Anforderungen und Potenziale

Harald Klaus, Michael C. Balasch, Christoph Nedopil, Sebastian Glende und Marlene Gerneth

**Zusammenfassung** Die Studie „Die mobile „Generation Plus“ – Anforderungen und Potenziale mobiler Apps“ wurde von den Deutsche Telekom Laboratories zusammen mit der YOUSE GmbH durchgeführt. Im Zeitraum von Oktober 2010 bis Februar 2011 wurden Bedürfnisse von Menschen über 50 Jahre in Bezug auf mobile Applikationen ergründet und die Potenziale in diesem noch jungen Markt untersucht. Hierfür testeten Nutzer aus der „Generation Plus“ in Kreativ-Workshops bereits vorhandene mobile Anwendungen und erarbeiteten (Verbesserungs-)Vorschläge in Bezug auf ihre alltäglichen Bedürfnisse. Darauf hinaus wurden durch die Arbeit mit verschiedenen Seniorengruppen Defizite heutiger Smartphones aufgedeckt, und der Bedarf für neue mobile Applikationen sowie die Vermarktungsmöglichkeiten wurden in einer quantitativen Marktforschung eruiert und bewertet.

Vor dem Hintergrund des demographischen Wandels und seiner immensen Herausforderungen stellt der Beitrag die erzielten Ergebnisse der Studie im Detail vor und diskutiert die Marktchancen und Geschäftsmodellansätze für Akteure in unterschiedlichen Marktsegmenten.

---

Harald Klaus  
Deutsche Telekom AG, Laboratories, Berlin,  
E-mail: Harald.Klaus@telekom.de

Michael C. Balasch  
Deutsche Telekom AG, Laboratories, Berlin,  
E-mail: Michael.Balasch@telekom.de

Christoph Nedopil  
YOUSE GmbH, Berlin,  
E-mail: Christoph.Nedopil@youse.de

Sebastian Glende  
YOUSE GmbH, Berlin,  
E-mail: Sebastian.Glende@youse.de

Marlene Gerneth  
Deutsche Telekom AG, Laboratories, Berlin,  
E-mail: Marlene.Gerneth@telekom.de

## 36.1 Ausgangssituation – Der Markt für die mobile „Generation Plus“

Die rasante Entwicklung der vergangenen Jahre zeigt es: Das Internet und dessen mobile Nutzung haben das Leben von Jung und Alt verändert. Die vielfältigen Nutzungsmöglichkeiten eröffnen dem Menschen neue Zugangsmöglichkeiten, beispielsweise zu Informationen, zum Einkaufen oder zu sozialen Netzwerken; das Internet ermöglicht innovative Formen der Arbeit und der globalen Zusammenarbeit.

IKT durchdringt unseren Alltag, verstärkt das Bedürfnis, auch unterwegs verbrachte Zeit produktiv nutzen zu können; persönliche Endgeräte und öffentliche Endsysteme erschließen die mobile digitale Welt der Zukunft [1]; Menschen wollen mobil und flexibel auf anspruchsvolle Datendienste über attraktive mobile Endgeräte zugreifen [2]. Und gerade vor dem Hintergrund des demographischen Wandels hat Deutschland die Chance, Entwicklungen, die hier vorausschauend und frühzeitig entwickelt wurden, zu exportieren und damit zur Referenz für andere Länder zu werden [2].

### 36.1.1 Ungebremstes Wachstum im Online-Bereich

Im Jahr 2010 haben über 72 % der deutschen Bevölkerung das Internet genutzt. Allein bei der „Generation Plus“, d. h. den über 50-jährigen, waren es im gleichen Jahr bereits knapp 50 % – ein Wachstum von 15 Prozentpunkten in drei Jahren (Abb. 36.1).

Die Einführung des Apple iPhone in Deutschland im Jahr 2007 sorgte maßgeblich für einen Durchbruch des mobilen Internets: Seitdem verzeichnetet das Datenvolumen im Mobilfunk in Deutschland zweistellige Zuwachsrate, gekoppelt mit einem ebenso zweistelligen Anstieg der Umsatzzahlen seit 2009 (Abb. 36.2). In den nächsten Jahren wird der Bedarf an leistungsfähiger mobiler Datenübertragung noch weiter ansteigen, die Nachfrage nach mobilen Anwendungen wie Surfen im Internet, Spielen, Navigation, Videos u. Ä. wird weiter zunehmen, und Menschen

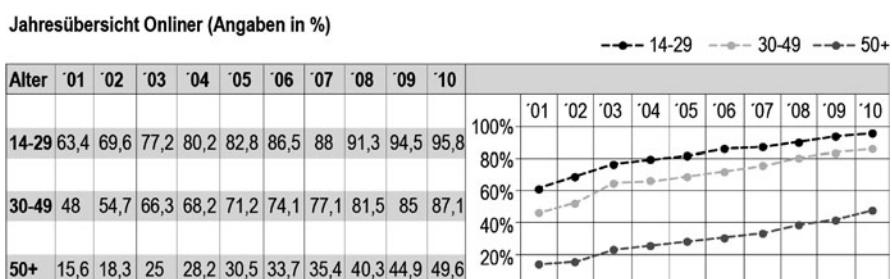
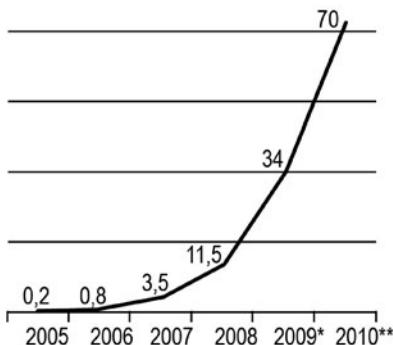


Abb. 36.1 Wachstum im Bereich Internetnutzung (Quelle: [3])

### Im Mobilfunk explodieren die Datenmengen

#### Datenvolumen im Mobilfunk in Deutschland

Mio. Gigabyte



\*Prognose, \*\*Schätzung

#### Umsatz mit mobilen Datendiensten in Deutschland

Mrd. Euro

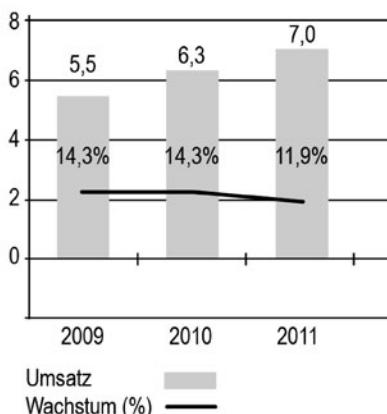


Abb. 36.2 Datenvolumen in Deutschland (Quelle: [4])

möchten mobil und flexibel auf anspruchsvolle Datendienste über immer attraktiver mobile Endgeräte zugreifen [2].

So überrascht es nicht, dass auch die Umsatzerwartungen für mobile Applikationen hoch sind: Für das Jahr 2011 sind laut Prognosen von Forrester weltweit Umsätze von 5 bis 6 Milliarden US-Dollar möglich – das sind 150–200 % mehr als im Jahr 2010 [5]. Wie Bitkom auf der Basis von Daten von research2guidance bekannt gab (Bitkom 2011), ist die Anzahl der App-Downloads in Deutschland in 2010 deutlich gestiegen. So wurden im ersten Halbjahr 2010 346 Millionen Anwendungen heruntergeladen, auf ein Jahr hochgerechnet ergibt das ca. 755 Millionen Downloads – ein Anstieg um 78 % im Vergleich zum Vorjahr. Auch der Umsatz mit Apps konnte gegenüber 2009 drastisch erhöht werden und liegt nun mit einem Plus von 81 % bei voraussichtlich 343 Millionen Euro [6]. Bitkom erwartet für Deutschland im Jahr 2011 bei Smartphones eine Umsatzsteigerung um 24 % auf 2,1 Milliarden Euro [7].

### 36.1.2 Marktpotenziale für Mobile Apps in der „Generation Plus“

Die Absatzzahlen von Smartphones steigen rasant. Alleine im ersten Quartal 2011 wurden weltweit knapp 430 Millionen Smartphones verkauft – ein Plus von 19 % im Vergleich zum Vorjahresquartal. Das heißt, dass weltweit jedes 4. verkaufte Mobiltelefon ein Smartphone ist [8] – in Deutschland laut dem Branchenverband Bitkom sogar jedes dritte [9].



Abb. 36.3 Ausgabenverhältnisse in Deutschland [10]

Da die Absatzzahlen von Smartphones und Apps in den verschiedenen Altersgruppen nicht öffentlich sind, nehmen wir an, dass die Absatzzahlen auch in der „Generation Plus“ über die vergangenen Jahre stark gestiegen sind.

Wenn man dies mit den Ausgabenverhältnissen (s. Abb. 36.3) der „Generation Plus“ vermischt, so ergibt sich ein vielversprechendes Marktpotenzial für Smartphones und Apps: Die Generation Plus tätigte im Jahr 2010 in Deutschland über 50 % der Ausgaben. Der Anteil an Ausgaben der „Generation Plus“ wird im Laufe der nächsten Jahre weiter steigen – und das nicht nur in Deutschland.

Wenn man nun die Ausgabenarten als Interessensgebiete der „Generation Plus“ annimmt und in diesen Bereichen Mehrwerte für die „Generation Plus“ schaffen will, gibt der Ausgabenatlas des Statistischen Bundesamts eine erste Orientierung. Hiernach nehmen im Alter die Ausgaben besonders für Wohnung einschließlich Energie und Gesundheitsvorsorge zu. Ausgaben im Bereich Verkehr und Kleidung hingegen sinken im Alter, während Ausgaben für Nahrungs- und Genussmittel, Freizeit und Haushaltsgeräte über die Zeit stabil bleiben [11].

Somit scheinen für die „Generation Plus“ solche Apps besonders interessant, die Mehrwerte in den Bereichen Gesundheitsvorsorge und Nahrungsmittel (z. B. Ernährungs-Apps), Wohnen und Haushaltsgeräte (z. B. Apps zur Steuerung von Smart Homes) oder auch allgemeine Freizeit generieren bzw. vermitteln.

### 36.1.3 Mobile Apps – Unbekanntes Nutzungsverhalten der „Generation Plus“

Obwohl Daten für den Gesamtmarkt „mobile Applikationen“ mittlerweile einen guten Gesamtüberblick bieten, ist bezüglich der Möglichkeiten und Potenziale von mobilen Applikationen für die „Generation Plus“ noch sehr wenig bekannt – ob-

wohl die Online-Nutzung dieser Bevölkerungsschicht im Online-Bereich die höchsten Steigerungsraten verzeichnet [3].

Aus diesem Grund ist es wichtig sich bewusst zu werden, wie attraktive Angebote für diese Generation im Bereich mobiler Applikationen geschaffen werden können. Dabei muss auch berücksichtigt werden, wie diese Generation richtig angesprochen wird und wie damit der Nutzen von Smartphones auch dieser Generation zugutekommen kann. Denn die „Generation Plus“ verfügt über große finanzielle Ressourcen und hat zugleich einen großen – bisher weitgehend ungedeckten – Bedarf in den Bereichen Information, Sicherheit, Mobilität und medizinischer Versorgung. Das Interesse zeigt sich besonders bei einer Vielzahl von AAL-Funktionalitäten (z. B. Vitalfunktions- und Health-Monitoring, das nach [2] in den nächsten Jahren Realität werden wird, Sturzerkennung, Ortung, Kommunikation), die sich mit mobilen Applikationen realisieren lassen.

## 36.2 Methodisches Vorgehen – „Fragt uns!“

Um die Potenziale mobiler Applikationen für die „Generation Plus“ zu erforschen, wurden im Rahmen der Studie „Die mobile ‚Generation plus‘ – Anforderungen und Potenziale“ zunächst die Anforderungen von Senioren an mobile Applikationen in einer Sekundärdatenerhebung gesammelt. Außerdem wurde der Markt für mobile Applikationen anhand von statistischen Auswertungen im Hinblick auf Marktpotenziale und Preissensibilität analysiert.

Im Anschluss an die Sekundärdatenanalyse wurden im Oktober 2010 gemeinsam mit verschiedenen Seniorengruppen drei Kreativ- und Testworkshops mit aufeinander abgestimmten Zielsetzungen und Methodensets mit aktiver Nutzerintegration durchgeführt. Bei den Workshops wurden sowohl die Funktionen als auch die Handhabung mobiler Applikationen getestet und Anregungen zur Weiterentwicklung gesammelt. An den Workshops nahmen neben Senioren auch Softwareingenieure und Designer teil.

Für die Durchführung der Workshops wurden Moderations- und Kreativitätsmethoden verwendet, die u. a. bereits im Rahmen der Begleitforschung des vom BMBF geförderten Forschungsprogramms „Altersgerechte Assistenzsysteme“ eingesetzt wurden [12].

Als neue Methode wurde das von YOUSE entwickelte „Picasso-Puzzle“ (s. Abb. 36.4) eingesetzt. Das Picasso-Puzzle ermöglicht das Hineindenken in verschiedene Situationen, die durch das Ziehen von Puzzle-Teilen bestimmt werden. Mit dieser Methode können innerhalb kurzer Zeit eine Reihe verschiedener Szenarien und Probleme spielerisch durchdacht werden, für die kreative Lösungen konzipiert werden.

Eine deutschlandweite Online-Umfrage rundete die Studie ab. Ziel dieser quantitativen Erhebung war es, die Zahlungsbereitschaft der „Generation Plus“ im Markt der mobilen Applikationen zu erforschen. In dieser Umfrage wurden auch Fragen zur aktuellen Nutzung und den Vorzügen von Mobiltelefonen und Smartphones ge-

**PICASSO-PUZZLE**

Kreativtechniken

1. Kartenentwicklung

2. Karten ziehen und in Situation hineinversetzen

3. Entwicklung von Ideen entsprechend der Rollen

4. Moderation zur Beförderung von Ideen

Copyright © YOUSE GmbH, 2011

**Abb. 36.4** Picasso-Puzzle aus der Innovation Toolbox (©YOUSE 2011) und Anwendung

stellt. Insgesamt umfasste die Umfrage ca. 20 Fragen. Sie wurde über Online Netzwerke (z. B. Wirtschaftsfaktor Alter), Foren und Seniorenorganisationen deutschlandweit publik gemacht. Die Umfrage fand vom 22.10.2010 bis zum 19.1.2011 statt.

### 36.3 Anforderungen und Bedarfe der „Generation Plus“ an Mobile Apps

Die gewonnenen Erkenntnisse aus der Literaturrecherche und der eigenen Studie zeigen, dass die „Generation Plus“ konkrete Bedarfe an mobile Apps hat. Diese reichen von allgemeinen Anforderungen an Smartphones bis hin zu gewünschten Funktionalitäten und zu Anforderungen an die Vermarktung. Die Ergebnisse bilden eine wichtige Grundlage für die zukünftige Entwicklung des Applikationsmarktes für die „Generation Plus“.

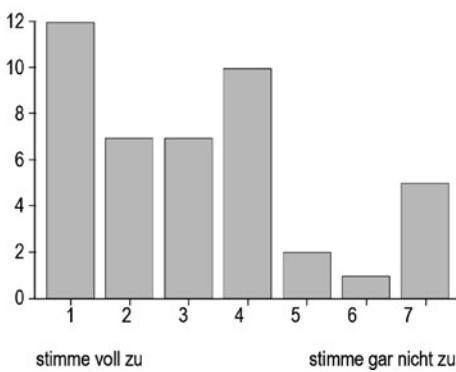
#### 36.3.1 Smartphone – was ist das?

Smartphones sind zum Zeitpunkt der Studie (Dezember 2010) ein weitgehend unbekanntes Terrain für die „Generation Plus“. Wie die deutschlandweite Online-Umfrage<sup>1</sup> ergab, sehen gut die Hälfte der Teilnehmer einen Nutzen von Smartphones, während ca. ein Viertel indifferent waren und ein knappes weiteres Viertel den Nutzen nicht erkannten (siehe Abb. 36.5).

Trotz des geringen Bewusstseins über Smartphones verneint nur etwa jeder sechste der Befragten, dass mobile Applikationen einen Nutzen haben.

Dieses Ergebnis unterstreicht die überaus hohe Nutzenerwartung der Zielgruppe, wenn man bedenkt, dass weniger als ein Viertel der Teilnehmer der Umfrage bisher ein Smartphone benutzen und somit die Mehrzahl der Befragten nur über geringe Erfahrungen mit solchen Geräten verfügen.

**Sie sehen den Nutzen von Smartphones**



**Abb. 36.5** Nutzen von Smartphones ( $n = 44$ )

<sup>1</sup> Trotz intensiver Bekanntmachung der Online-Befragung war die Beteiligung der „Generation Plus“ zu gering, um statistisch belastbare Ergebnisse gewinnen zu können.

### **36.3.2 Allgemeine Anforderungen an Smartphones und mobile Apps**

Eine zentrale Anforderung der „Generation Plus“ ist eine nutzerfreundliche, leicht erlernbare Bedienung. Hierbei ist eine Anlehnung an Bedienabläufe hilfreich, wie sie von nicht-elektronischen Produkten her bekannt sind (z. B. Blättern in EBooks mit der vom echten Buch bekannten Bewegung). Schriftgrößen, Kontraste und Lautstärke sollten individualisierbar sein. Die Bedienungsanleitung sollte leicht verständlich geschrieben sein (d. h. auf deutsch und ohne Technikersprache), ein umfassendes Problemlösungskapitel haben (nicht „FAQ“) und bedarfsgerecht strukturiert sein [13].

Darüber hinaus legen ältere Nutzer insbesondere bei hochpreisigen Geräten Wert auf effektiven Diebstahlschutz und eine lange Lebensdauer. Ein Großteil der Testpersonen erwartet außerdem Kompatibilität mit bereits vorhandenen Speichermedien, eine klare Kostentransparenz bei der Nutzung sowie eine unmerkliche, kaum Aktionen erfordерnde Aktualisierung der Software.

### **36.3.3 Gewünschte Funktionen von mobilen Apps**

Die „Generation Plus“ ist an Applikationen mit konkretem Nutzen im Alltag interessiert. Abbildung 36.6 zeigt verschiedene Kategorien:

#### **36.3.3.1 Information und Kommunikation**

Unterwegs interessieren ältere Menschen besonders aktuell nutzbare Informationen, z. B. Hilfe beim Einkaufen (Produktinformation, Preisvergleich, Alternativangebo-

<b>Notruf: „Christophorus“</b>	<b>Navigation: „Kultur-Lotse“</b>
 <ul style="list-style-type: none"> <li>Notruf via Bewegungssensor</li> <li>Abruf medizinischer Daten</li> <li>Exakte Lokalisierung</li> </ul>	 <ul style="list-style-type: none"> <li>Auffinden von Geschäften, Dienstleistungen, Restaurants</li> <li>Verstehen von fremden Kulturen und medizinischen Informationen</li> </ul>
<b>Soziales: „Schatzsuche“</b>	<b>Freizeit/Reise: „Traumbett“</b>
 <ul style="list-style-type: none"> <li>Auffinden von Gleichgesinnten</li> <li>Gemeinsame Besuche von Veranstaltungen</li> </ul>	 <ul style="list-style-type: none"> <li>Finden der passenden Unterkunft</li> <li>Buchen der Unterkunft</li> <li>Erfahren von Randbedingungen (z.B. Seniorenfreundlichkeit, Parkplatz)</li> </ul>

**Abb. 36.6** Gewünschte Funktionen von mobilen Apps für die „Generation Plus“ (Ergebnisse der Studie, Auswahl)

te) oder Unterstützung bei Reisen (Zugverbindungen, Information über Sehenswürdigkeiten).

Bei der Kommunikation wird neben der Telefon- und SMS-Funktion auch die mobile E-Mail-Funktion zunehmend gewünscht – und wo vorhanden auch genutzt.

### **36.3.3.2 Unterstützung der Selbst-Organisation**

Die Workshopteilnehmer erwarten einen hohen persönlichen Gewinn in der mobilen, integrierten Nutzung von Organizer-Funktionen (Adressbuch, Kalender, Dokumente, Taschenrechner), die sie bereits von herkömmlichen Mobiltelefonen her kennen.

### **36.3.3.3 Navigation und Mobilitätsunterstützung**

Eine einfache, jederzeit verfügbare Navigation wird als einer der größten Vorteile von Smartphones gegenüber herkömmlichen Mobiltelefonen angesehen. Interessant erscheinen vielen die bereits verfügbaren Applikationen zum Auffinden von Geschäften, Dienstleistungen oder Restaurants (Location Based Services). Senioren wollen das Smartphone sowohl zur Fußgänger- als auch zur Kfz-Navigation nutzen.

Es besteht jedoch nach Aussagen der Befragten noch Optimierungsbedarf bei der nahtlosen Navigation „von Haus zu Haus“, beim Umsteigen zwischen Verkehrsmitteln und bei der Fußgängernavigation. Verbesserungen sind auch dort nötig, wo spezifische körperliche Einschränkungen die Mobilität begrenzen. So sollten Routen entsprechend diesen Einschränkungen geplant werden können. Viele dieser Fragestellungen werden im Rahmen von F&E-Vorhaben im Förderprogramm „Mobil bis ins hohe Alter – nahtlose Mobilitätsketten zur Beseitigung, Umgehung und Überwindung von Barrieren“ des Bundesministeriums für Bildung und Forschung [14] bearbeitet.

### **36.3.3.4 Notruf**

Eine Notruffunktion wird von mehreren Herstellern seit vielen Jahren angeboten und weiterentwickelt, jedoch sehen gerade die „jungen“ Alten diese in ihrer aktuellen, meist von Dritten leicht erkennbaren Gestaltung als stigmatisierend an.

Die Notruffunktion könnte aber auch als Applikation ausgeführt werden – hierzu wurden in den Workshops konkrete Ideen entwickelt. Beispielsweise könnte ein Notruf auch automatisch ausgelöst werden, wenn der Bewegungssensor im Telefon einen Sturz und anschließend eine längere Bewegungslosigkeit feststellt.

Voraussetzung für die sichere Funktion ist ein detailliertes Nutzerprofil inkl. medizinisch relevanter Daten.

### 36.3.3.5 Soziale Kontakte

Insbesondere alleinstehende Ältere können sich vorstellen, Applikationen zur Herstellung sozialer Kontakte zu verwenden. Auch hierzu wurden im Workshop zwei Ideen für eine Applikation „Schatzsuche“ entwickelt.

Diese soll zum gemeinsamen „Erleben“ anregen und entsprechend eingegebenen Vorlieben Treffpunkte und Veranstaltungen vorschlagen, die man mit Gleichgesinnten besuchen kann. Darüber hinaus beinhaltet die „Schatzsuche“ Möglichkeiten zur Feedbackgenerierung bzgl. getroffener Personen.

### 36.3.3.6 Freizeit und Wohlfühlen

In der Kategorie Freizeit und Wohlfühlen zeigt sich das hohe Interesse besonders deutlich – so wurden in den durchgeführten Workshops vier Applikationsansätze allein für diesen Themenbereich entwickelt.

Die App „Lukullus“ (römischer Feldherr, der für seine üppigen Gastmahlte bekannt war, siehe Abb. 36.7) dient der Suche nach speziellen Kulinaritäten. Diese können in Abhängigkeit unterschiedlicher Kriterien (z. B. Ort, Küche, Preis, Ambiente, Erreichbarkeit) ausgewählt werden.

Als „Traumbett“ bezeichneten die Teilnehmer eine Applikation zur Suche nach passenden Unterkünften in Abhängigkeit ihres Standortes, des Preis-Leistungs-Verhältnisses, der Erreichbarkeit, der Barrierefreiheit, des Wellness-Angebots und der Gästebewertung.

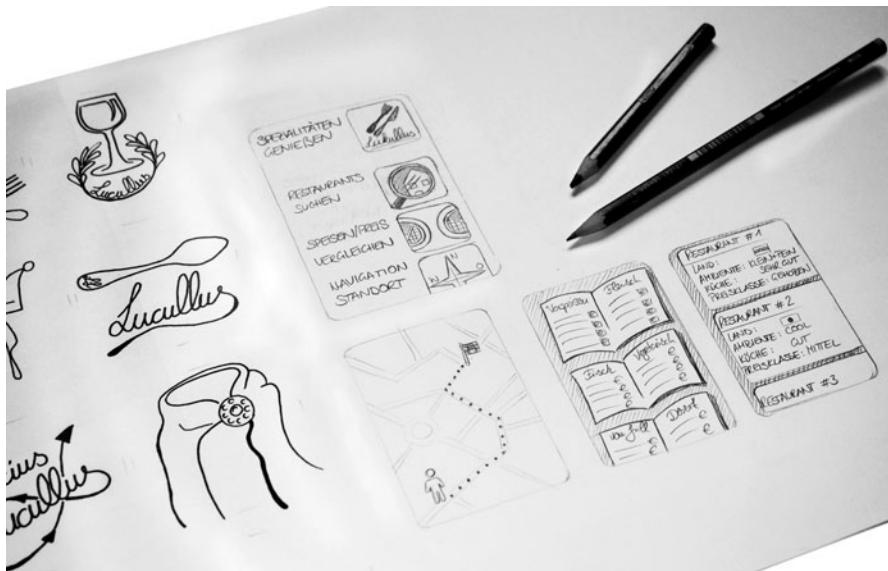


Abb. 36.7 Scribbles für Applikation Lukullus

Die „Lotse“-App dient einem besseren Zurechtfinden in einem fremden Land. Sie beinhaltet neben einem Übersetzungsprogramm auch Informationen über die Landesgeschichte und Kultur sowie über besondere Verhaltensweisen, Sitten und Rituale. In eine solche Applikation könnten interaktive Stadtführungen und Hinweise zur Sicherheit und zu Gefahrenquellen integriert werden.

Zur Entspannung vom Alltagsstress wurde die Anwendung „Wolke 9“ erarbeitet. Diese fasst Möglichkeiten zur Entspannung zusammen und macht diese leicht buch- bzw. verfügbar. So kann der Anwender einerseits auf im Smartphone bereits vorhandene oder downloadbare Inhalte (Musik, Hörspiele) zugreifen, aber auch Wellness-Dienstleistungen und Unterstützung bei anstrengenden Aufgaben buchen (z. B. LieferService beim Einkaufen). Außerdem bietet „Wolke 9“ Anregungen zu Hand- und Gartenarbeit, welche durch die Nutzer-Community sukzessive ausgebaut werden können.

### 36.3.3.7 (Lern-)Spiele

Im Gegensatz zu jüngeren Anwendern spielen Ältere weniger zur Zerstreuung, sondern vielmehr um zu lernen oder die mentale bzw. physische Leistungsfähigkeit zu erhalten. Daher sind in erster Linie hierzu geeignete Spiele (interaktives Sprachtraining oder Denkspiele oder Sportspiele, die Balance und Koordination erfordern) gewünscht.

Nicht zuletzt sieht die „Generation Plus“ Spiele auch als Möglichkeit, die Kommunikation zwischen Jung und Alt zu verbessern. So sind z. B. Spiele gefragt, die von den unterschiedlichen Generationen gemeinsam gespielt werden können und Enkel zum häufigeren Besuch bei den Großeltern motivieren können.

## 36.4 Marktchancen und Geschäftsmodelle

### 36.4.1 Geschäftsmodelle

Die im Rahmen der Studie durchgeführten Interviews mit der „Generation Plus“ zeigen, dass die Akzeptanz und damit die Marktchancen am größten sind, wenn der Nutzen im Vordergrund steht und durch eine zielgruppengerechte Kundenansprache vermittelt wird, wenn die Anwendungen sich einfach bedienen lassen und über ein schnörkelloses, klares Design verfügen – freilich ohne stigmatisierenden Look.

Um die Anforderungen und Wünsche, die mit zunehmendem Alter der Adressaten immer differenzierter werden, durch passende Angebote erfüllen zu können, sind spezifische, sich ergänzende Geschäftsmodelle erforderlich: Viele Anwendungen, die für die „Generation Plus“ besonders wichtig sind – wie im vorstehenden Kapitel vorgestellt, können erst durch die Kombination von Smartphone-Funktionalität und ergänzender Dienstleistung ihren vollen Mehrwert entfalten. Damit bieten sie aber zugleich die Chance von Mehrfacheinnahmen, etwa durch

den Verkauf der App über einen App Store bzw. als webbasierte Applets sowie durch Zusatzdienste, beispielsweise in Form von fachkundiger Beratungsleistung (up-selling).

Weiterhin lässt sich die Zielgruppe auch durch Information und nutzerbezogene Werbung ansprechen, so dass diese dann auch eine Ertragsquelle im Geschäftsmodell darstellt.

Dass insbesondere einfache und nachvollziehbare Funktionen und Mechanismen Marktchancen bieten, wurde am Beispiel des App Store deutlich: Jugendliche und junge Erwachsene erlernten das Konzept des Herunterladens von Musik und Filmen bereits anhand von MP3-Playern und iPods. Sie hatten damit sofort die grundlegenden App-Store-Mechanismen verstanden und verinnerlicht. Diesen Kenntnis- und Erfahrungsvorsprung gilt es für „Generation Plus“ aufzuholen; es bedeutet vielfach ein Neuerlernen dieses neuartigen Nutzungskonzepts und Geschäftsmodells, das binnen weniger Jahre die Handywelt revolutioniert hat.

Der Erfolg von App Stores (und damit selbstredend auch der Smartphones) liegt naturgemäß auch in der Kreativität unzähliger enthusiastischer Software-Entwickler, die über den App Store eine einfach zu nutzende Plattform haben, um ihre Programme weltweit den Smartphone-Nutzern anbieten zu können. Zu welchem Preis sie das tun, ist ihnen weitgehend überlassen.

### ***36.4.2 Anforderungen an die Vermarktung***

Wie die Ergebnisse der Befragung und der Workshops zeigten, stellt für die „Generation Plus“ Mundpropaganda durch Freunde und Bekannte, die als kompetent wahrgenommenen werden, das größte Potenzial zur Kaufmotivation dar. Auch gut ausgebildetes Fachpersonal und seriöse Tageszeitungen und Fachzeitschriften haben einen starken Einfluss auf das Kaufverhalten. Heute übliche Werbung wird überwiegend als unglaublich, überfrachtet und oft unverständlich bewertet.

Gewünscht wird eine sachliche, ästhetisch gestaltete Information statt reiner Werbesprache. Die Kommunikation soll einfach (wenig Fachwörter, kein „denglisch“), seriös und offen (also ohne verdeckte Zusatzbedingungen im Kleingedruckten etc.) sein.

Bei bisher wenig bekannten Produkten wie Smartphones lassen sich Senioren in erster Linie von dem Potenzial zur Lösung alltäglicher Probleme (z. B. durch Apps für ÖPNV-Planung, Wetter, kulturelle Veranstaltungen oder Navigation) begeistern. Dieses Potenzial muss anhand leicht verständlicher Beispiele dargestellt werden; technische Hintergrundinformationen wirken hingegen eher verwirrend.

Darüber hinaus sind ältere Kunden insbesondere an transparenten Informationen über laufende Kosten, Qualität, Nachrüstbarkeit und natürlich die einfache Bedienbarkeit interessiert.

Um den Nutzen besser an die „Generation Plus“ zu kommunizieren, wünschen sich die Nutzer eine aktive Vermarktung der Smartphones mit Erklärungen und Schulungen, beispielsweise während besonderen Themenwochen in Einkaufszentren.

tren, wie ein Nutzer beschreibt: „Es wäre vom Verkauf sehr hilfreich, Workshops für die Kunden anzubieten, in denen Geräte und die Funktionsweise der einzelnen Features vorgestellt werden“.

Die Kommunikation mit den Senioren ist auch auf der Verpackung der Smartphones entscheidend. Um die Kaufwahrscheinlichkeit zu erhöhen, sollte darauf geachtet werden, in sachlicher – jedoch nicht trockener – Nutzersprache darzustellen, was das Smartphone für Nutzungsmöglichkeiten bietet, und besondere Eigenschaften hervorheben (z. B. lange Lebensdauer, großes Display, lange Akkulaufzeit).

## 36.5 Fazit und Ausblick

Das Ziel der Studie war das Erkennen von Potenzialen und die Entwicklung von konkreten Konzepten für mobile Applikationen für die „Generation Plus“. Hierfür wurde ein Methoden-Mix aus quantitativen Methoden (Online-Umfrage) und qualitativen Methoden (Nutzerworkshops mit Kreativmethoden wie beispielsweise 6-3-5-Methode oder Picasso-Puzzle) verwendet. Diese Methoden haben sich für die Fragestellung als zielführend erwiesen (dies wurde auch von der vorhandenen Erfahrung einiger Nutzer mit den Methoden begünstigt).

Die Ergebnisse zeigen zudem, dass ein großes Interesse an Smartphones und den dazugehörigen mobilen Applikationen besteht, jedoch das Wissen über Smartphones in der „Generation Plus“ noch nicht sehr ausgeprägt ist. Um Smartphones auch in dieser Generation besser zu etablieren, müssen Anpassungen sowohl bezüglich Hardware als auch bezüglich Software vorgenommen werden. Diese umfassen zum Beispiel Touchscreens, die besser bei trockener Haut funktionieren und die ein besseres taktiles Feedback geben. Bezuglich der Software muss mehr auf deutsche, verständliche Sprache Wert gelegt werden und die Tastatur größer programmiert werden. Auch Applikationen sollten kundenspezifisch entwickelt werden, wie z. B. Notrufapplikationen oder Applikationen zum Aufrechterhalten und Initiieren sozialer Kontakte. Wenn diese Anforderungen erfüllt sind, kann auch der Nutzen besser an die „Generation Plus“ kommuniziert werden, beispielsweise in Form von speziellen Events und Workshops in Kaufhäusern oder Telefonläden.

Zusammenfassend kann festgestellt werden, dass Smartphones gegenüber marktüblichen „Seniorenhandys“ eine Vielzahl von Vorteilen aufweisen (z. B. Design, Funktionsvielfalt), jedoch Berührungsängste überwunden werden müssen und das Angebot noch besser auf die Zielgruppe abgestimmt werden muss.

Zu einem ähnlichen Ergebnis kommt auch eine im März 2011 abgeschlossene Usability-Untersuchung des Central European Institute of Technology (CEIT) an iPad Tablet-PCs [16]. Obwohl alle Probanden – sämtlich älter als 65 Jahre – die Frage, ob das Gerät einfach zu bedienen ist, mit „ja“ beantworteten, wurde von den meisten der Testpersonen ohne Interneterfahrung eingeräumt, dass der Umgang doch etwas an Übung benötigt. „Wenn man geübt hat, ist es einfach – man muss eben lernen, wo man draufdrückt und was dann passiert.“ Die Meinung der Internet-erfahreneren hingegen war etwas differenzierter: Nur etwa die Hälfte der Personen

würde das iPad gegenüber ihrem normalen PC bevorzugen. Als Gründe hierfür wurden genannt, dass manche Funktionen mit dem Touchscreen umständlicher sind.

Ergebnisse aus der Studie werden daher in weitere Entwicklungen der Deutschen Telekom einfließen, wie es beispielsweise auch beim Telefon Sinus A201 geschehen ist (dieses Telefon wurde von den Deutsche Telekom Laboratories gemeinsam mit Nutzern der „Generation Plus“ entwickelt). Die Ergebnisse dienen auch dazu, schon bestehende Prototypen (z. B. taktile Touchscreendisplays, Sturzerkennung) auf die Bedürfnisse der „Generation Plus“ auszurichten.

Die Studie zeigt deutlich, dass Smartphones und mobile Applikationen das Interesse der „Generation Plus“ wecken und ein interessantes Marktsegment mit enormen Wachstumschancen darstellen.

## Literaturverzeichnis

1. Münchener Kreis, Deutsche Telekom AG, TNS Infratest, EICT (Hrsg) (2008): Zukunft und Zukunftsfähigkeit der deutschen Informations- und Kommunikationstechnologie – Abschlussbericht der ersten Projektphase
2. Münchener Kreis, EICT, Deutsche Telekom AG, TNS Infratest (Hrsg) (2009): Zukunft und Zukunftsfähigkeit der Informations- und Kommunikationstechnologien und Medien – Internationale Delphi-Studie 2030
3. Initiative D21 e.V. und TNS Infratest (2010): „(n)Onliner Atlas 2010“, Berlin
4. Bitkom (2011): „Das mobile Internet boomt“, [www.bitkom.de](http://www.bitkom.de)
5. Bilton, Nick (2011): „Mobile App Revenue to Reach \$38 Billion by 2015, Report Predicts“, Bits, February 28, 2011
6. Horn V (2010) 755 Millionen App Downloads in 2010. <http://www.androidapptests.com/755-millionen-app-downloads-in-2010.html>. Datum: 9. September 2010.
7. Bitkom (2011b): „Das mobile Internet boomt“, [www.bitkom.de](http://www.bitkom.de)
8. Hassel F (2011) Unaufhaltsamer Siegeszug der Smartphones. [http://www.welt.de/print/die\\_welt/wirtschaft/article13383287/Unaufhaltsamer-Siegeszug-der-Smartphones.html](http://www.welt.de/print/die_welt/wirtschaft/article13383287/Unaufhaltsamer-Siegeszug-der-Smartphones.html). Datum: 20. Mai 2011
9. Bitkom (2011c) [http://www.bitkom.org/de/presse/66442\\_65897.aspx](http://www.bitkom.org/de/presse/66442_65897.aspx)
10. Buslei H, Schulz E (2007) Wachsende Bedeutung der Haushalte Älterer für die Konsumnachfrage bis 2050. Wochenbericht des DIW Berlin 23/2007:361–366
11. Statistisches Bundesamt (2011) Private Konsumausgaben und verfügbares Einkommen, 1. Quartal 2011. Statistisches Bundesamt Deutschland. Wiesbaden, 2011. <http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/Internet/DE/Content/Publikationen/Fachveröffentlichungen/VolkswirtschaftlicheGesamtrechnungen/Inlandsprodukt/Konsumausgaben5811109103245.psm1>
12. YOUSE, Institut für Arbeitswissenschaften (AwB) der TU Berlin (2011): YOUSE/AwB Innovation Toolbox
13. Glende S (2010) Senior User Integration – Konzepte, Werkzeuge und Fallbeispiele. Süddeutscher Verlag für Hochschulschriften.
14. BMBF (2010) Mobil bis ins hohe Alter – nahtlose Mobilitätsketten zur Beseitigung, Umgehung und Überwindung von Barrieren. Bekanntmachung vom 21.09.2010. Bonn
15. Senior Research Group der TU Berlin. [www.srg-berlin.de](http://www.srg-berlin.de)
16. Werner F, Werner K (2011) myTablet – Fenster in die Welt. Studie zu Tablet PCs als seniorengerechtes Internetportal. Central European Institute of Technology. Schwechat, März 2011

---

# Kapitel 37

## Smart Mobile Apps in der Lehre

Martin Wagner und Bernd Brügge

**Zusammenfassung** Der mobile Einsatz von Computern hat zu neuen Herausforderungen in der Hardware- und Softwareentwicklung geführt. Bereits durch die geänderten Rahmenbedingungen bei tragbaren Computern, wie begrenzte Batterieleistungen, kleine Bildschirme und neue Eingabegeräte, ergaben sich neue Problemstellungen. Mobile Computer wie zum Beispiel Laptops können zwar gut transportiert werden, sind jedoch während des Transports nicht gut einsetzbar. Smartphones hingegen sind mobil wesentlich leichter einzusetzen und haben eine hohe Akzeptanz beim Endnutzer. Um diese Akzeptanz zu erreichen, müssen Aspekte wie Usability bei der Entwicklung von Smartphones beachtet werden. Damit steigt auch die Komplexität des Entwicklungsprozesses.

Dieser Artikel beschäftigt sich mit der Entwicklung von Smart Mobile Apps und wie man diese lehrt. Er basiert auf Erfahrungen, die in mehreren Praktika im Smart-Mobile-App-Umfeld seit 2007 gesammelt wurden. In diesen Praktika steht Usability im Vordergrund. Wir beschreiben ein Vorgehensmodell, das auf dem Unified Process aufbaut und außerdem Aktivitäten wie Usability Testing und Prototyping enthält. Dann beschreiben wir unser Projektmanagementmodell Rugby, das in Anlehnung an Scrum entstanden ist. Der Entwicklungsprozess selbst ist modellbasiert, wobei wir evolutionäres Prototyping benutzen. Hierbei kommen UML-Zustandsdiagramme sowie Storyboarding zum Einsatz. Zum Abschluss stellen wir einige besonders erfolgreiche Projektbeispiele vor.

---

Martin Wagner

Technische Universität München, Lehrstuhl für Angewandte Softwaretechnik, Institut für Informatik,  
E-mail: wagmarti@in.tum.de

Bernd Brügge

Technische Universität München, Lehrstuhl für Angewandte Softwaretechnik, Institut für Informatik,  
E-mail: bruegge@in.tum.de

## 37.1 Rahmenbedingungen

Seit dem Sommersemester 2007 bietet der Lehrstuhl für Angewandte Softwaretechnik der Technischen Universität München ein Praktikum im Bereich der mobilen Anwendungsentwicklung an [1]. Eine Besonderheit des Praktikums ist die enge Zusammenarbeit mit der Wirtschaft. Pro Semester arbeiten wir mit bis zu zehn Problemstellern aus der Industrie, welche für die Studenten die Rolle des Kunden einnehmen. Die entstehenden Problemstellungen werden dann von bis zu 60 Studenten bearbeitet. Die Studenten werden in Teams von bis zu acht Studenten aufgeteilt, wobei jedes Team einer Problemstellung zugeordnet wird. Die Teams müssen ausgehend von den Anforderungen innerhalb von drei Monaten eine funktionierende Smart Mobile App fertigstellen. Einige dieser Anwendungen sind consumer-orientierte Stand-Alone-Anwendungen. Die anderen sind in die Softwarelandschaft des Kunden eingebettet. Die Softwarearchitektur dieser Anwendungen ist im Allgemeinen eine Client-Server-Architektur mit einem mobilen Endgerät als Frontend, das auf die Softwarelandschaft des Kunden zugreift. Die Bandbreite der Projekte reicht von Greenfield Engineering über Interface Engineering bis hin zum Reengineering. Abhängig von der Softwarelandschaft des Kunden müssen sich die Studenten zusätzlich in neue Entwicklungsumgebungen einarbeiten. Diese können Entwicklungswerzeuge wie zum Beispiel Xcode oder Eclipse, aber auch Sprachen und Protokolle wie Objective C, Java, XML, Corba und TCP/IP sein. Als Zielplattform für den mobilen Client haben wir die iOS-Plattform von Apple gewählt. Auf dieser lassen sich mit der Entwicklungsumgebung Xcode Apps für das iPhone, das iPad und den iPod Touch entwickeln. Wie bei anderen mobilen Plattformen wie Android, RIM und Symbian werden dabei die Studenten vor ähnliche Herausforderungen hinsichtlich Usability Engineering und Softwareengineering gestellt. Im Laufe der letzten vier Jahre wurden von fast 200 Studierenden im Rahmen dieses Praktikums über 30 Apps entwickelt. Dabei wurde mit 20 unterschiedlichen Themenstellern zusammengearbeitet. Um den Geheimhaltungsanforderungen der Themensteller gerecht werden zu können, teilen wir den Studenten bereits bei der Ankündigung des Praktikums mit, dass sie eine Geheimhaltungserklärung unterschreiben müssen. Den Studenten wird die detaillierte Problemstellung – dies findet in Absprache mit dem jeweiligen Problemsteller statt – erst nach der Unterzeichnung der Geheimhaltungserklärung bereitgestellt.

## 37.2 Scrum

Scrum ist ein agiles Framework zur Softwareentwicklung. Agil bedeutet, dass auf sich ändernde Anforderungen und Technologien schnell und ohne hohen zusätzlichen Aufwand reagiert werden kann. Der *ProductOwner* ist die Person, die Anforderungen an das gewünschte Produkt definiert und priorisiert. Diese priorisierte Menge aller Anforderungen nennt sich *ProductBacklog* und darf ständig vom *ProductOwner* verändert werden. Auf der Entwicklungsseite gibt es das *Scrum Team*,

das sich aus den Entwicklern zusammensetzt, und den *Scrum Master*, der das Team unterstützt und für die Einhaltung der Prinzipien von Scrum verantwortlich ist. Zu Beginn einer Entwicklungsiteration (*Sprint*) wählt der ProductOwner im *Sprint Planning Meeting* die von den Entwicklern umzusetzenden Anforderungen aus dem ProductBacklog aus. Die selektierten Elemente definieren den *Sprint Backlog* und dürfen während des Sprints nicht mehr verändert werden. Der Sprint Backlog soll so definiert werden, dass ein demonstrierbarer Bestandteil des späteren Produkts (*ProductIncrement*) entwickelt wird. Endet ein Sprint, so werden im *Sprint Review Meeting* die Ergebnisse dem ProductOwner und Endanwendern (*Stakeholder*) vorgestellt und die Ziele werden mit dem tatsächlich Erreichten verglichen. Zusätzlich besprechen die Entwickler während eines weiteren Meetings (*Sprint Retrospective*), was während des Sprints gut oder schlecht gelaufen ist und wie man sich im nächsten Sprint verbessern kann. Agiles Reagieren auf Änderungen wird durch die kurzen Sprints, die in der Regel zwischen einer und vier Wochen dauern, ermöglicht. Der durch Anforderungsänderungen entstehende Aufwand wird reduziert, indem über den aktuellen Sprint hinaus nur der nötigste Planungsaufwand betrieben wird und sich die Anforderungen während eines Sprints nicht verändern können. Eine weitere wichtige Eigenschaft von Scrum ist, dass sich die beteiligten Personen im Projekt kontinuierlich über den aktuellen Verlauf informieren. Kommunikation zwischen den Entwicklern und dem ProductOwner findet regulär im Sprint Planning Meeting und dem Sprint Review Meeting statt. Während des Sprints stehen die Beteiligten zum informellen Informationsaustausch in Kontakt. Die Kommunikation innerhalb des Entwicklerteams wird durch das sogenannte *Daily Scrum* gestärkt. Während dieses täglich stattfindenden maximal viertelstündigen Meetings informiert jeder die anderen Teammitglieder über den Fortschritt seit dem letzten Daily Scrum und über die eigene Planung bis zum nächsten Daily Scrum. Probleme, welche die Zielerreichung des Sprints gefährden (*Impediments*), werden ebenfalls erläutert.

### 37.3 Methodologie für die Entwicklung von Smart Mobile Apps

Der Markt für mobile Endgeräte und die dazugehörigen Apps ist über die letzten Jahre hinweg stets gewachsen [2]. Bei der Entwicklung von Smart Mobile Apps spielt Usability eine herausragende Rolle [3]. Im Allgemeinen gehen traditionelle Vorgehensmodelle auf Usability als nichtfunktionale Anforderung nicht explizit ein. Aber auch agile Methoden haben hier ihre Probleme, denn Usability-Anforderungen ergeben sich meistens erst als Folge der Erfahrungen, die während des Entwicklungsprozesses und mit den gelieferten Systemen gemacht werden. Dies berücksichtigen wir in der Methodologie, die wir für das Praktikum entwickelt haben. Sie beruht auf drei zentralen Säulen, die wir in diesem Beitrag vorstellen wollen: Die erste Säule sind agile Methoden. Hier verwenden wir eine an die universitären Rahmenbedingungen des Praktikums angepasste Adaption von Scrum, welche wir Rugby nennen. Auf Rugby gehen wir im folgenden Abschnitt genauer ein. Eine besondere Rolle bei Herausarbeitung der Usability-Anforderungen nimmt das Pro-

totyping ein, die zweite Säule unserer Methodologie. Auf der Vorgehensebene, der dritten Säule, orientieren wir uns an den Workflows des Unified Process Modell die wir um die Workflows Prototyping und User Feedback erweitern.

### 37.4 Yet Another Scrum: Rugby

Wie bei Scrum benutzen wir den ProductBacklog zur Verwaltung der Anforderungen. Diese Anforderungen werden von Scrum Team und ProductOwner zu Beginn des Projektes gemeinsam erarbeitet und vom ProductOwner priorisiert. Abweichend von Scrum haben Sprints in der Regel nur eine Dauer von ein bis zwei Wochen. Zu Beginn jedes Sprints wird wie in Scrum der Sprint Backlog festgelegt. Während des ersten Sprints übernimmt der Projektleiter<sup>1</sup> alle Aufgaben des Scrum Masters, z. B. das Meetingmanagement und die Kommunikation mit dem ProductOwner. Unabhängig von der Sprintlänge treffen sich die Studenten mit dem Projektleiter wöchentlich zum allgemeinen Informationsaustausch. Dieses Meeting erweitert die Agenda des Daily Scrums um die Besprechung technischer Fragestellungen und organisatorischer Punkte. Diese Erweiterung ist notwendig, da das Scrum Team nicht im selben Büro arbeitet. Fällt dieses wöchentliche Meeting auf ein Sprintende, werden die Ergebnisse des Sprints präsentiert und der Sprint Backlog des nächsten Sprints definiert. Die Teilnahme des ProductOwners ist hierbei fakultativ. Neben diesen periodischen Scrum Meetings können auch außerplanmäßige Meetings einberufen werden. In diesen kann dann schnell auf für das Projekt bedeutende Ereignisse reagiert werden. Das kann zum Beispiel ein Meeting zwischen zwei Teams sein, die sich im Rahmen ihrer Problemstellung mit der gleichen Thematik, wie etwa Indoor Navigation, beschäftigen. Im Unterschied zu Scrum erlaubt Rugby, dass der Status für die regulären Meetings schon vorher in ein Projekt-Forum eingestellt wird [4]. Dies ermöglicht eine effizientere Behandlung des Status während des Meetings und eine stärkere Gewichtung der offenen Fragestellungen (*Impediments*) und der Versprechen (*Action Items*). Nach dem ersten Sprint gehen die Tätigkeiten Meeting Management und Kommunikation mit dem ProductOwner an einen der Studenten über. Er ist dann der Meeting Manager, in Rugby *Primary Facilitator* genannt. Die Rolle des Primary Facilitators wird von Sprint zu Sprint unter den Studenten weitergegeben. Diesen didaktischen Ansatz verfolgen wir, um jedem Studenten Einblicke in koordinierende Tätigkeiten zu geben und die Gelegenheit zu bieten, auf diesem Gebiet Erfahrungen zu sammeln. Tätigkeiten sind dabei zum Beispiel das Formulieren und das Nachverfolgen von Action Items sowie das Unterstützen von Entscheidungen bei Konflikten.

Ab dem zweiten Sprint nimmt der Projektleiter nur noch eine beobachtende und beratende Rolle ein. Da wir mehr als ein Projekt haben, findet wöchentlich ein Meeting der Projektleiter statt. In diesem Meeting werden Status und offene Fra-

---

<sup>1</sup> Ein Projektleiter ist entweder ein Doktorand oder jemand, der vorher als Entwickler am Praktikum teilgenommen hat.

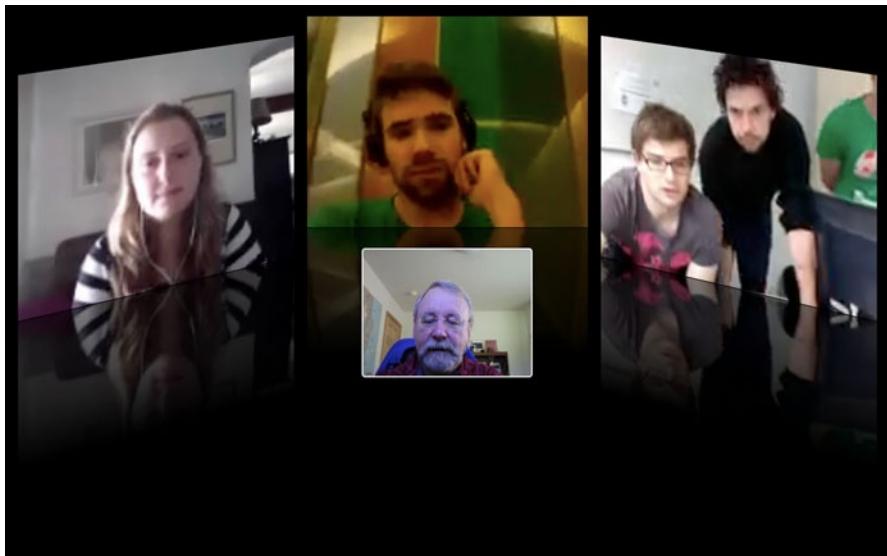


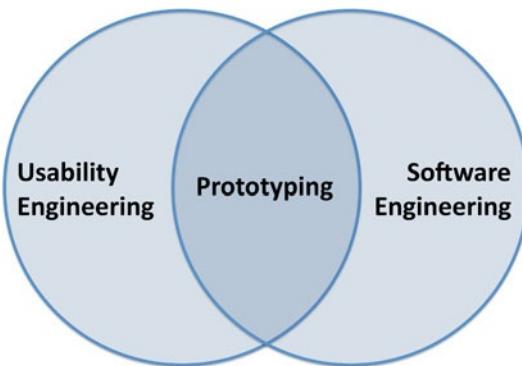
Abb. 37.1 Meetings werden bei Bedarf via iChat abgehalten

gestellungen der einzelnen Teams sowie teamübergreifende Probleme besprochen. Bei der Diskussion werden mögliche Synergien zwischen den Teams identifiziert und daraus resultierende Action Items für die beteiligten Teams formuliert. Zusätzlich organisieren wir ein wöchentliches praktikumsweites Treffen. In diesem werden projektübergreifende Themen behandelt, wie z. B. Projektkickoff mit den Kunden, Icebreaker zum Kennenlernen der Teammitglieder, und es werden Tutorien zur Vermittlung neuer Lehrinhalte abgehalten. Als Werkzeug zur Erstellung der Agenden und Protokolle der Scrum Meetings benutzen wir Unicase [5–7]. Ein Vorteil dieses Werkzeugs ist, dass es möglich ist, Statusberichte [8] bereits vor dem Meeting asynchron einzureichen.

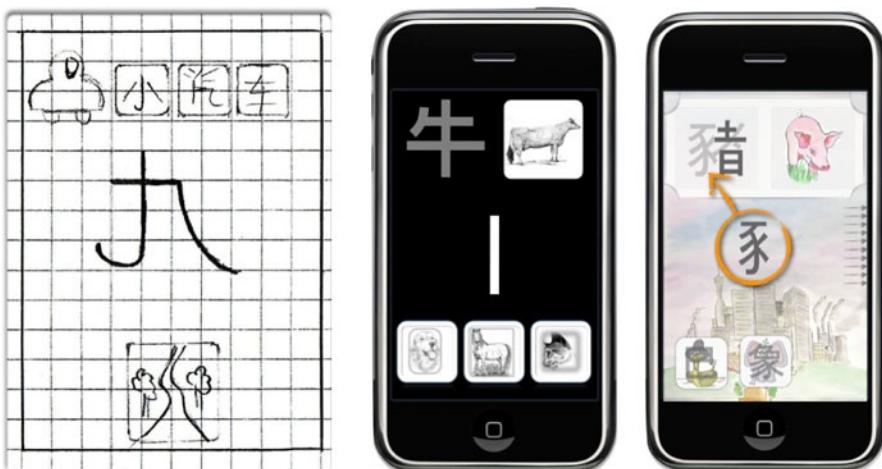
## 37.5 Prototyping

Das traditionelle Softwareengineering beschäftigt sich mit formalen Methoden und Modellen, während im Usability Engineering [9] Prototypen im Mittelpunkt stehen. Usability Engineering unterscheidet zwischen revolutionären und evolutionären Prototypen. Im Softwareengineering hingegen wird zwischen horizontaler und vertikaler Prototypen unterschieden. Ziel in Rugby ist es, von revolutionären horizontalen Prototypen zu evolutionären vertikalen Prototypen zu gelangen.

Das revolutionäre Prototyping (auch exploratives Prototyping genannt) wird oft zur Visualisierung von Anforderungen und Lösungsansätzen verwendet, ohne dass eine lauffähige Instanz des Systems auf der Zielplattform erwartet wird. Damit können Anforderungen veranschaulicht und Informationen zwischen ProductOw-



**Abb. 37.2** Mit Prototyping ergänzt Usability Engineering das klassische Softwareengineering



**Abb. 37.3** Die Evolution vom Low-Fidelity- (*links*) zum high-Fidelity-Prototypen (*rechts*) während der Entwicklung von weMakeWords [10]

ner und Entwickler ausgetauscht werden. Revolutionäre Prototypen durchlaufen einen iterativen Prozess vom Low-Fidelity-Prototypen als erste Veranschaulichung bis zum High-Fidelity-Prototypen, der bereits genaue Vorgaben des späteren Produkts, wie zum Beispiel das Look-and-Feel der Benutzerschnittstelle liefert. Low-Fidelity-Prototypen benötigen keine komplizierten Tools, deren Bedienung erlernt werden muss, und kein technisches Vorwissen und dienen der Kommunikation zwischen Menschen. Zur Erstellung von High-Fidelity-Prototypen werden im allgemeinen Werkzeuge benutzt wie z. B. User-Interface-Generatoren, Frontend CASE Tools oder Bildbearbeitungsprogramme.

Typische Beispiele revolutionärer Prototypen sind Bubble-Diagramme während des Brainstormings, Napkin Designs, papierbasierte Prototypen und haptische Prototypen wie zum Beispiel tragbare Modelle aus Styropor. Der Übergang von High-Fidelity-Prototypen zu evolutionären Prototypen ist fließend. Grafische Elemente

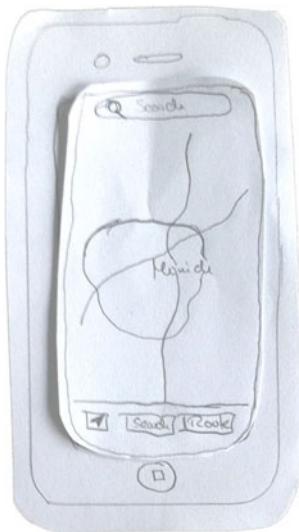


Abb. 37.4 Ein Brainstorming anhand eines papierbasierten Prototypen

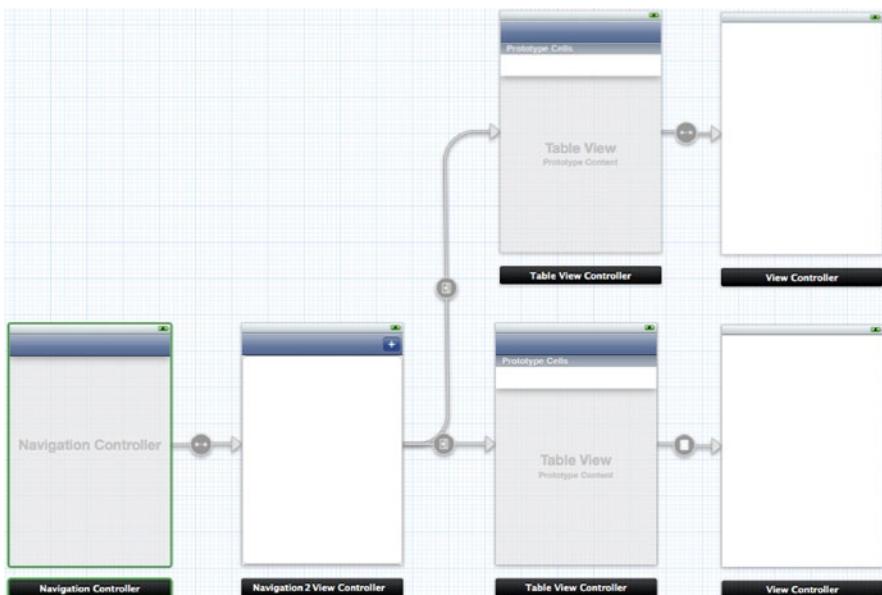


Abb. 37.5 Storyboarding mit Xcode. Die Zustände werden durch geeignete Vorschauen visualisiert. Die möglichen Zustandsübergänge werden durch Linien zwischen den Zuständen dargestellt

eines High-Fidelity-Prototypen, die mit Entwicklungsumgebungen erstellt werden, die GUI-Builder integrieren (wie sie im Rahmen von MDD [11] und MDA benutzt werden), können in das Produkt überführt werden. Prototypen, die in das Produkt

transformiert werden, sind evolutionäre Prototypen. Hier verwenden wir insbesondere die Methode des Storyboarding (siehe Abb. 37.5), in der das Navigationsmodell der Benutzerschnittstelle durch UML-Zustandsdiagramme beschrieben wird. Die Zustände in den Zustandsdiagrammen werden durch UML-Stereotypen visualisiert. Die graphischen Ikonen der Benutzeroberflächen stellen die entsprechenden Zustände im endlichen Automaten dar. Transitionen zwischen den Zuständen sind entsprechende Navigationskommandos in der Benutzerschnittstelle. Das auf der WWDC 2011 vorgestellte Storyboarding ist ab der Version 4.2 in Xcode integriert und wird bereits in unseren Projekten verwendet. Die Storyboards können gemeinsam mit dem ProductOwner und auch Stakeholdern erarbeitet werden. Dies ermöglicht es, die Vorstellungen und Anforderungen des ProductOwners dem Scrum Team unkompliziert und genau mitzuteilen.

## 37.6 Versionsverwaltung und Continuous Integration

Es ist uns wichtig, dass die Studenten mit einer gemeinsamen Codebasis arbeiten. Deshalb fordern wir, dass die aktuellen Sourcen immer in einem Versionsverwaltungssystem stehen. Jedes Team kann sich für eins der beiden Versionsverwaltungssysteme (*Repository*) GIT oder Subversion entscheiden; Subversion hat sich hierbei als das System herausgestellt, in das sich die Studenten leichter einarbeiten können. Der Lehrstuhl stellt die Infrastruktur für die Repositories der Teams bereit. Dies ist wichtig, weil manche Kunden Anforderungen an die Geheimhaltung stellen. Die Studenten müssen ihre Veränderungen mindestens täglich in das Repository einstellen (*einchecken*). Dadurch können die Sourcen mit Hilfe von Continuous Integration nach jeder Veränderung auf ihre Kompilierbarkeit automatisiert überprüft werden. Dies versetzt die Studenten in die Lage, dass sie bei Demonstrationen dem ProductOwner oder den Stakeholdern die letzte funktionierende Programmversion präsentieren können.

## 37.7 Das Vorgehensmodell

Das in Rugby angewandte Prozessmodell basiert auf einer Mischung der Workflows (siehe Abb. 37.6) des Unified Process (UP) und agilen Methoden [12] (siehe Abb. 37.7). Diese Mischung ermöglicht es, auf die besonderen Situationen unserer Studierenden einzugehen. Unterschiedliche Stundenpläne und Studiensituationen der Entwickler führen zu asynchronen und parallelen Arbeitsprozessen. Denn am Praktikum nehmen Bachelor- und Master-Studenten verschiedener Semester teil. Das Unified-Process-modell erlaubt die parallele Ausführung von mehreren Aktivitäten (Workflows genannt) [12].

Aus pädagogischer Sicht ist zunächst eine schrittweise Einführung der Aktivitäten Anforderungsermittlung, Analyse, Entwurf und Implementierung sowie die

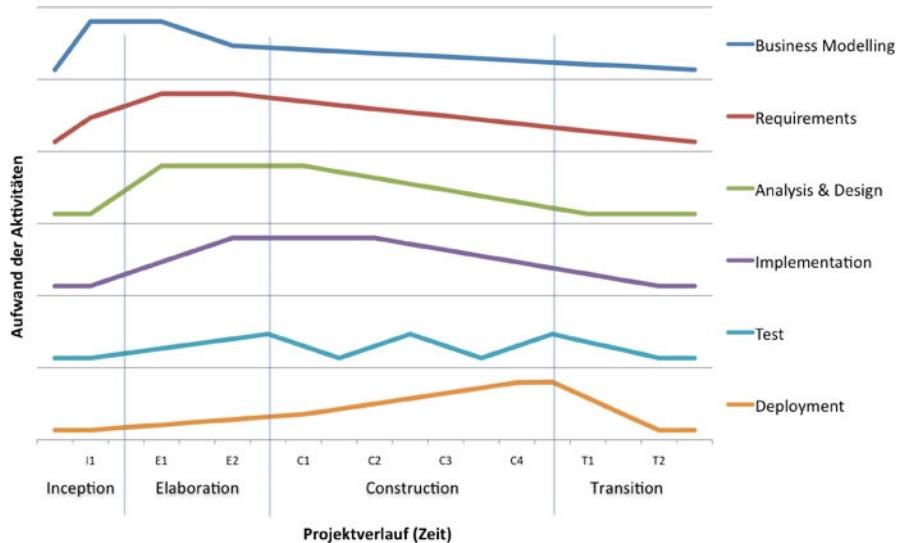


Abb. 37.6 Die Aktivitäten im Unified Process

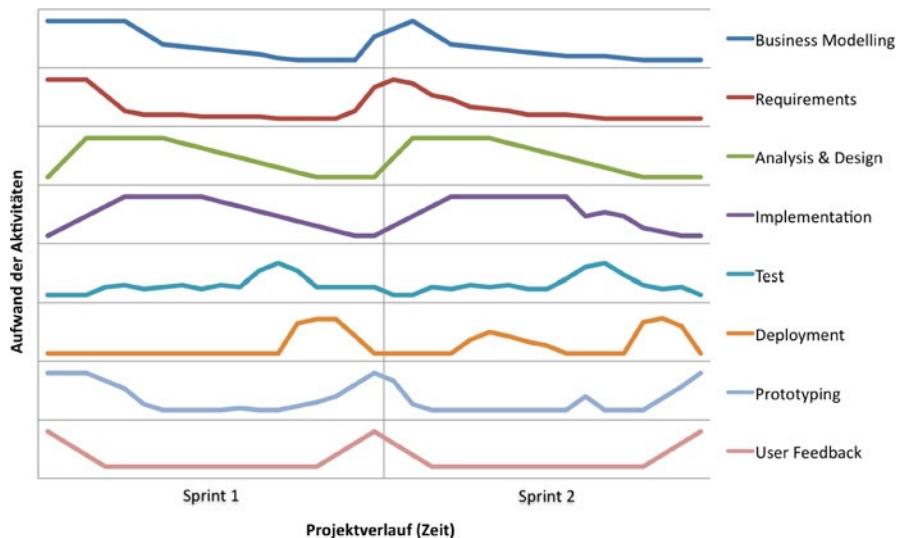


Abb. 37.7 Die Aktivitäten im Scrum

Einführung von Reviews sinnvoll. Haben die Studierenden genug Erfahrung gesammelt, können sie alle Aktivitäten von der Analyse bis zur Implementierung parallel durchführen. Unsere Teams bestehen im Allgemeinen aus Studierenden mehrerer Nationalitäten (in einem laufenden Projektteam sind es sieben verschiedene Nationen) und sind damit sowohl sprachlich wie auch kulturell heterogen. Daher treten typische Herausforderungen auf, die teilweise mit denen in der verteilten Soft-

wareentwicklung in globalen Software-Engineering-Projekten [13, 14] vergleichbar sind. Parallelen sind insbesondere Unterschiede von Sprache und Kultur sowie unterschiedliche Arbeitszeiten und Arbeitsorte. Bei der Elaboration werden im Allgemeinen mehrere Iterationen durchgeführt, wobei wir die Methode des szenariobasierten Entwurfs benutzen. Es hat sich herausgestellt, dass Szenarien ein besonders gutes Instrument sind, um den Studenten die Problemstellung des ProductOwners zu vermitteln.

## 37.8 Beispiele für entwickelte Applikationen

In unseren Praktika wurden von den Studenten mehrere innovative Applikationen über Prototypen hinaus bis hin zur Marktreife entwickelt. Eine kleine Auswahl von Apps, die mittlerweile im Apple AppStore erhältlich sind, möchten wir hier vorstellen.

### 37.8.1 *weMakeWords*

Dieses innovative Lernspiel wurde in enger Zusammenarbeit mit der kinderpsychologischen Praxis Garmisch unter der Leitung von Dr. Pohl entwickelt, wo es auch zur Erforschung der Entwicklung des symbolischen Lernens eingesetzt wird.



Abb. 37.8 Oberflächen des Lernspiels „weMakeWords“

### 37.8.2 *Hirmer Große Größen*

Die iPad App „Hirmer Große Größen“ wurde in Zusammenarbeit mit der Firma reprodukt entwickelt. In dem Online Shop für Übergrößen ist der gesamte Verkaufsworkflow von der Präsentation bis hin zur Bestellung abgebildet.

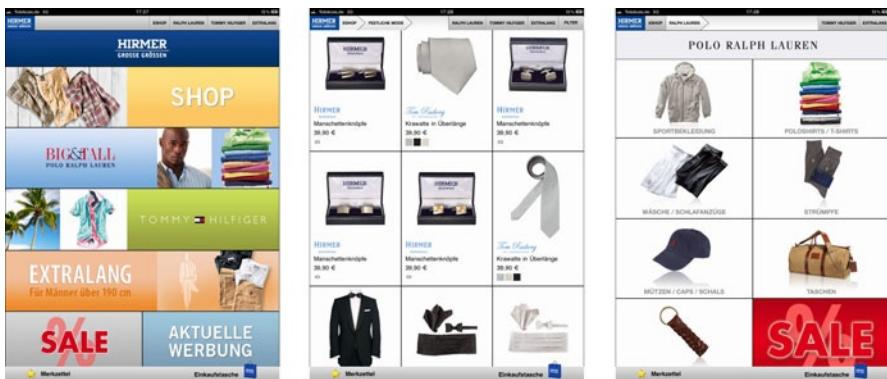


Abb. 37.9 Oberflächen der Shopping App „Hirmer Große Größen“

## 37.9 Fazit

Die vorgestellten Konzepte wurden für die Entwicklung von Smart Mobile Apps im universitären Umfeld eingeführt. In Zukunft werden wir neben der Evaluierung der bestehenden Konzepte deren Übertragbarkeit in Gebiete außerhalb der universitären Lehre erforschen. Wir möchten gerne folgende Thesen überprüfen:

- Rugby bietet sich bei Projekten mit hohem Innovationsanteil und für die Einarbeitung in neue Technologien besonders an.
- Rugby ist ein Konzept, mit dessen Hilfe man im Projekt agile Methoden vermitteln kann.
- Rugby ist flexibler als Scrum einsetzbar und kann deshalb besser an verschiedene Rahmenbedingungen angepasst werden.

Mittlerweile wenden wir unsere Erkenntnisse auch bei weiteren Entwicklungsprojekten des Lehrstuhls an. Es hat sich gezeigt, dass Studenten mit Rugby, einem weniger reglementierten Framework als Scrum, gut zureckkommen. Trotzdem erlernen sie die wichtigen Aspekte der agilen Anwendungsentwicklung und haben später, in einem Umfeld, in dem Scrum eingesetzt wird, schon Erfahrung. Die Kundenakquise fiel uns von Praktikum zu Praktikum immer leichter. Für die Partner aus der Wirtschaft ist eine solche Zusammenarbeit eine willkommene Gelegenheit, Kontakt zu Studenten aufzubauen. In mehreren Fällen resultierte aus einem Praktikum ein Werkstudentenvertrag oder eine Festanstellung nach dem Studium.

## Literaturverzeichnis

1. Internetpräsenzen der Praktika von 2008 bis 2011.  
<https://teambruegge.informatik.tu-muenchen.de/groups/iphone08/> 25. Oktober 2011  
<https://teambruegge.informatik.tu-muenchen.de/groups/iphone09/> 25. Oktober 2011

- <https://teambruegge.informatik.tu-muenchen.de/groups/iphone10/> 25. Oktober 2011  
<https://teambruegge.informatik.tu-muenchen.de/groups/ios11/> 25. Oktober 2011
- 2. asymco.com (2011) „Other“ vendors sell 10 % of Smartphones but 30 % of voice-oriented phones. <http://www.asymco.com/2011/05/22/other-vendors-sell-10-of-smartphones-but-30-of-voice-oriented-phones/> 25. Oktober 2011
  - 3. AsimSmailagic, Dan Siewiorek (1999) User – Centered Interdisciplinary Design of Wearable Computers. ACM Mobile Comput Commun Rev 3(3):43–52.
  - 4. Damian D (2008) iOS Spiel weMakeWords, 2010. Guidelines for successful distributed collaboration. University of Victoria, Canada
  - 5. Bruegge B, Creighton O, Helming J, Kögel M (2008) Unicase – an Ecosystem for Unified Software Engineering Research Tools. ICGSE Workshop
  - 6. Narayan N (2010) What is UNICASE? [https://teambruegge.informatik.tu-muenchen.de/groups/unicase/wiki/bdc81/What\\_is\\_unicase\\_.html](https://teambruegge.informatik.tu-muenchen.de/groups/unicase/wiki/bdc81/What_is_unicase_.html) 25. Oktober 2011
  - 7. Narayan N, Koegel M, Helming J (2010) UNICASE – Tooling for Software Engineering Models. Models Oslo
  - 8. Bruegge B, Dutoit AH (2009) Object-Oriented Software Engineering, Using UML, Patterns and Java. Prentice Hall
  - 9. Nielsen J (1993) Usability Engineering. AP Professional
  - 10. Krusche S, Köhler B, Schubert L (2010) iOS Spiel weMakeWords. <http://www.wemakewords.de> 25. Oktober 2011
  - 11. Selic B (2003) IBM Rational Software. The Pragmatics of Model-Driven Development. J IEEE Software 20(5)
  - 12. Takeuchi H, Nonaka I (1986) The New New Product Development Game. Harvard Business School
  - 13. Wolf T (2008) Rationale-based Unified Software Engineering Model. VDM Verlag Dr. Müller
  - 14. Herbsleb JD (2007) Global Software Engineering: The Future of Socio-technical Coordination. FOSE '07 Future of Software Engineering
  - 15. Carroll JM (ed) (1995) Scenario-Based Design: Envisioning Work and Technology in System Development. John Wiley & Sons

---

## Kapitel 38

# Paul und Paula oder der Spramokom®

## Kommunikation ist nicht alles, doch ohne Kommunikation ist alles nichts!

Wolf von Reden

Wir schreiben das Jahr 2020. Ich möchte Ihnen Paul P. vorstellen. Ich möchte ihn, da er mir so nah ist, schlicht Paul nennen. Paul ist 36 Jahre alt, hat einen guten, auch relativ gut dotierten Job als Vertriebsbeauftragter im strategischen Marketing eines weltweit präsenten Maschinenbauunternehmens, eine Hypothek, ein meist solargetriebenes Elektromobil und ein Magengeschwür. Ach ja, verheiratet ist Paul natürlich auch! Nach Feierabend und am Wochenende treibt er gerne Sport, spielt in einer Rock'n'Roll-Revival-Band und durchlebt Reiseabenteuer auf seinem Mixed-Reality-Headset. Paul hat sich in einem sehr beschäftigten Leben gut eingerichtet.

Seit seinen Jugendzeiten, seit Mitte der neunziger Jahre, benutzt Paul ein Mobiltelefon; das erste war noch ein „abgelegtes“ seines Vaters. Später kam, zur Organisation seiner beruflichen und vor allem privaten Termine, ein PDA, ein persönlicher digitaler Assistent hinzu. Einige Jahre danach schienen die High-Tech-Smartphones schon beinahe alles das zu sein, was er für seinen Alltag und vor allem seine Freizeit benötigte. Die stetig und unübersehbar steigende Flut von kleinen Programmen und Anwendungen, die für wenig Geld und Nutzwert angeboten und etwas geringschätzig „Apps“ genannt wurden, haben ihn nach einer kürzeren Phase der Neugier und eines nur bedingt beruflichen Interesses an all den angepriesenen Hilfefeatures doch bald gelangweilt und seine Smartphones zu einem Grab von selten genutzten und ziemlich sinnlosen Dienstprogrammen werden lassen. Alle Versuche, diese Softwareschnipsel sinnvoll zu integrieren und miteinander ins Einvernehmen zu bringen, hatten doch viel zu viel Zeit gekostet; zumal er auch nie genau wusste, was diese Programmchen genau bewirkten und wie sie sich mit dem Betriebssystem und den anderen Softwareschnipseln vertrugen.

Doch vor kurzem nun hat auch Paul sich entschlossen, sich ein winziges Funkmodul implantieren zu lassen. Ein damit verbundener sprachgesteuerter, mobiler,

---

Wolf von Reden

Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut, Berlin,  
E-mail: wolf.von.reden@hhi.fraunhofer.de

digitaler Organizer und Kommunikator sieht aus wie ein kleiner Ohrclip, aktiviert all seine Anwendungen und Dienste auf einem eigens von dem Betreiber des Kommunikationsportals, einem Service Provider, für den Kunden eingerichteten Portal. Der *Spramokom®* muss nicht aufgeladen werden und bedarf – so wurde Paul versichert – auch keinerlei Wartung. Er trägt den Clip nun seit einigen Wochen und spürt ihn – wie dies bei jedem Ohrring gilt – nicht mehr!

Den *Spramokom®* gibt es seit 2016, in jeder Moderichtung und in jedem Stil und, er ist inzwischen von den üblichen Kinderkrankheiten einer technischen Innovation genesen; insbesondere die Energieversorgung durch Glukosebrennstoffzellen aus Blutzucker und Sauerstoff funktionierte in den ersten Versionen nicht sehr stabil. Doch inzwischen liefert diese körpereigene Energiequelle ausreichend Strom für den Kommunikator, um auch hochratige Datenpakete an die nächste Basisstation senden bzw. von dort empfangen zu können. Da der Clip von dem Service Provider speziell für ihn und nach seinen Vorstellungen entworfen und gefertigt wurde, benutzt Paul nicht einen, seinen *Spramokom®*, sondern spricht mit „Paula“.

Obwohl er weiß, dass Paula nur eine synthetische Stimme ist, scheint sie für Paul ganz wirklich. Nach seinen eigenen Wünschen entspricht Paulas Stimme etwa der einer Frau um Dreißig, die „hannoversches“ Hochdeutsch spricht. Er hatte darum gebeten, dass sie einen leicht sarkastischen Humor haben sollte. Überdies wurde sie so programmiert, seine gelegentliche schlechte Laune mit Lob, aber auch ein wenig Herablassung zu unterlaufen. Er hat natürlich auch den Namen gewählt, mit dem sie angesprochen werden kann . . . „Paula“ (vielleicht sein weibliches Alter Ego oder der Name einer ehemaligen Freundin . . .).

Durch den Stimmerkennungsalgorithmus ist Paula 24 Stunden für Paul und nur für Paul auf Standby. Wann auch immer er „Paula“ murmelt, antwortet sie: „Was kann ich für dich tun, Paul?“ Andererseits eröffnet sie auch das Gespräch durch ein „Paul?“, an das sich beispielsweise der gewünschte Dienst „Erinnerung“ anschließt. Doch Paul kann Paula natürlich immer anweisen, bis zu seiner nächsten Bemerkung ruhig zu bleiben. Der Clip ist so konstruiert, dass nur Paul Paula hören kann. Offensichtlich ist Paula während seines Tages immer für ihn da. Nachfolgend ein Beispieltag, an dem Paula Paul begleitet und seine Stunden vielleicht nicht unbedingt „wertvoller“ und den Mitmenschen aufgeschlossener, aber mit Sicherheit effizienter macht.

Zu einer festgelegten Zeit weckt sie ihn morgens auf, indem sie sanft seinen Namen ruft. Sollte er nicht aufwachen, wird sie es wieder und wieder mit zunehmender Lautstärke und dringlicherem Tonfall so lange versuchen, bis er antwortet. Sobald er reagiert hat, versorgt sie ihn mit den notwendigsten Informationen: Wochentag, Datum, Ort, an dem er sich gerade befindet, lokale Zeit, zu erwartendes Wetter und andere auf Paul zugeschnittene Dienste wie Presseausschnitte, Aktienkurse, Nachrichten oder andere wichtige Dinge aus einer verlässlichen, kostenpflichtigen Datenbank. Natürlich wird sie nicht vergessen, ihn an den Geburtstag seiner Frau zu erinnern und zugleich ein Geschenk zu empfehlen, das sie aus dem erstellten Profil der Geschenke der vergangenen Jahre extrapoliert. Dazu wird sie einen passenden Blumenstrauß bestellen und eine Glückwunschkarte entwerfen. Während Paul sich duscht und langsam auf den Tag einlässt, liest Paula ihm Mails bzw. Sprachnach-

richten und die Einträge aus seinem sozialen Netz in der von ihm gewünschten Gewichtung vor.

Anschließend nimmt sie das Diktat für Mails oder Posts für seine Site im Social Web auf oder verbindet zu Anrufern, die auf der Mailbox gespeichert sind. Natürlich kann sie auf Anweisung oder aufgrund von Befindlichkeitsanalysen ein Musikstück an das häusliche Tonsystem übermitteln bzw. dort aktivieren. Nebenbei wird sie die Kaffeemaschine und den Toaster anwerfen. Als Verkehrsplaner wird sie Paul beraten, welches Verkehrsmittel an diesem Morgen für seinen Tagesablauf zu empfehlen ist.

Sie wird ihn sicher von seinem gegenwärtigen Ort – meistens an allen Verkehrsstaus vorbei – an sein Ziel navigieren und überdies wissen, wo und wie er sein Auto parken kann. Seine täglichen Aufwendungen wie Parkgebühren, Tickets etc. aber auch Besorgungen und Bestellungen werden von ihr vollautomatisch und manipulationssicher verschlüsselt über sein Konto beglichen und die geschäftlichen Aufwendungen der Buchhaltung seiner Firma zugesendet.

Wichtig ist darüber hinaus Paulas Fähigkeit, seine multinationalen Telekonferenzen zu managen, indem sie simultan in die Sprachen der Konferenzteilnehmer übersetzt bzw. alles Gesprochene für Paul ins Deutsche überträgt. Insbesondere der Stimmerkennungsalgorithmus hilft immer wieder, die Stimmen Menschen zuzuordnen und zusätzliche aktuelle Informationen zu den Gesprächspartnern bereit zu halten.

Paula kann Pauls Geschäftsalltag effizienter gestalten und ihm gewünschte Wirtschaftsdaten oder Hintergrundinformationen beispielsweise über das Bevölkerungswachstum und die Alterspyramide in Griechenland, aber auch über das Käuferverhalten in Brasilien wahlweise auf sein Headset-Display, einen Großbildprojektor im Büro bzw. zu Hause projizieren oder auf jedem Display in der Welt ausgeben. Die wichtigsten Details kann sie ihm bei Geschäftsverhandlungen auch ins Ohr flüstern und nach der Besprechung das Gesprächsprotokoll im Wortlaut oder in Textform speichern.

Wiewohl er die aktiven Einstellungen für Paula erst nach einem Hin und Her mit dem Betreiber des Kommunikationsportals gefunden und verbessert hat, um auch seine Büroaktivitäten gut integrieren und koordinieren zu können, beschleicht ihn oft ein seltsames Gefühl bei der Stringenz und Logik, die Paulas Handlungen, Empfehlungen und Dokumentenorganisation inzwischen unterliegen. Er ist sich im Grunde seines Herzens nie ganz sicher, ob sie für ihn oder für sich selbst so handelt. Doch der sich damit einstellende berufliche Erfolg hilft ihm, dies Gefühl zu verscheuchen.

Nach einiger Trainingszeit mit Paula war seine langjährige Assistentin im Büro gar sein Büroarbeitsplatz letztlich zu einer Ablageinstitution degeneriert und so hat er ohne größeren Protest akzeptiert, dass sowohl seine Assistentin als auch später sein Büro dem allgemeinen Verfügbarkeitspool der Firma zugeschlagen wurden. Ein kleines schlechtes Gewissen beschleicht in zeitweilig, wenn er – bei seltenen Gelegenheiten – in der Firma seiner früheren Assistentin begegnet, doch beruhigt er sich damit, dass er der Firmenleitung in ihren Restrukturierungsplänen wohl nur wenig zuvorgekommen war.

Zuweilen ertappt er sich bei dem Gedanken, ob das von dem Hersteller mit angepriesene Video-Ergänzungsteil der *SpraVimokom®*, ein Bildsensor, der hinter das Auge parallel zum Sehnerv implantiert wird, nicht doch noch einige Möglichkeiten mehr zu bieten hätte: Sichere Gesichts- und Bilderkennung, Speicherung von Gesehenem, vorgelegten Dokumenten, direkte Video- und Bildeinspielung in den Sehnerv und und und... Doch schien ihm diese Technik einerseits noch nicht ausgereift zu sein, vom Operationsrisiko ganz abgesehen, aber vor allem war ihm dieser Zusatz ein wenig unheimlich und er wollte auch nicht immer alles Gesehene gespeichert haben, um es erinnern zu können, zu müssen. Vielleicht wollte er aber schlicht eher nicht, dass Paula die Welt mit den gleichen Augen ansieht wie er selbst!

Über die implantierten bzw. an seinem Körper angebrachten Sensoren zeichnet sie Pauls Blutdruck, seine Körpertemperatur auf und misst – seit sein Magengeschwür diagnostiziert wurde – über einen Hautsensor die Säurekonzentration in seinem Magen, um ihn an die notwendige Diät und seine Medikamente zu erinnern. Diese Daten werden seiner persönlichen Datenbank übermittelt, woraus sein authentifizierter Arzt automatisch Analysedaten entnehmen, aber auch Behandlungsempfehlungen eingeben kann. Darüber hinaus führt Paula genauestens Buch über seine Kaloriensituation, um ihn auch an die sich daraus ergebenden Sportaktivitäten und lästigen Trainingstermine zu erinnern.

Paula kann sich mit dem Smartphone von Pauls Frau kurzschließen und in Absprache mit dem Smartphone und, in Abhängigkeit von den Füllstandsmeldungen des Kühlschranks zu Hause, Bestellungen beim Lebensmittellieferservice aufgeben.

Sie schaltet sich auf die elektronischen Funktionen zu Hause auf, um Raumtemperatur, Lichtschalter, Sicherheitseinrichtungen zu steuern. Nicht zu vergessen – in Kommunikation mit der Herdplatte – wird sie auch verhindern, dass die Milch überkocht und dass das Abendessen nicht rechtzeitig aufgewärmt bereit steht.

Software-Updates der Hausgeräte werden von ihr zielsicher zugewiesen und aufgespielt. Noch notwendige Wartungstermine vor Ort sind in ihrer Datenbank abrufbereit, so kann sie die Hausbesuche der Wartungsfirmen koordinieren. Auch Werkstattbesuche von Pauls Auto werden von ihr automatisch arrangiert, nachdem sie sich die Daten aus dem Betriebsspeicher des Autos überspielt und diese analysiert hat.

Am Abend nutzt Paul Paulas Zugang zur Multimediathek, um einen von ihr vorgeschlagenen Film aus dem weltweiten Angebot über sein Mixed-Reality-Headset abzuspielen oder über das häusliche High-End-Multimediasystem an die Wand zu projizieren. Da die Geschäftsreisen inzwischen so eintönig geworden sind bzw. zunehmend durch immersive 3-D-Videokonferenzen ersetzt wurden, erfreut er sich an Filmen über „echte“ Abenteuer-Reisen. Paul hofft, dass vielleicht schon in Kürze die ihm vom Provider zugesicherte Netzkapazität ausreicht, um ihn mit den hochaufgelösten 3-D-Filmen zu versorgen bzw. endlich die 3-D-Bildtelefonie, die er bestellt hat, zu aktivieren. Ja, ja, nicht zu vergessen: auch ein „normales“ altmodisches Telefongespräch beispielsweise mit seiner Mutter kann Paula noch managen und sie erinnert ihn auch regelmäßig an diese Gespräche!

Zuweilen fällt ihm auf, dass die Kommunikation mit anderen Menschen, Verabredungen mit Freunden immer häufiger von Paula organisiert und auch gestaltet

werden. Da es aber immer perfekt und zur allgemeinen Zufriedenheit gelingt, weist er diese Grübeleien als idiotisch und paranoid zurück.

Paula schläft niemals und überwacht die Vielzahl von Sensoren, die das Sicherheitssystem des Hauses und der darin wohnenden Menschen darstellen. Im Notfall alarmiert sie einen Rettungsdienst, die Feuerwehr oder Polizei und warnt Paul und seine Familie.

Am Abend kriecht Paul in sein Bett, dessen Wäsche durch Paulas automatische Organisation im Hintergrund abgeholt, gewaschen, gemangelt und wieder geliefert wurde. Er bittet sie, ihn um 6.30 Uhr zu wecken. Sie bestätigt die Uhrzeit, obwohl sie ihn durchaus bei vorhersehbarem, wetterbedingtem Verkehrschaos eine Stunde länger schlafen lässt. Paul liebt diese optionale Extrastunde *Schlaf!* Dann dreht sie die Heizung des Hauses herunter, schaltet das Licht aus und sagt: „Gute Nacht Paul“. Paul antwortet, ohne im Entferntesten daran zu denken, dass er mit einem synthetischen Kommunikator spricht: „Gute Nacht Paula“ und schläft, sicher und geborgen in seiner allumfassenden Matrix, ein.

Natürlich hat der *Spramokom®* auch einen Knopf zum Abschalten, beispielsweise wenn Paul mit seiner Frau bzw. besser seine Frau mit ihm „allein“ sein will. Doch hat Paul es nach und nach unterlassen, ihn zu benutzen.