

Block 1 – Lernziele

In diesem Block werden Sie lernen ...

- warum die Kenntnis von mobilen Technologien und die Entwicklung Apps eine wichtige Kompetenz darstellt
- was Mobilität im Bereich der Softwareentwicklung bedeutet
- was Kontexte im Bereich der mobilen Anwendungen sind
- warum sich mobile Anwendungen an Kontexte adaptieren müssen
- aus welchen Architekturkomponenten mobile Anwendungen bestehen
- zu welchen Architekturmustern die Architekturkomponenten zusammengesetzt werden können
- und welche Anwendungstypen unterschieden werden.

1.1 Motivation (Einstiegsfragen)

Warum mobile Anwendungen entwickeln?

Gegenfrage(n):

- Wie häufig und wie lange nutzen Sie ihr Smartphone täglich?
- In welchen Situationen ist eine mobile Anwendung besonders nützlich und warum?
- Auf welche mobilen Anwendungen könnten Sie nicht verzichten, wie vielen Anwendungen nutzen Sie regelmäßig?
- ...

1.1 Motivation (Zahlen und Statistiken)

Wichtige Zahlen und Statistiken (Vgl. MindSea Meta-Bericht [12]):

- Durchschnittliche tägliche Interaktion mit dem Smartphone in 2021 liegt bei 4,8 Stunden (App Annie [1])
- Mobil genutztes Internet dominiert klassische Internetnutzung (Mary Meeker Report [2]), 2016 erstmals Trendwende mit 51.3 % (Vgl. Statcounter [16])
- 90% der mobilen Internetzeit (2021) wird innerhalb von Apps (nicht mobiler Webbrowser) verbracht (eMarketer [3])
- 2,9 Mio. Apps (2021) bzw. 4,75 Mio. App (2021) innerhalb des Google Play Store und Apple App Store (Vgl. Statista [4][5])
- Durchschnittlich werden 9 (täglich) und 30 (monatlich) mobile Anwendungen genutzt (Vgl. TechJury [6])

1.1 Motivation (Auswirkungen)

Mobile Endgeräte/Technologien

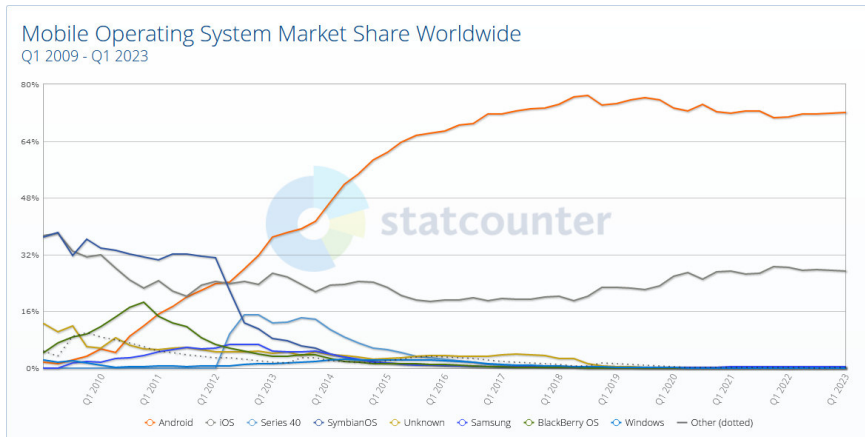
- ... lösen die klassischen Einsatzszenarios für PCs ab.
- ... lassen sich zeit- und ortsunabhängiger nutzen.
- ... erlauben Echtzeitreaktionen bzw. Echtzeitinformationssysteme.
- ... bieten ganz neue technische und wirtschaftliche Einsatzszenarien.
- ... sind Teil einer digitalen Wirklichkeit (Vernetzung von Geräten, Daten und Menschen).
- ... oftmals unverzichtbarer Teils eines Geschäftskonzepts.

Das Thema **Mobile Anwendungen** gilt als zukunftssicher! aber

- Marktverwerfungen
- Vorhersageproblematik
- Plattformheterogenität und -evolution

1.1 Motivation (Marktverwerfungen)

Beispiel (Marktverwerfungen [7]):



Marktanteile (weltweit) 'Mobile Betriebssysteme' (Quelle Statcounter)



1.1 Motivation (Vorhersageproblematik)

Seit dem Jahr 2014 gilt der Markt für mobile Betriebssysteme vorerst als konsolidiert (Android/Apple iOS).

Beispiel (Vorhersageproblematik):

Mit welchen Technologien sollen mobile Anwendungen heute entwickelt werden?

- Native Entwicklung erreicht nur eine begrenzte Anzahl von Nutzern (Plattform)
- Wie robust sind plattformunabhängige Entwicklungsansätze (Cross-Platform Ansätze), wie z. B. Flutter/Dart, React?
- Welche Entwicklungs- und Testwerkzeuge stehen jeweils bereit und wie werden diese dokumentiert u. gepflegt?



1.1 Motivation (Plattformheterogenität und -evolution)

Beispiel (Plattformheterogenität und -evolution, Vgl. Gronli et al. 2014 [10]):

Wie viele von 100 Benutzern kann einer nativen Anwendung erreichen?

- Theoretisch: Im Fall von Android 90 Benutzer (90 % Marktanteil)
- Praktische Einschränkung 'Version': max. 16.9 % der Android Benutzer (d. h. 15,21 Benutzer) bei Version 6.0 (API 23)
- Praktische Einschränkung 'Displaygröße': max. 13,2 % der Android Benutzer (d. h. 2 Benutzer) mit größerem Display
- Praktische Einschränkung 'Sensoren': ...
- usw.

Die Marktanteile sind trügerisch, da die Anforderungen beliebig kleine Fraktionen ergeben.



1.2 Mobile Technologien

Definition 1.1 (Mobile Technologien): Mit mobilen Technologien werden alle technischen Mittel zur Realisierung einer mobilen Anwendung bezeichnet d. h.

- Programmier- und (GUI) Entwicklungssprachen
- Entwicklungs-, Test- und Emulationsumgebungen
- Logische Ausführungsplattformen (Betriebssystem)
- Physische Ausführungsplattformen (Smartphones, z. B. Samsung, iPhone, etc.)
- Kommunikationstechnologien (5G Internet, WiFi, etc.)
- Backendkomponenten (Datenbanken, Services, etc.)
- ...

Diese Technologien sind selbst nicht mobil, können aber zusammengekommen die Entwicklung einer mobiler Anwendung ermöglichen.

1.3 Mobilität (Objekte und Subjekte)

Eine mobile Anwendung ergibt sich aus dem Zusammenspiel verschiedener Objekte und Subjekte (und Kontexten):

- Dienste (Objekt): Dienste können im Kontext von räumlichen Orten (Ort A oder Ort B) stehen
- Endgeräte (Objekt): Das Endgerät kann im Kontext von räumlichen Orten (Ort A oder Ort B) oder Benutzern stehen (Benutzer A oder Benutzer B)
- Benutzer (Subjekt): Benutzer können im Kontext von räumlichen Orten (Ort A oder Ort B) stehen oder im Kontext von Endgeräten (Endgerät A oder Endgerät B) stehen.

Die betroffenen Subjekte/Objekte werden oft als Mobilitätsdimension bezeichnet (Vgl. B'Far 2004 [8], Asoke et al. 2010 [17] und Pandya 2004 [14]).

1.3 Mobilität (Portabilität /Mobilität)

Definition 1.2 (Portabilität /Mobilität):

Portabilität beschreibt, dass vor und nach einer räumlichen Bewegung/Verlagerung sowie einem Benutzer- oder Gerätewechsel die Erhaltung aller Funktionen garantiert wird.

Mobilität beschreibt (oft entgegen der allgemeinen Sprachverwendung) darüber hinaus die Erhaltung aller Funktionen während der räumlichen Bewegung/Verlagerung sowie einem Benutzer- oder Gerätewechsel.

Ist ein Kontextwechsel (Ort, Benutzer, Gerät) nicht möglich, liegt weder Portabilität noch Mobilität vor (keine Definitionen).

1.3 Mobilität (Dienste)

Definition 1.3 (Dienstportabilität bzgl. Ort):

Ein Dienst kann mit einer Unterbrechung an verschiedenen räumlichen Orten (Ort A oder Ort B) genutzt werden.

Beispiel: Festnetztelefongespräch ohne Telefonanlage (Keine dynamische Umleitung des Gesprächs)

Definition 1.4 (Dienstmobilität bzgl. Ort):

Ein Dienst kann ohne Unterbrechung an verschiedenen räumlichen Orten (Ort A oder Ort B) genutzt werden.

Beispiel: Festnetztelefongespräch mit Telefonanlage (dynamische Umleitung des Gesprächs)

1.3 Mobilität (Endgeräte)

Definition 1.5 (Endgeräteportabilität bzgl. Ort/Benutzer):

Ein Endgerät kann mit einer Unterbrechung an verschiedenen räumlichen Orten (Ort A oder Ort B) genutzt werden.

Beispiel: Notebook mit WiFi-Verbindung¹ (Kurze Unterbrechung bei Wechsel des Access-Points)

Ein Endgerät kann mit einer Unterbrechung an einen anderen Benutzer (Benutzer A oder Benutzer B) übergeben werden.

Beispiel: Wechsel eines Wearable-Endgerätes

¹Ohne Fast Transition o. Ä.

1.3 Mobilität (Endgeräte - Forts.)

Definition 1.6 (Endgerätemobilität bzw. Ort/Benutzer):

Ein Endgerät kann ohne Unterbrechung an verschiedenen räumlichen Orten (Ort A oder Ort B) genutzt werden.

Beispiel: Notebook mit mobiler Internetverbindung
(Telekommunikationsinfrastruktur sorgt bei einem Funkzellenwechsel für einen unterbrechungsfreien Wechsel)

Ein Endgerät kann ohne Unterbrechung an einen anderen Benutzer (Benutzer A oder Benutzer B) übergeben werden.

Beispiel: Weitergabe eines Gesprächs (Mobiltelefon) an eine anwesende Person. Ein Benutzerwechsel schränkt i. d. R. die Endgerätefunktionalität nicht ein.

1.3 Mobilität (Benutzer)

Definition 1.7 (Benutzerportabilität bzgl. Ort/Gerät):

Der Benutzer muss bei einem Ortswechsel (Ort A oder Ort B) die Nutzung des Endgerätes oder Dienstes unterbrechen.

Beispiel: siehe Dienstportabilität (Festnetztelefongespräch)

Der Benutzer muss bei einem Gerätewechsel (Gerät A oder Gerät B) seine Dienstnutzung unterbrechen.

Beispiel: Wechsel eines Mobiltelefons aufgrund niedrigen Akkustands. Ein Gerätewechsel schränkt regelmäßig die Dienstnutzung durch den Benutzer ein.

1.3 Mobilität (Benutzer - Forts.)

Definition 1.8 (Benutzermobilität bzgl. Ort/Gerät):

Der Benutzer kann bei einem Ortswechsel (Ort A oder Ort B) sowohl das Endgerät als auch den Dienst unterbrechungsfrei nutzen.

Beispiel: siehe Endgerätemobilität (Notebook mit mobilem Internet)

Der Benutzer kann bei einem Gerätewechsel (Gerät A oder Gerät B) die Dienstnutzung unterbrechungsfrei fortsetzen.

Beispiel: Audio-Streaming an verschiedene Smart Speaker

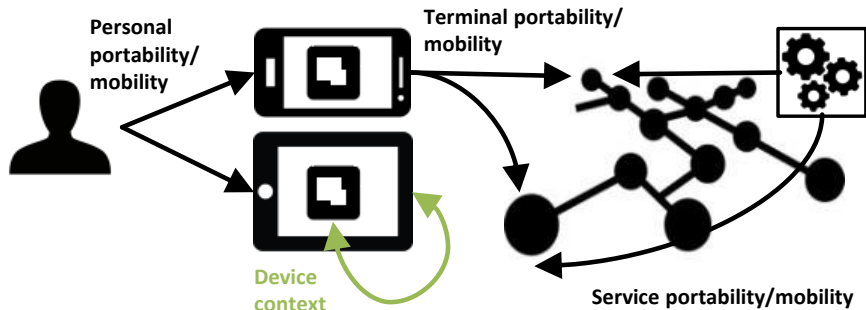
1.4 Kontexte

Die Services, Endgeräte und Benutzer sowie Orte stehen wechselseitig in eine veränderlichen Kontext, d. h. diese Kontexte sind dynamisch.

Weiterhin gibt es noch die App (Programm), welche im direkten Kontext zu dem Endgerät steht. Dieses Kontextverhältnis ist allerdings statisch, da die App – nach der Installation auf dem Endgerät – selbst nicht direkt mobil ist, d. h. das Endgerät nicht wechseln wird.

Bei der Entwicklung der App muss daher auch auf die Passfähigkeit für verschiedene Endgerätekontexte geachtet werden.

1.4 Kontexte (Zusammenfassung)



Der Kontext der mobilen Anwendung (App) ist das jeweilige Endgerät und damit **statisch** d. h. unveränderlich. Alle anderen Kontexte (Benutzer, Endgeräte, Services) können sich durch die Bewegung des Benutzers (und des mitgeführten Endgerätes) **dynamisch** verändern.

1.5 Adaptierbarkeit

Eine Herausforderung bei der Entwicklung von Apps ist der Umgang mit den dynamischen Kontextwechseln.

Einige Aspekte (z. B. Funkzellenwechsel, o. ä.) werden bereits durch die mobilen Technologien (Vgl. Definition 1.1) abgefangen.

Bedingt durch die Endgerätemobilität (oder die Forderung nach dieser), müssen die Apps sich jedoch den verschiedenen Kontexten anpassen können.

Dies bedeutet beispielsweise, dass eine App ggf. auch mit einem verbindungslosen Zustand umgehen muss.

1.6 Architekturkomponenten (Übersicht)

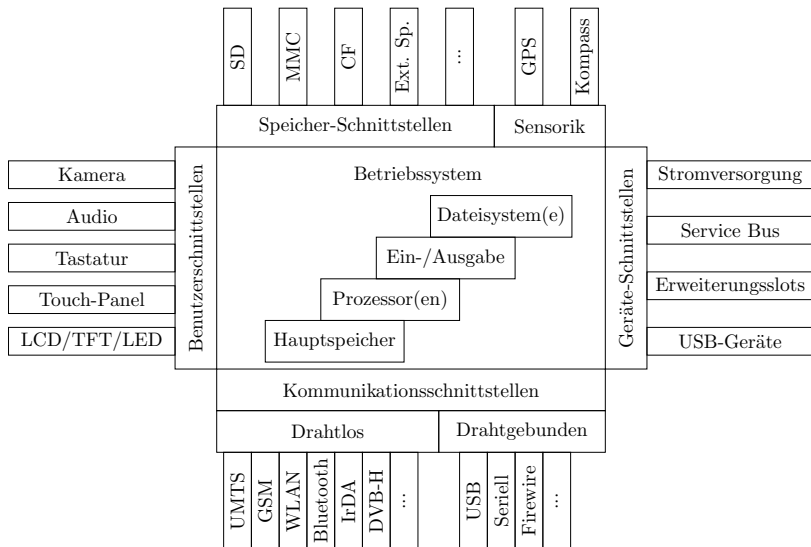
Mobile Technologien umfassen eine Vielzahl von Komponenten. Wir unterscheiden zwischen dem

Frontend: Das Frontend umfasst das mobile Endgerät (Hardware), das darauf installierte Betriebssystem sowie die genutzten mobilen Anwendungen (Rich-Client o. Client). Als Standard-Frontend wird ein internetfähiges mobiles Endgerät mit einem Webbrowser bezeichnet.

Backend: Das Backend ist ein optionaler Teil einer mobilen Anwendung. Hierbei handelt es sich um eine Serveranwendung, welche Daten (z. B. in einer Datenbank) oder Funktionen bereitstellt.

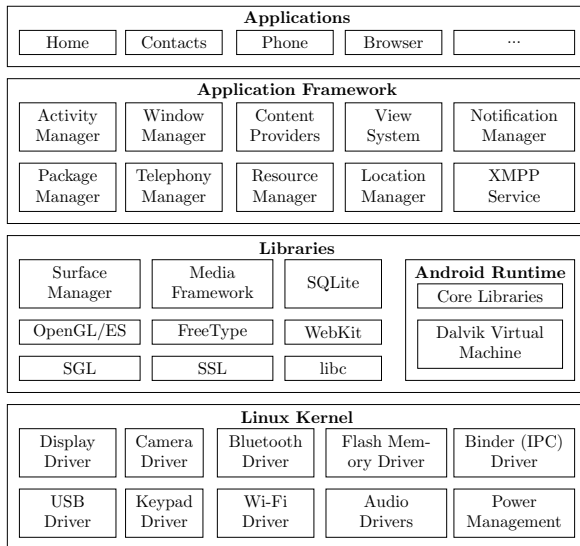
Die Aufteilung der Datenhaltung und der Funktionalität zwischen Front- und Backend hängt von dem Architekturmuster ab.

1.6 Architekturkomponenten (Frontend – Hardware)



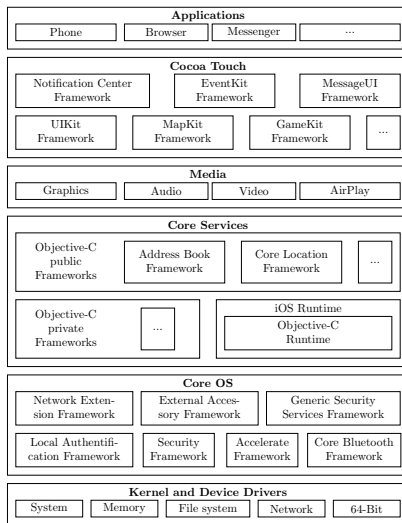
Schematischer Hardwareaufbau

1.6 Architekturkomponenten (Frontend – Android)



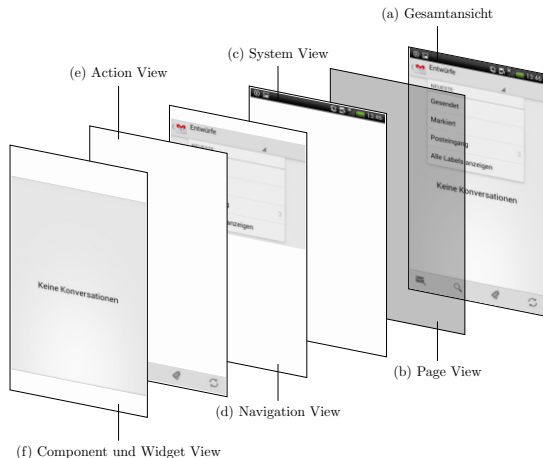
Android Software Stack

1.6 Architekturkomponenten (Frontend – iOS)



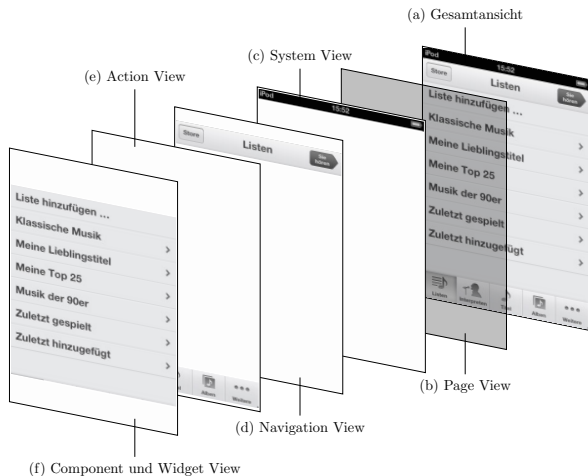
iOS Software Stack

1.6 Architekturkomponenten (Frontend – GUI Android)



Benutzerseitiger Aufbau einer Android-Anwendung
(Vgl. Hooper u. Berkman 2011 [11], Tidwell 2011 [18])

1.6 Architekturkomponenten (Frontend – GUI iOS)



Benutzerseitiger Aufbau einer iOS-Anwendung
(Vgl. Hooper u. Berkman 2011 [11], Tidwell 2011 [18])

1.6 Architekturkomponenten (Backend)

Das Backend wird in der Regel mit gängigen Webtechnologien realisiert.

Es kann sich um eine für die mobile Ansicht optimierte Webanwendung (zugänglich über ein Standard-Frontend) handeln.

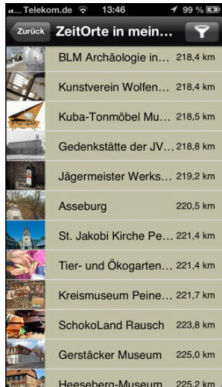
Wird das Frontend als Rich-Client umgesetzt reduziert sich der Umfang des Backends.

Das Backend bietet i.d.R. ebenfalls eine Benutzerschnittstelle zur Administration von Daten und Funktionen an.

Nicht selten erfolgt die Administration des Backend allerdings ebenfalls über mobile Anwendungen.

1.6 Architekturkomponenten (Beispiel)

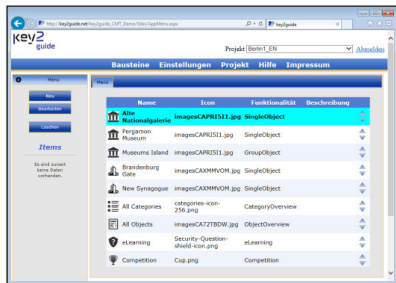
Beispiel (Natives Frontend mit Backend):



(A) key2guide (iOS)



(B) key2guide (Android)



(C) key2guide
Content Management
System (CMS)

Touristische Anwendung 'key2guide' (Fa. Advenco Consulting GmbH /
sedapta advenco GmbH)



1.7 Architekturmuster

Es existierende verschiedene konkurrierende Architekturmuster für mobile Anwendungen (Vgl. net solutions [13]).

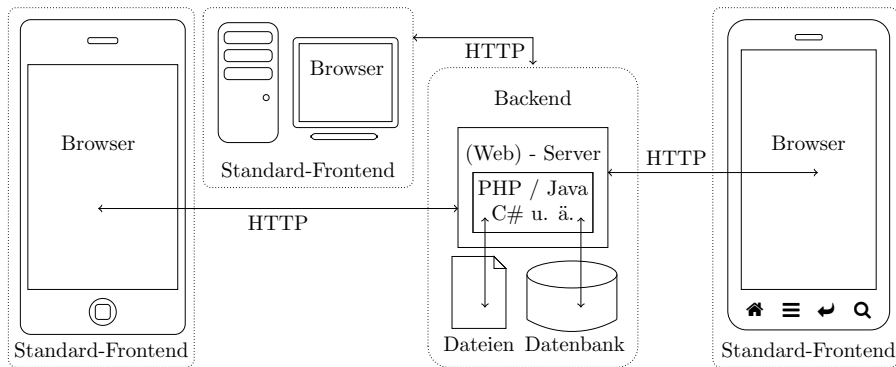
- Webanwendungen (Standard)
- (Progressive) Web Applications
- Native Anwendungen
- Hybride Anwendungen
- Cross-Plattform Anwendungen (Interpretierende Anwendungen)

Die Architekturmuster haben unterschiedlichste Eigenschaften.

Die Auswahl ist nicht trivial, da zu bestimmten Anwendungsanforderungen ggf. mehrere Architekturmuster passend sind.

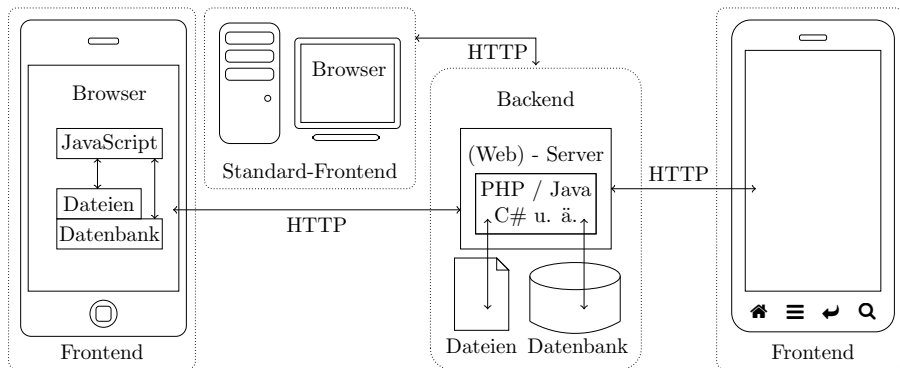
Für bestimmte Anwendungsanforderungen gibt es auch eindeutig unpassende Architekturmuster (was in der Praxis eindrucksvoll zu finden ist).

1.7 Architekturmuster (Webanwendungen)



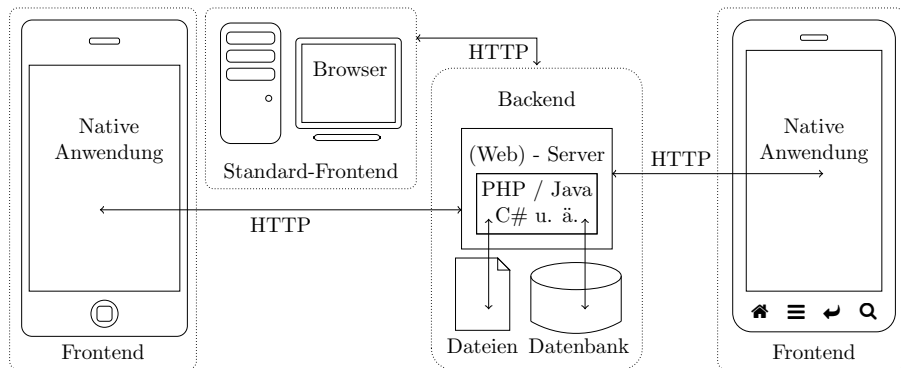
Architektur einer Webanwendung

1.7 Architekturmuster ((Progressive) Web Applications)



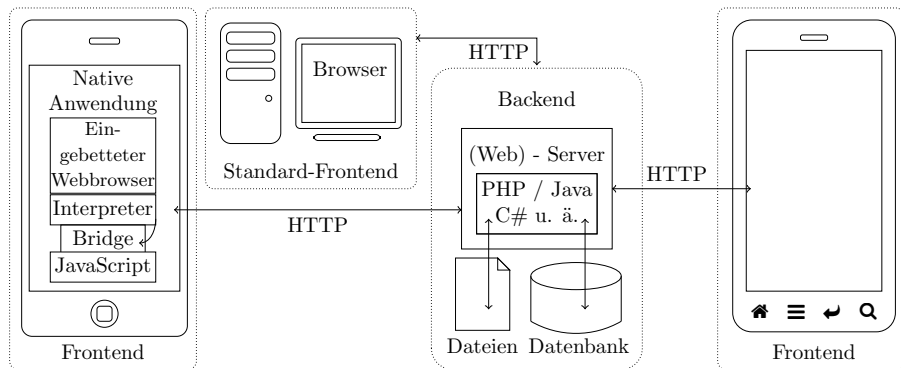
Architektur einer client-seitigen Anwendung

1.7 Architekturmuster (Native Anwendungen)



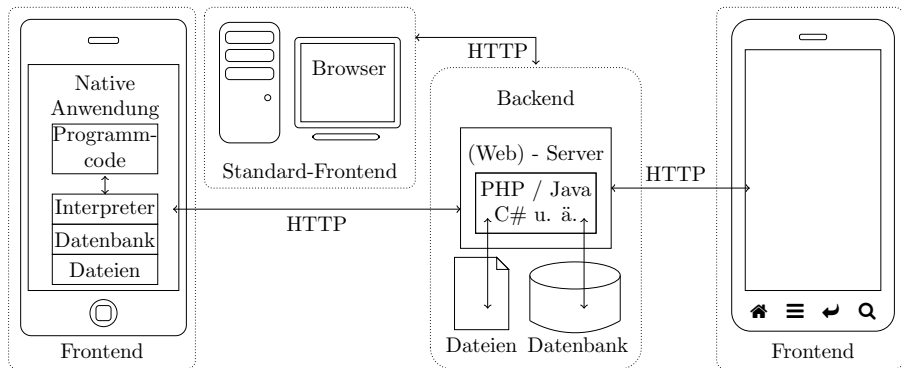
Architektur einer nativen Anwendung

1.7 Architekturmuster (Hybride Anwendung)



Architektur einer hybriden Anwendung

1.7 Architekturmuster (Cross-Plattform Anwendungen)



Architektur einer Cross-Plattform (interpretierenden) Anwendung

1.7 Architekturmuster (Auswahl)

Die Auswahl eines Architekturmusters ist eine nicht triviale Aufgabe für Softwareentwickler.

Die Auswahl eines Architekturmusters muss in einer relativ frühen Phase des Softwareentwicklungsprojekts getroffen werden.

Dabei hat die Entscheidung ggf. massive Auswirkungen auf die folgenden Phasen und kann nur schwer revidiert werden.

In der Praxis werden oft Fehlentscheidungen getroffen, da Vergleichs- und Erfahrungswerte fehlen. Die Forschung gibt einige strukturierte Entscheidungshilfen zur Architekturmusterwahl (Vgl. Bresnahan et al. 2014 [9] u. Puvvala et al. 2016 [15])

Ein Ansatz zur Auswahl von Architekturmustern ist die Einstufung der zu realisierenden Anwendung in Anwendungstypen ...

1.8 Anwendungstypen (Übersicht)

Nachdem in den letzten Jahren die Speicherkapazität und die Rechenleistung von mobilen Endgeräten drastisch zugenommen hat, können diese Faktoren bei der Auswahl eines Architekturmuster zurückgestellt werden.

Das wichtigste Kriterium ist daher die **Konnektivitätsanforderung** d. h. welche Daten werden zum Betrieb der mobilen Anwendung benötigt.

Es werden daher vier Anwendungstypen unterschieden:

- Datenunabhängige mobile Anwendungen
- Datenabhängige mobile Anwendungen
 - ▶ Datenabhängige mobile Anwendungen mit unidirektionalem Datenfluss
 - ▶ Datenabhängige mobile Anwendungen mit bidirektionalem Datenfluss
- Datenabhängige mobile Anwendungen mit Peer-to-Peer Datenfluss

Darüberhinaus lassen sich die datenabhängigen mobilen Anwendungen in Single- und Multiuser-Systeme einstufen.

1.8 Anwendungstypen (Datenunabhängige mobile Anwendungen)

Datenunabhängige Anwendungen sind Anwendungen, die zum Betrieb keine Daten aus dem Backend benötigen.

Nach Abschluss der Berechnungen werden die Daten entsprechend nicht wieder an das Backend übermittelt.

Sofern Daten verarbeitet werden, sind diese lokal entstanden und gespeichert.

Geeignete Architekturmuster: Native, hybride und Cross-Plattform Anwendungen (jeweils ohne Backend)

Beispiel (Datenunabhängige mobile Anwendungen):

Taschenrechner, ToDo-Liste, Kalender, Telefonbuch etc.



1.8 Anwendungstypen (Datenabhängige mobile Anwendungen - unidirektional)

Datenabhängige Anwendungen sind Anwendungen, die zum Betrieb Daten aus dem Backend benötigen.

Nach Abschluss der Berechnungen werden die Daten jedoch nicht wieder an das Backend übermittelt.

Sofern die Daten bearbeitet werden, werden diese ggf. lokal gespeichert.

Geeignete Architekturmuster: Native, hybride und Cross-Plattform Anwendungen (jeweils bei lokaler Speicherung der Daten), Webanwendung (ohne lokale Speicherung)

Beispiel (Datenabhängige mobile Anwendungen - unidirektional):
Bahn- oder Flugauskunft, Wetterauskunft, Newsreader etc. ☐

Diese Anwendungstypen werden auch als **Informationssysteme** bezeichnet.

1.8 Anwendungstypen (Datenabhängige mobile Anwendungen - bidirektional)

Datenabhängige Anwendungen sind Anwendungen, die zum Betrieb Daten aus dem Backend benötigen.

Nach Abschluss der Berechnungen werden die Daten wieder an das Backend übermittelt d. h. die Daten werden bearbeitet.

Die Daten werden ggf. lokal gespeichert.

Geeignete Architekturmuster: Native, hybride und Cross-Plattform Anwendungen (jeweils bei lokaler Speicherung der Daten), Webanwendung (ohne lokale Speicherung)

Beispiel (Datenabhängige mobile Anwendungen - bidirektional):
Bezahlenwendungen, E-Commerce (Shopsysteme, Auktionen),
Reservierung (Reisen, Autovermietung), etc. ☐

Diese Anwendungstypen werden auch als **Transaktionssysteme** bezeichnet.

1.8 Anwendungstypen (Datenabhängige mobile Anwendungen - Peer-to-Peer)

Datenabhängige Anwendungen in eine Peer-to-Peer Netzwerk tauschen Daten bidirektional mit einem oder mehreren Kommunikationspartnern aus.

Ein Backend wird aufgrund der Kommunikationstopologie (Peer-to-Peer) nicht verwendet.

Die Daten werden ggf. lokal gespeichert.

Geeignete Architekturmuster: Native, hybride und Cross-Plattform Anwendungen (jeweils ohne Backend)

Beispiel (Datenabhängige mobile Anwendungen - Peer-to-Peer):
Social software, Volp, FaceTime, Skype, etc. ☐

Diese Anwendungstypen werden auch als **Kommunikationssysteme** bezeichnet.

1.9 Fokus

Der **Fokus** der Veranstaltung liegt auf nativen mobilen Anwendungen und der Android Plattform (inkl. aller praktischen Übungen).

Schwerpunkt: Entwicklung einer nativen mobilen Anwendung in Form einer Projektarbeit

Behandelte Aspekte: Entwicklungsumgebung, Softwarearchitektur von mobilen Anwendungen (zentrale Klassen, Lebenszyklus, etc.), Layout, Datenspeicherung, Datenaustausch, Integration von Smartphone Funktionen (z.B. Kamera), Auslesen von Sensoren, ...

Nebenaspekte der Veranstaltung werden auf Konzepten wie z. B. dieser Einführung und dem technologieunabhängigen Entwurf und dem Testen von mobilen Anwendungen liegen.

Block 1 – Zusammenfassung

- Mobile Technologien sind ein zukunftsweisendes Arbeitsgebiet
- **Mobilität** bezieht sich auf Endgeräte, Dienste und Benutzer
- Mobile Anwendungen bestehen aus vielfältigen **Architekturkomponenten**
- Die Architekturkomponenten lassen sich zu verschiedenen **Architekturmustern** komponieren
- Die Architekturmuster sind für bestimmte **Anwendungstypen** geeignet

Block 1 – Weitere Aufgaben

- Wählen Sie eine oder mehrere mobilen Anwendungen aus ihrem täglichen Gebrauch aus und erklären Sie, ob diese portabel o. mobil bezüglich des Benutzers, des Endgerätes und des Services sind.
- Finden (ggf. durch eine (Web)-Recherche) Beispiele von konkreten mobilen Anwendungen, welchen den vorgestellten Architekturmustern jeweils entsprechenden.
- Finden Sie Vor- und Nachteile für die jeweiligen Architekturmuster.
- Geben Sie jeweils ein weiteres Beispiel für ein Information-, Transaktions- und Kommunikationssystem.
- Warum wird die Dienstportabilität/Dienstmobilität nicht direkt bzgl. des Benutzers betrachtet?

Block 1 – Literatur I

- [1] URL: <https://www.appannie.com/en/insights/market-data/state-of-mobile-2022/>.
- [2] URL: <https://www.bondcap.com/report/itr19/>.
- [3] URL: <https://www.emarketer.com/content/us-time-spent-with-media-2021-update>.
- [4] URL: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>.
- [5] URL: <https://www.statista.com/statistics/268251/number-of-apps-in-the-itunes-app-store-since-2008/>.
- [6] URL: <https://techjury.net/blog/app-usage-statistics/>.
- [7] URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide/\#quarterly-200901-202301>.

Block 1 – Literatur II

- [8] Reza B'far. *Mobile computing principles: designing and developing mobile applications with UML and XML*. Cambridge UK: Cambridge University Press, 2004.
- [9] Timothy Bresnahan, Joe Orsini und Pai-Ling Yin. „Platform Choice by Mobile App Developers“. In: *NBER Working Paper* (2014).
- [10] Tor-Morten Gronli u. a. „Mobile application platform heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS“. In: *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*. IEEE, 2014, S. 635–641.
- [11] Steven Hoober und Eric Berkman. *Designing Mobile Interfaces - Patterns for Interaction Design*. O'Reilly, 2011, S. I–XXXIII, 1–545. ISBN: 978-1-449-39463-9.
- [12] MindSea Team. *25 Mobile App Statistics To Know In 2022*. <https://mindsea.com/app-stats/>. 2022.

Block 1 – Literatur III

- [13] net solutions. *Mobile AppDevelopment: A Comprehensive Guide*.
<https://www.netsolutions.com/hub/mobile-app-development>. 2023.
- [14] Ray Pandya. *Mobile and Personal Communication Systems and Services*. New York City (NY) USA: Wiley, 2004.
- [15] Abhinay Puvvala u. a. „Mobile Application Developers’ Platform Choice Model“. In: *49th Hawaii International Conference on System Sciences, HICSS 2016, Koloa, HI, USA, January 5-8, 2016*. Hrsg. von Tung X. Bui und Ralph H. Sprague Jr. IEEE Computer Society, 2016, S. 5721–5730.
- [16] StatCounter - Free Invisible Web Tracker, Hit Counter and Web Stats. *Desktop, mobile, and tablet comparison*.
<http://gs.statcounter.com/#desktop+mobile+tablet-comparison-ww-monthly-201610-201610-bar>. 2016.

Block 1 – Literatur IV

- [17] A.K. Talukdar. *Mobile Computing, 2E*. New York City (NY) USA: McGraw-Hill Inc., 2010.
- [18] Jenifer Tidwell. *Designing interfaces - patterns for effective interaction design (2. ed.)* O'Reilly, 2011, S. I–XXV, 1–547. ISBN: 978-1-449-37970-4.