

# Software Architektur - Service Oriented Architecture



Modul Systemarchitektur - Software Architektur  
16. & 23.11.2015  
Daniel Liebhart

# **Software Architektur – Service Oriented Architecture**

Daniel Liebhart, 16. & 23.11.2015

Verfasser: Daniel Liebhart  
Lektorat: Bruno Holliger (2. Auflage), Jürg Fuchs (3. Auflage)  
4. Auflage: 2014 (Neugestaltung nach D. Liebhart: SOA goes  
Real, Hanser 2007)  
Version 6.0

© by Daniel Liebhart

# Inhalt

<b>Referenzen und Abkürzungen .....</b>	<b>3</b>
<b>SOA Referenzmodell.....</b>	<b>6</b>
1.1 Einleitung .....	6
1.2 Der Service als Grundkomponente .....	6
1.2.1 Dienste statt Applikationen.....	7
1.2.2 Service Layering.....	8
1.3 Architektur .....	9
1.3.1 Eigenschaften des SOA-Modells.....	10
1.4 Referenzmodelle .....	10
1.5 Das SOA Referenzmodell des W3C.....	11
1.5.1 Bedeutung.....	12
1.6 Referenzmodelle für generische Unternehmen.....	13
1.6.1 GereriCo.....	13
1.6.2 Real-Time Enterprise .....	14
1.6.3 Bedeutung.....	16
1.7 Beispiel: IT Supply Chain dank SOA.....	16
<b>Grundlegende SOA Modelle der grossen Hersteller.....</b>	<b>18</b>
2.1 Einleitung .....	18
2.2 IBM.....	18
2.2.1 IBM SOA Foundation .....	19
2.2.2 On Demand Operating Environment .....	20
2.2.3 Bedeutung.....	21
2.3 SAP .....	21
2.3.1 Enterprise Service Architecture (ESA) .....	21
2.3.2 SAP NetWeaver .....	23
2.3.3 Die SAP XI Exchange-Plattform (heute SAP PI) .....	24
2.3.4 Bedeutung.....	24
2.4 Oracle.....	24
2.4.1 Oracle Business Integration .....	25
2.4.2 Oracle Fusion.....	25

2.4.3	Die SOA Suite .....	27
2.4.4	Bedeutung .....	27
2.5	Microsoft.....	27
2.5.1	Die Service-Architektur der Application Platform.....	28
2.5.2	Application Architecture.....	29
2.5.3	SOA-Grundsätze, Praktiken und Frameworks .....	29
2.5.4	Bedeutung .....	30
<b>SOA Komponenten .....</b>		<b>31</b>
3.1	Einleitung .....	31
3.2	SOA Komponenten auf einen Blick .....	32
3.3	Presentation Ebene.....	33
3.3.1	Portale.....	33
3.3.1.1	Bedeutung .....	34
3.3.2	Office Business Applications .....	34
3.3.2.1	Bedeutung .....	35
3.3.3	Client Applications .....	36
3.3.3.1	Bedeutung .....	37
3.4	Orchestration Ebene .....	37
3.4.1	Business Process Management .....	37
3.4.1.1	Bedeutung .....	38
3.4.2	Ausführbare Prozesse.....	38
3.4.2.1	Bedeutung .....	39
3.4.3	Geschäftsregeln .....	39
3.4.3.1	Bedeutung .....	40
3.5	Service Ebene.....	40
3.5.1	Service Management .....	41
3.5.1.1	Bedeutung .....	41
3.5.2	Service Interfaces.....	42
3.5.2.1	Bedeutung .....	42
3.5.3	Spezialisierte Services .....	43
3.5.3.1	Bedeutung .....	44
3.6	Integration Architecture Ebene .....	44
3.6.1	Logische Integration .....	44
3.6.1.1	Bedeutung .....	45
3.6.2	Enterprise Service Bus.....	45
3.6.2.1	Bedeutung .....	45
3.6.3	Data Integration.....	45
3.6.3.1	Bedeutung .....	45

# Referenzen und Abkürzungen

## Referenzen

[Allen, Bartkus 2007]	Allen, C., Markus, K.: HR-XML Consortium Library, 2007 April 15
[Alt, Österle 2003]	Alt, R., Österle, H.: Real-Time Business. Lösungen, Bausteine und Potentiale des Business Networking, Springer, Berlin; Auflage: korr. Nachdr., Oktober 2003
[Banerjee 2007]	Banerjee, A.: Building Office Business Applications, Microsoft Architect Journal, January 2007
[Beck 2005]	Beck, H.: Einsatz von Rule-Engines zur flexiblen Wissensverarbeitung im betrieblichen Umfeld, Fachhochschule Darmstadt, Fachbereich Informatik 18. Oktober 2005
[Booth et Al. 2004]	Booth, D., Liu, C.K.: Web Service Description Language (WSDL) Version 2.0 Part 0: Primer, W3C Candidate Recommendation, Mar. 2006
[Crawford et Al. 2005]	Crawford, C.H., Bate, G.P., Cherbakov, L., Holley, K., Tsocanos, C.: Toward an on demand service-oriented architecture, IBM Systems Journal, Vol 44, No 1, 2005
[Damodaran 2004]	Damodaran, S.: B2B integration over the Internet with XML: RosettaNet successes and challenges, International World Wide Web Conference ACM 2004
[Demarest, Heller 2006]	Demarest, G., Heller, P.: Oracle Fusion Architecture and Oracle Fusion Applications, Oracle Technical White Paper, January 2006
[Fritz 2004]	Fritz, F.J.: When Does a Web Service Become an Enterprise Service?, SAP insider, Apr – May – Jun 2004
[Hazra 2002]	Hazra, T.K.: Building Enterprise Portals: Principles to Practice, ACM International Conference on Software Engineering, May 19-25 2002, Orlando, Florida
[High et Al. 2005]	High, R., Kinder, S., Graham, S.: IBM's SOA Foundation – An Architectural Introduction and Overview, IBM November 2005
[Jacobs, Walsh 2004]	Jacobs, I., Walsh, N.: Architecture of the World Wide Web, Volume One, W3C Recommendation, 15 December 2004
[Krafzig et Al. 2005]	Krafzig, D., Banke, K., Slama, D.: Enterprise SOA. Service Oriented Architecture Best Practices, Prentice Hall International, 2005
[Kurian 2006]	Kurian, T.: Oracle's SOA Strategy, Oracle Corporation, April 2006
[Maloney 2004]	Maloney, D.: Enterprise Envisioning Session, SAP AG, 2004
[Mellor et Al. 2003]	Mellor, S.J., Clark, A.N., Futagami, T.: Model-Driven Development, IEEE Software, September/October 2003
[Nussdorfer, Martin, 2003]	Nussdorfer, R., Martin, W.: RTE – Real-Time-orientierte IT-Architektur, EAI Forum, 2003
[Oracle 2006]	Oracle Corporation: Oracle Business Integration, Oracle Corporation, 2006
[Papazoglou, Georgakopoulos 2003]	Papazoglou, M.P., Georgakopoulos, D.: Service-Oriented Computing, Communications of the ACM, October 2003 / Vol. 46, No. 10
[Pellengahr 2005]	Pellengahr, M.: GeneriCo – Best-Practices und Design-Guidelines für SOA-Anwendungen in Form der sog. SOA-Blueprints, Westfälische Wilhelms-Universität Münster, 11.7.2005
[Platt 2002]	Platt, M.: Service Oriented Architecture, Microsoft Platform Strategy Group, Microsoft Corporation 2002
[Rosenberg, Dustdar 2005]	Rosenberg, F., Dustdar, S.: Business Rules Integration in BPEL – A Service-Oriented Approach, Vienna University of Technology, 10. Feb 2005

[Salminen 1995]	Salminen, A.: EDIFACT for business computers: has it succeeded?, ACM StandardView Volume 3 , Issue 1, March 1995
[Santos 2006]	Santos, J.: The EFX Architectural-Guidance Software Factory, Microsoft Corporation, October 2006
[Scheer et Al., 2003]	Scheer, A.-W., Abolhassan, F., Bosch, W.: Real-Time Enterprise, Springer- Verlag, Berlin, 2003
[Schmale 2004]	Schmale, T.: Mit EAI und SOA zum Real-Time Enterprise, Buchbeitrag zu "EAI – Serviceorientierung und nachhaltige Architekturen", inubit AG, 2004
[Schumann 2007]	Schumann, R.: Enterprise SOA, SAP AG, 2007
[Seymour 2005]	Seymour, M.: Enabling Adaptive Business Processes: Oracle E-Business Suite and Service-Oriented Architecture, Oracle White Paper, August 2005
[Sondermann 2007]	Sondermann, K.: BPM & SOA oder Wertschöpfung durch serviceorientierte IT-Strategie, SOA Initiative 2007, Frankfurt, 25. Januar 2007
[Sprott, Wilkes 2004]	Sprott, D., Wilkes, L.: Understanding Service-Oriented Architecture, Microsoft Architects Journal, January 2004
[Stevens 2003]	Stevens, H.: The HL7 3.0 Introduction, Canadian Institute for Health Information, 2003
[Stotz 2003]	Stotz, M.: SAP NetWeaver's Composite Application Framework, SAP AG, 2003
[Uhr 2004]	Uhr, W.: Computer Integrated Business, Vorlesungsskript, Technische Universität Dresden, Lehrstuhl Wirtschaftsinformatik, 2004
[Weitzel, Martin 2002]	Weitzer. T., Martin, S.V.: XML-Standards für ein Straight Through Processing im Wertpapiergeschäft, Frankfurt/Main : Johann Wolfgang Goethe University. 2002
[Wilkes, Harby 2004]	Wilkes, S., Harby, J.: SOA Blueprints Initiative Definition, Draft v0.5, June 2004
[Allen, Bartkus 2007]	Allen, C., Markus, K.: HR-XML Consortium Library, 2007 April 15

## Abkürzungen

ABAP	Advanced Business Application Programming
ADS	Active Directory System
ASCII	American Standard Code for Information Interchange
ASP	Active Server Pages
BAM	Business Activity Monitoring
BAPI	Business Application Programming Interface
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modelling Notation
CAF	Composite Application Framework
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
DNS	Domain Name Service
EDA	Event Driven Architecture
EDIFACT	United Nations Electronic Data Interchange For Administration, Commerce and Transport
EII	Enterprise Information Integration
EIS	Enterprise Integration System
EIS	Enterprise Integration Bus
ELT	Extract Load Transfer
ERP	Enterprise Resource Management
ESA	Enterprise Service Architecture
ESB	Enterprise Service Bus
FI / CO	Finance / Controlling (SAP Modul)
HL7	Health Level Seven
HR-XML	Human Resources eXtensible Markup Language
HTTP	Hypertext Transfer Protocol
IAM	Identity and Access Management
ISP	Internet Service Provider
ITIL	IT Infrastructure Library

ITSM	IT Service Management
J2EE	Java 2 Enterprise Architecture
LDAP	Lightweight Directory Access Protocol
MDSD	Modell Driven Software Development
OBA	Office Business Applications
OID	Oracle Internet Directory
PKI	Publik Key Infrastructure
PLM	Product Lifecycle Management
QoS	Quality of Service
RFC	Request for Comment
RTE	Real Time Enterprise
SaaS	Software as a Service
SAP	Systeme Anwendungen Programme
SCM	Supply Chain Management
SFA	Sales Force Automation
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing
SRM	Supply Relationship Management
SWIFT	Society for Worldwide Interbank Financial Transactions
TCP/IP	Transfer Control Protocol / Internet Protocol
UDDI	Universal Description, Discovery and Integration
URI	Uniform Resource Identification
W3C	World Wide Web Consortium
WFMC	Workflow Management Coalition
WSDL	Web Service Description Language
WS-RM	Webservice Remote
WSRP	Web Services for Remote Portlets
xCOM	Cost and Quotation Management
xIEP	Integrated Exploration and Production
xMII	Manufacturing Integration and Intelligence
xPD	Product Definition
xRPM	Resource and Portfolio Management
XSLT	Extensible Stylesheet Language Transformations

# SOA Referenzmodel

## 1.1 Einleitung

Service Oriented Architecture (SOA) steht für eine ganzheitliche Betrachtung einer IT-Systemlandschaft als Unterstützungsfunktion für betriebliche Prozesse. Funktionen, die durch einzelne Systeme abgedeckt werden, sind dank SOA in einer standardisierten Form unternehmensweit zugänglich. Diese Funktionen sind sogar über Unternehmensgrenzen hinweg zugänglich.

SOA ist eine Standardarchitektur, die eine logische Teilung zwischen Applikationen, Integrationsmechanismen, Diensten und der Orchestrierung vorsieht. Die Applikationen beinhalten bestehende oder auch neue Systeme und logische Datenspeicher während die Dienste Schnittstellen zu den einzelnen Anwendungen oder funktionalen Bereichen darstellen. Zur Steuerung von Abläufen unter Einbeziehung mehrerer Dienste dient die Orchestrierung. Die Kommunikation zwischen verschiedenen Diensten und die Kommunikation zwischen Diensten und deren Implementierung (Applikation und Daten) erfolgt über eine logische Integrationsarchitektur.

Der Bau eines konkreten Systems, also die Lösung eines anstehenden Problems, ist der Ausgangspunkt für die Auswahl der Komponenten einer SOA. Nur wenn eine Komponente in einer konkreten Situation als nützliches Instrument eingesetzt werden kann und gleichzeitig einen wichtigen Teil der Gesamtarchitektur darstellt, lohnt sich der Aufwand für die Auswahl.

Ausgehend vom pragmatischen und herstellerunabhängigen SOA-Modell, können auf jeder Ebene der Architektur die für ein Unternehmen wichtigen Bestandteile isoliert werden. Die wichtigen Komponenten einer SOA sind auf den Ebenen Presentation, Orchestration, Services und Integration Architecture zu finden.

## 1.2 Der Service als Grundkomponente

Das wichtigste Element einer SOA ist der Service als standardisierte Darstellung von Funktionalität. Service Oriented Computing (SOC) ist ein Paradigma, das Services (Dienste) als fundamentales Element für die Erstellung von Applikationen verwendet. Dieses fundamentale Element besteht aus einem Basisdienst, seiner Beschreibung sowie einer Reihe von Basisoperationen (Publication, Discovery, Selection und Binding) [Papazoglou, Georgakopoulos 2003].

Ein Service ist eine sich selbst beschreibende, offene Komponente, die eine schnelle und kostengünstige Zusammenstellung von verteilten Applikationen ermöglicht. Services werden durch so genannte Service Provider bereitgestellt. Service Provider sind Organisationen, die eine Service Implementation bereitstellen, die Service-Beschreibung publizieren und den technischen und kaufmännischen Support für einen Service zur Verfügung stellen. Dienste sind Software-Module, die über ihren Namen via Schnittstelle aufgerufen werden, typischerweise in einem Anfrage-Antwort(Request/Reply)-Modus. Ein Servicenehmer (Service Consumer) ist eine Software, die einen Dienst in Form eines Interface Proxy intern darstellt.



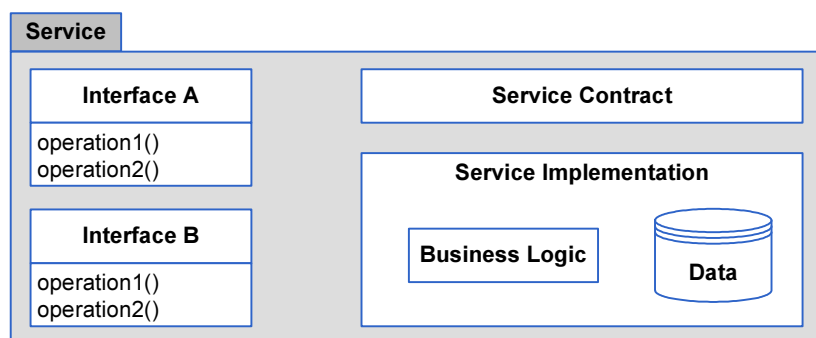


Abbildung 1: Bestandteile eines Service

Damit ein Dienst überhaupt als SOA Service zugänglich gemacht werden kann, muss er eine Reihe von Bedingungen erfüllen. Er muss aufruf-bar sein, eine definierte Funktionalität aufweisen und definierte Rahmenbedingungen ein-halten. Jeder Dienst besteht aus mindestens drei Komponenten: der Schnittstelle, dem „Service Contract“ und der „Service Implementation“ [Krafzig et Al. 2005].

- **Service:** Der Service selbst muss einen Namen oder, falls er unternehmensweit zugänglich gemacht werden soll, sogar einen eindeutigen Namen aufweisen.
- **Service Interface(s):** Schnittstellen des Dienstes, die den Zugriffspunkt darstellen (ein und derselbe Dienst kann dabei verschiedene Schnittstellen aufweisen).
- **Service Contract:** Der Service Contract ist eine informelle Spezifikation der Verantwortlichkeit, der Funktionalität, der Bedingungen und Einschränkungen sowie der Verwendung des Service.
- **Service Implementation:** Die technische Realisierung des Service, also die Umsetzung der Business-Logik sowie die persistente Haltung eventuell notwendiger Daten sind die wichtigsten Bestandteile.

### 1.2.1 Dienste statt Applikationen

Die Grundidee hinter „Dienste statt Applikationen“ ist die Wiederverwendung ganzer Systeme und die Kombination bestehender Systeme zu einem funktional erweiterten neuen Gesamtsystem.

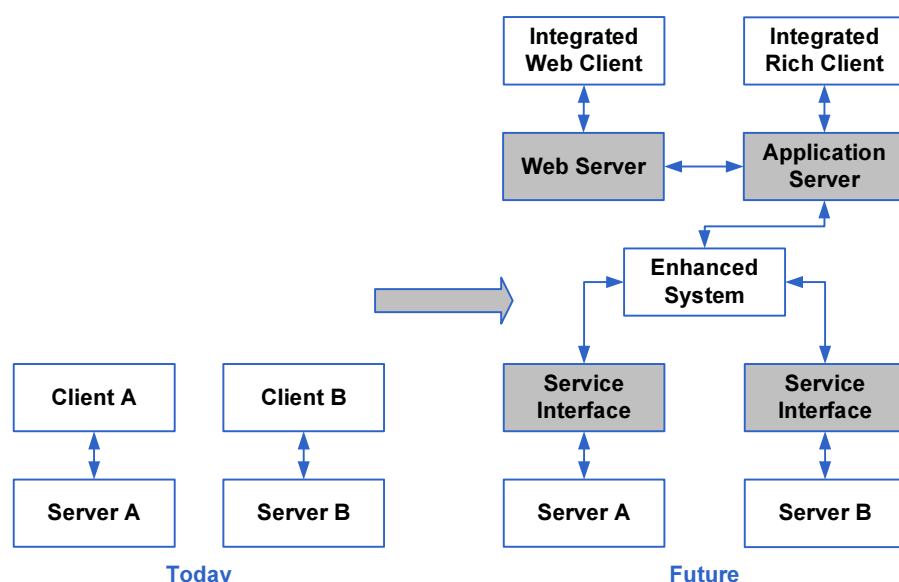


Abbildung 2: Wiederverwendung auf hohem Niveau

Wiederverwendung auf hohem Niveau wird durch eine Kapselung ganzer Systeme durch definierte Service-Schnittstellen erreicht. Bestehende Client-Server-, Host- oder Legacy-Systeme werden also nicht durch eine neue Applikation ersetzt, sondern die Serverkomponente wird um ein Interface erweitert, das einen unternehmensweit zugänglichen Dienst definiert. So können bestehende Systeme zu erweiterten Gesamtlösungen kombiniert werden.

### 1.2.2 Service Layering

Die Gruppierung einzelner Services zu einem funktionierenden Ganzen erfolgt über eine logische Gliederung der unterstützenden Funktionen in Schichten. Dabei werden die unterstützenden Funktionen ebenfalls als einzelne Services definiert. So ist beispielsweise ein SLA (Service Level Agreement) eines Service wiederum ein Dienst, der vom betroffenen Service benutzt werden kann. In diesem Schichtenmodell unterscheidet man nicht zwischen technischen und organisatorischen Diensten. Eine IT-Organisation kann dank dieser Betrachtungsweise sehr nahe am Service Layering umgesetzt werden. Dabei wird der Leistungsauftrag einzelner Abteilungen, Teams oder Personen direkt als Service formuliert. So kann beispielsweise ein dediziertes Team für das Service Monitoring oder für QoS (Quality of Service) zuständig sein. Diese Aufteilung entspricht dem Aufbau einer Informatik-Organisation als interner oder auch externer Dienstleister. Die Messung und eventuell auch die Verrechnung der Leistung kann direkt auf die „Service Usage“ und die Service-Qualität und Verfügbarkeit bezogen werden.

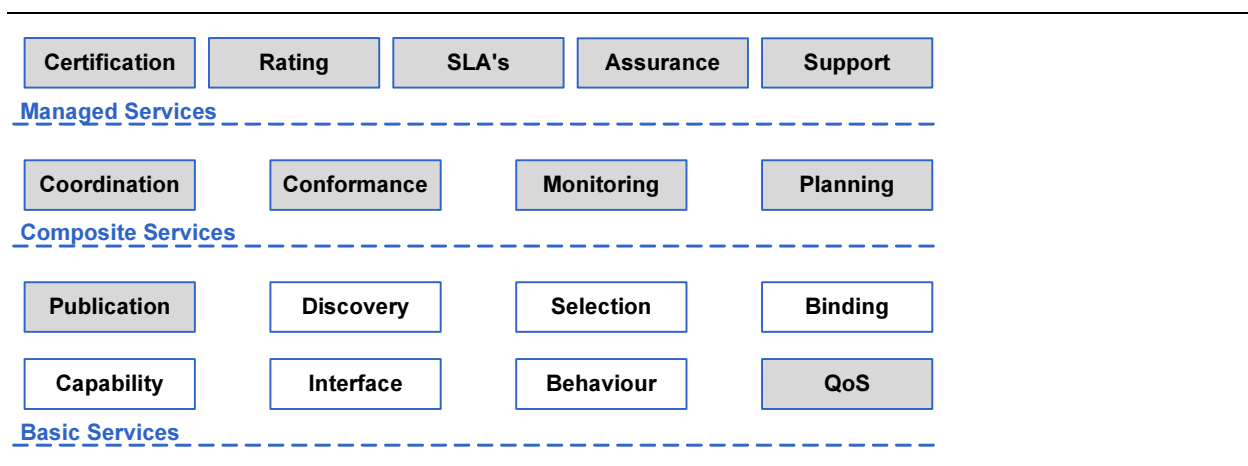


Abbildung 3: Service Layering für organisatorische und technische Dienste

Die unterstützenden, organisatorischen Aspekte umfassenden und von der IT-Organisation eines Unternehmens bereitzustellenden Services können in den Bereich Basic Services, Composite Services und Managed Services unterteilt werden.

- **Basic Services:** Jeder Service verwendet seinerseits wiederum diese Basisdienste, um die notwendigen Grundmechanismen, die für eine Einbindung des Dienstes in eine verteilte Applikation notwendig sind, bereitzustellen. Die Publication, also das „unternehmensweite Sichtbarmachen“ von Diensten in einem Servicekatalog reicht nicht aus. Damit bekannt wird, dass ein Dienst existiert, muss er kommuniziert, die Verantwortlichkeit geregelt und beworben werden. Die Quality of Service eines Dienstes hängt stark von der organisatorischen Unterstützung der technischen Plattform ab.
- **Composite Services:** Diese Dienste sind für die Bereitstellung eines einzelnen zusammen-gesetzten Service notwendig. Jeder einzelne Dienst dieser Schicht umfasst organisatorische Aufgaben. Insbesondere ist die Planung von Diensten ein wichtiger Einflussfaktor für die effektive Verwendung auf Unternehmensebene. Nur wenn Services zeitgerecht bereitgestellt werden, lassen sich diese von diversen Applikationen einsetzen. Die Konformitätsprüfung (Conformance), Koordination und das Monitoring von Diensten sind weitere Aufgaben mit ausgeprägten organisatorischen Aspekten.
- **Managed Services:** Insbesondere die für den geregelten Betrieb notwendigen Dienste (Assurance und Support) sind zentrale „Managed Services“. Die Zertifizierung und Bewertung von Diensten sollen bei der Auswahl von Services helfen. Jeder Dienst ist mit einem definierten SLA (Service Level Agreement) zu versehen, der einem Servicenehmer vertraglich eine garantierte Dienstverfügbarkeit zusichert. Diese Schicht umfasst also fast nur organisatorische Aufgaben.

### 1.3 Architektur

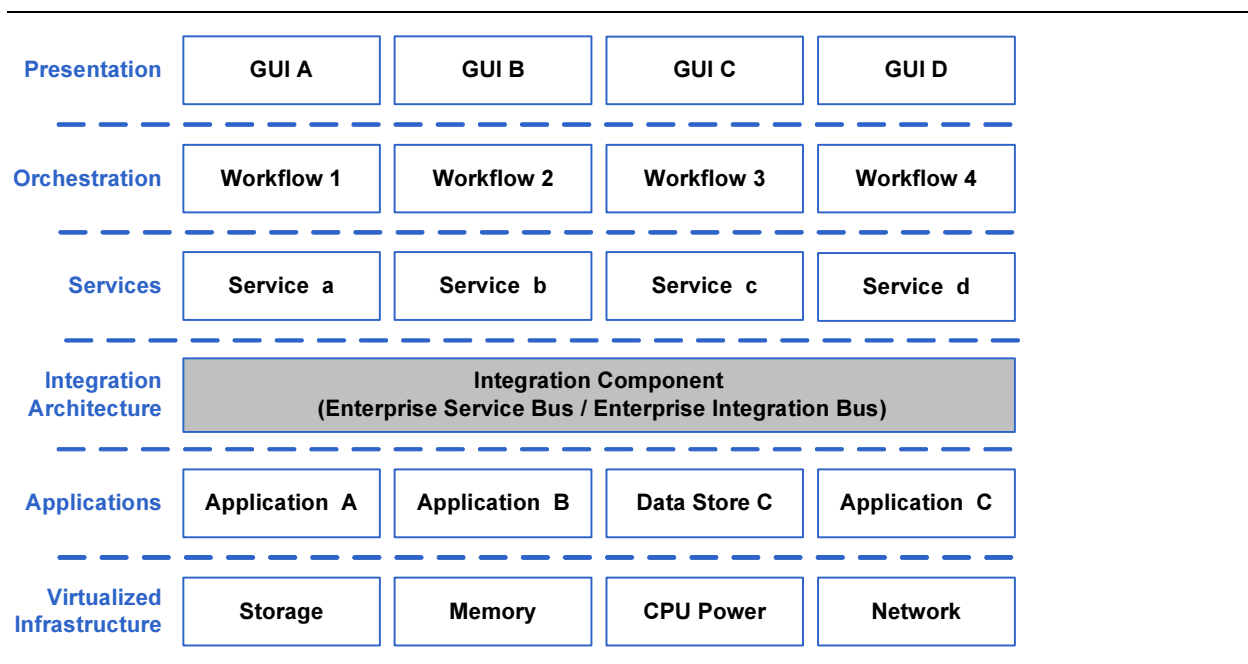


Abbildung 4: Die Architektur des SOA-Modells

Das SOA-Modell umfasst folgende Ebenen:

- **Presentation:** Das User Interface einer bestimmten Applikation.
- **Orchestration:** Der Ablauf einer Applikation wird als Workflow festgelegt. Diese Abläufe bestehen aus einer Sequenz von aufgerufenen Services, die gleichzeitig den Datenfluss zwischen diesen Services steuert.
- **Services:** In dieser Ebene sind sämtliche Dienste enthalten.
- **Integration Architecture:** Die Infrastruktur zur Verbindung der verschiedenen Dienste und zur Verbindung von Diensten mit bestehenden Applikationen/Systemen oder Datenbanken.
- **Applications:** Diese Ebene besteht aus den neuen oder bereits vorhandenen Anwendungen – respektive deren funktionalen Blöcken – sowie aus Datenbanken oder anderen Datenquellen eines Unternehmens.
- **Virtualized Infrastructure:** Diese Ebene umfasst sämtliche zum Betrieb aller Komponenten des SOA Modells notwendigen Ressourcen.

Das SOA-Modell versteht SOA als ganzheitliche Betrachtung einer IT-Systemlandschaft, die als Unterstützungsfunktion für betriebliche Prozesse realisiert wird. Funktionen, die durch einzelne Systeme abgedeckt werden, sind dank SOA in einer standardisierten Form unternehmensweit zugänglich. Diese Funktionen sind sogar über Unternehmensgrenzen hinweg zugänglich.

Das SOA-Modell sieht eine logische Teilung zwischen Applikationen, Integrationsmechanismen, Diensten und der Orchestrierung vor. Die Applikationen (Applications) beinhalten bestehende oder auch neue Systeme und logische Datenspeicher. Die Dienste (Services) stellen Schnittstellen zu den einzelnen Anwendungen dar. Die Orchestrierung (Orchestration) dient zur Steuerung von Abläufen, die mittels Einbezug mehrerer Dienste durchgeführt werden. Die Kommunikation zwischen Diensten untereinander und deren applikatorische Umsetzung (Applikation und Daten) erfolgt über eine Integrationsarchitektur (Integration Architecture).

Die Wiederverwendung bestehender Systeme ist integraler Bestandteil von SOA. Dies bedeutet, dass ein strukturierter Umgang mit Legacy-Systemen notwendig ist, um deren Funktionalität optimal nutzen zu können. Oft bleibt es jedoch nicht bei der reinen Nutzung bestehender Systeme. Wird ein solches System einmal mit einem Service Interface versehen, so werden die Mängel offensichtlich. In diesem Fall erweitert oder ersetzt man Teile oder gar das ganze System. Handelt es sich um ein Legacy-System mit großem Funktionsumfang, hilft nur ein Re-Engineering basierend auf einer durchgehenden Methodik, um qualitativ gute Resultate zu erreichen. Ein Re-Engineering eines größeren Systems in Richtung SOA ist auf jeden Fall mit einem mehrstufigen Verfahren durchzuführen, das eine Gewichtung der Komplexität anhand des Schwierigkeitsgrades der Umstellung erlaubt. Es lohnt sich selten, diejenigen

Teile eines Legacy-Systems weiter zu verwenden, die nur mit sehr großem Aufwand „servicefähig“ umgebaut werden können.

### 1.3.1 Eigenschaften des SOA-Modells

Grundlegende Eigenschaften des SOA-Modells:

- Die **Trennung der Business-Logik in dynamische (Prozesse und Regeln) und statische (applikatorische Umsetzung) Bereiche**. Der dynamische Bereich wird getrennt in einer separaten Schicht (Orchestration) als Prozess oder als Regel modelliert und ausgeführt. Ein Prozess wird immer als ausführbarer Prozess mit einer ausführbaren Prozessmodellierungssprache wie beispielsweise BPEL (Business Process Execution Language) oder BPMN (Business Process Modelling Notation) umgesetzt. Eine Regel wird immer als ausführbare Regel umgesetzt. Der statische Bereich der Business-Logik wird als Service realisiert. Dieser Service wird in jedem Fall als Web Service realisiert.
- Die **Integration Architecture** ist eine logische Komponente, die nicht unbedingt durch einen Enterprise Service Bus abgedeckt werden muss. Die minimale Realisierung einer Integration Architecture ist ein Netzwerk und die Unterstützung der Internet Protokoll-familie HTTP, DNS und TCP/IP.
- Die **Weiterverwendung** bestehender Anwendungen ist integraler Bestandteil der Architektur. Die Funktionalität dieser Anwendungen wird als Service gekapselt. Der Zugriff oder die Steuerung dieses Service erfolgt ausschließlich über eine Web Service-Schnittstelle.
- Der **Service Layer** basiert auf den Mechanismen, die den beiden Standards SOAP und WSDL entsprechen müssen. Andere Standards wie beispielsweise UDDI sind nicht unbedingt notwendig, um eine SOA auf Unternehmensebene zu betreiben.

## 1.4 Referenzmodelle

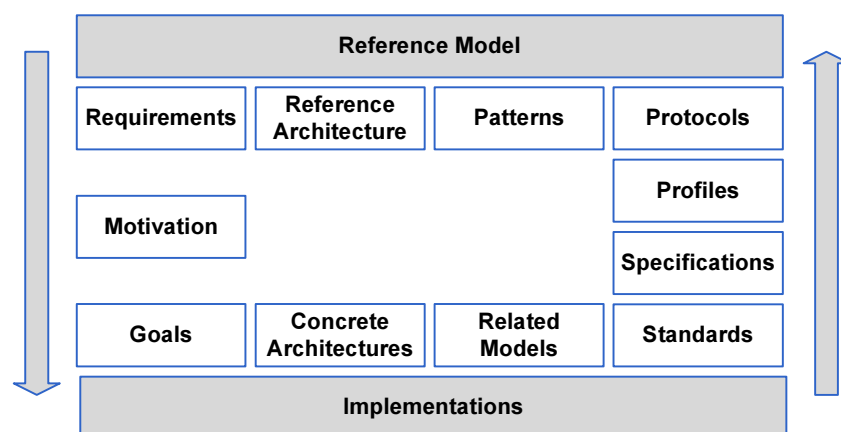


Abbildung 5: Referenzmodell & Implementation

Referenzmodelle zeigen eine idealisierte Sichtweise auf eine Architektur. Sie haben eine technologie- und hersteller-unabhängige Sichtweise auf eine bestimmte Architektur zum Ziel. Im Zentrum der Betrachtung stehen Referenzarchitekturen und Anwendungsszenarien, die bestimmte Anforderungen und Ziele erfüllen sollen und eine Reihe von Standards, Spezifikationen, Profile und Protokolle umsetzen.

Für die Praxis sind Referenzmodelle wichtig, da sie anhand einer Idealvorstellung die wichtigsten Konzepte hinter SOA oder einer anderen Standardarchitektur konkretisieren. Außerdem werden diese Modelle von vielen großen Beratungshäusern und Analysten verwendet. Ein grundlegendes Verständnis der Referenzmodelle hilft, die Angebote der Beratungshäuser, der Analysten und auch der Consultingabteilung der großen Hersteller zu bewerten und das Richtige auszuwählen.

## 1.5 Das SOA Referenzmodell des W3C

Die Web Services Architecture Working Group des World Wide Web Consortium (W3C) hat im Jahr 2004 eine so genannte Working Group Note veröffentlicht, die SOA als eine Ausprägung der Web Service Architecture beschreibt [Booth et Al. 2004]. Ausgangspunkt der Web Service Architecture ist die Definition des Web Service: „Ein Web Service ist ein Software-System, das konstruiert worden ist, um eine interoperable Maschinen-zu-Maschinen-Interaktion über ein Netzwerk zu unterstützen. Er hat eine Schnittstelle, die in einem für Maschinen bearbeitbaren Format beschrieben ist (WSDL – Web Service Description Language). Andere Systeme interagieren mit einem Web Service vorschriftsmäßig gemäß dessen Beschreibung über SOAP-Meldungen, die typischerweise über HTTP mittels XML-Serialisierung in Verbindung mit anderen Web Service-Standards versendet werden.“

Basis des SOA-Metamodells des W3C ist die WWW-Architektur, die als W3C Recommendation definiert ist [Jacobs, Walsh 2004]. Dieses Dokument beschreibt das Web als Informationsraum (Information Space) und dessen einzelne „Items of Interest“ als Ressourcen, die über URI (Uniform Ressource Identifiers) global identifiziert werden können.

Ein weiterer zentraler Begriff ist der Agent, der eine beteiligte Partei an einer Interaktion über das Web beschreibt. User sind „User Agents“, Servicenehmer „Requestor Agents“ und Services „Provider Agents“.

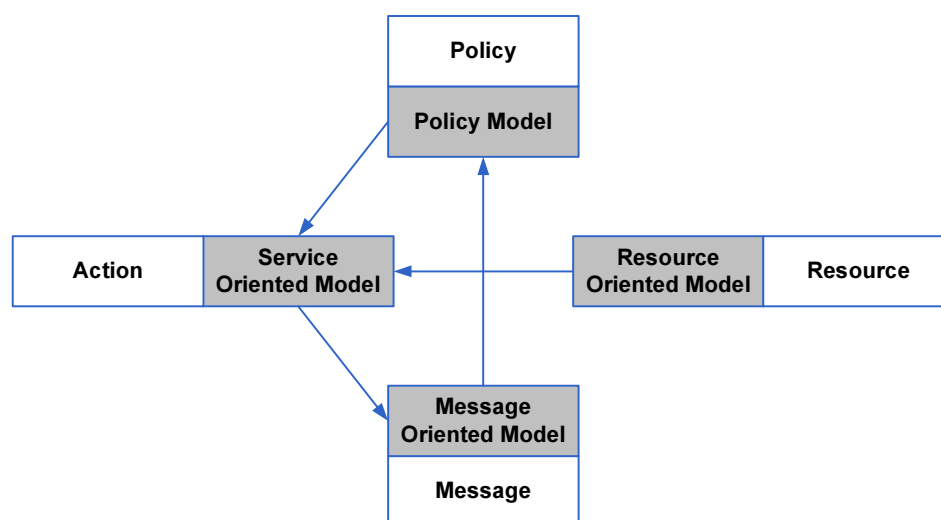


Abbildung 6: Die Architekturmodelle des W3C

Das SOA Meta-Modell des W3C umfasst vier Modelle, die „teilweise geschichtet“ miteinander korrelieren:

- **Message Oriented Model:** Dieses Modell beschreibt die Art und Weise, wie ein Agent – die konkrete Implementation eines Service – Meldungen sendet und empfängt.
- **Resource Oriented Model:** Das Modell beschreibt die existierenden Ressourcen. Dieses Modell basiert auf dem der WWW-Architektur des W3C, das eine Ressource als Entität beschreibt, die eindeutig identifiziert und mit der mittels definierter Formate interagiert werden kann.
- **Policy Model:** Das Policy Model definiert die für die Interaktion zwischen Ressourcen und Agents geltenden Grundsätze und Richtlinien. So werden beispielsweise Security und Quality of Service im Policy Model festgeschrieben.
- **Service Oriented Model:** Gemäß W3C ist dieses Modell eine Form eines verteilten Systems, das auf der Interaktion diverser getrennter Software-Agenten basiert. Die Eigenschaften dieses Modells sind folgende:
  - **Logical View:** Ein Service ist eine abstrahierte logische Sicht auf konkrete Programme, Datenbanken und Geschäftsprozesse. Diese Sicht definiert, was ein Service tut, typischerweise die Ausführung einer Operation auf Business-Ebene.
  - **Message Orientation:** Der Service wird formell durch die Meldungen, die zwischen einem „Provider Agent“ und dem „Requestor Agent“ ausgetauscht werden, definiert und nicht durch die Eigenschaften der Agenten selbst. Die konkrete Realisierung eines Service wird abstrahiert hinter der Service-Schnittstelle.

- **Description Orientation:** Ein Service wird durch maschinenlesbare Meta-Daten beschrieben. Die Semantik des Service wird ausschließlich durch seine Beschreibung dokumentiert.
- **Granularity:** Services tendieren dazu; relativ wenige Operationen werden mit komplexen Meldungen abgearbeitet.
- **Network Orientation:** Services werden dahingehend gestaltet, dass sie sich über ein Netzwerk ansprechen lassen.
- **Platform Neutral:** Die Meldungen werden in einem Plattform-unabhängigen und standardisierten Format übertragen.

Der Aufbau der vier Modelle und die beteiligten Objekte werden als Graph dargestellt. Sämtliche Objekte werden einzeln definiert.

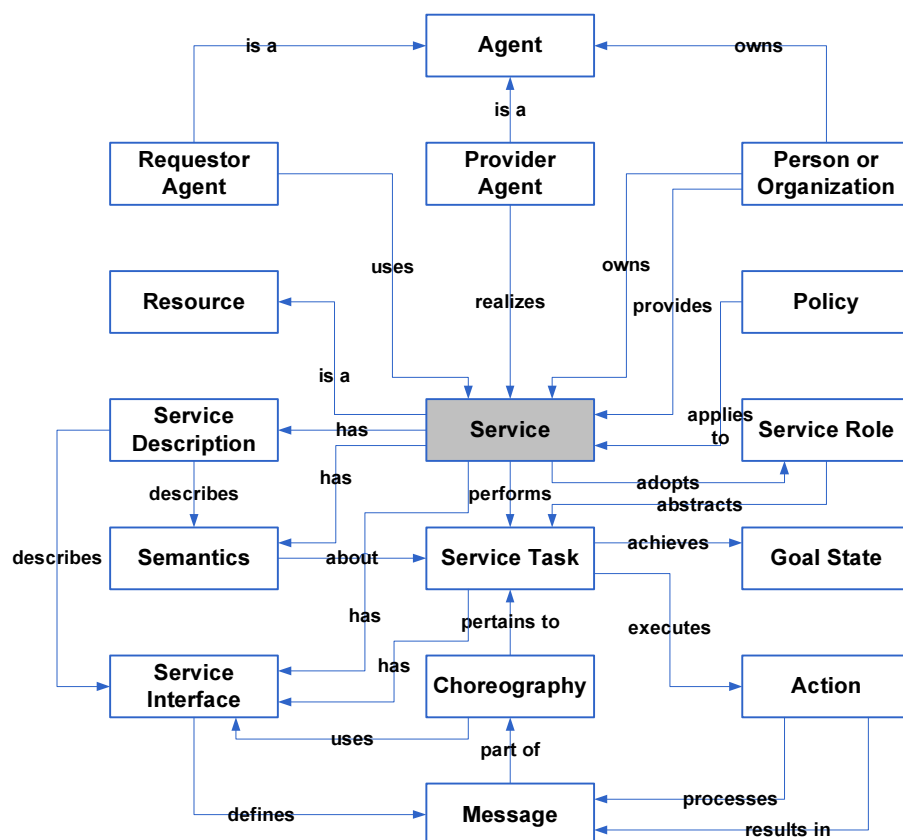


Abbildung 7: Das Service Oriented Model des W3C

Das Service Oriented Model des W3C beschreibt die statischen Beziehungen zwischen den Objekten, die an der Realisierung eines Service gemäß W3C beteiligt sein sollten. Zentrale Konzepte sind der Service, die Aktion und der Agent. Der Service wird als Partei verstanden, die Aktionen auf Geheiß einer anderen Partei ausführt. Interessanterweise werden die konkreten Aktionen der Provider und der Requestor Agents aus der Betrachtung des W3C ausgeschlossen. Als Agent wird ein Software-Agent im Sinne der WWW Architecture Recommendation verstanden. Ein Agent realisiert immer einen Web Service, er ruft auch immer einen Web Service auf.

### 1.5.1 Bedeutung

Das SOA Meta Model des W3C ist die Grundlage für das Verständnis der vielen Web Service-Standards, weil das Modell auf der globalen WWW-Architektur basiert. Die WWW-Architektur ist wiederum eine wichtige Grundlage für das korrekte Funktionieren des Informationsraums Internet. Die Stärke des Modells liegt in der klaren und einfachen Darstellung der Beziehungen zwischen den zentralen Objekten, wie dem Service, der Message, der Ressource und der Policy und anderen Entitäten wie beispielsweise der Aktion, der Serviceschnittstelle und vielen anderen mehr. Das W3C geht von einer globalen SOA-Architektur aus, also von einem globalen Markt der Funktionen. Darin bestehen auch die Schwächen des Modells in Bezug auf die Anwendbarkeit für ein Unternehmen. Die starke Betonung der

Modelle in Richtung globaler Einsatzmöglichkeiten detailliert die einzelnen Entitäten so weit, dass eine konkrete Realisierung nur mit sehr großem Aufwand erfolgen kann. Dennoch ist und bleibt das SOA Meta Model des W3C für jedes Unternehmen ein wichtiges Instrument, um die verschiedenen Standards zu verstehen. Für die spezialisierten Gremien (Architektur, Compliance, Governance) größerer Unternehmen ist es ein absolutes Muss, dieses Modell zu kennen.

## 1.6 Referenzmodelle für generische Unternehmen

### 1.6.1 GereriCo

Im Jahr 2004 lancierten 21 Firmen gemeinsam die SOA-Blueprints-Initiative, die versuchte, anhand einer SOA-Implementierung für das fiktive Unternehmen GeneriCo eine Referenzarchitektur zu realisieren. GeneriCo geht von einem betrieblichen Informationssystem für ein Unternehmen aus, das eine Reihe so genannter Enterprise Applications, einen allumfassenden Security-Mechanismus, ein Portal und eine Reihe von Individualanwendungen umfasst.

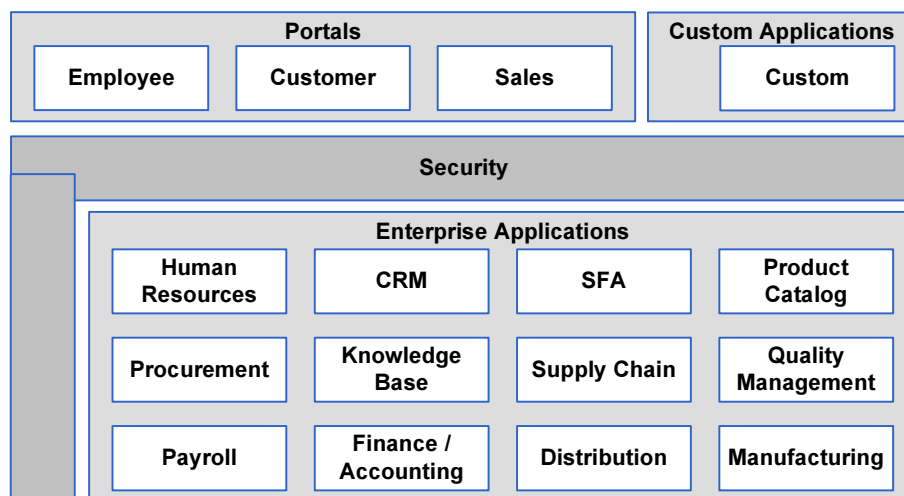


Abbildung 8: Betriebliches Informationssystem gemäß GeneriCo

Die Referenzarchitektur wurde abgestimmt auf eine Reihe von ausgewählten Funktionen [Pellengahr 2005]:

- Ein verteilter, unternehmensweiter Sicherheitsmechanismus
- Ein Mitarbeiterportal
- Ein Katalog von Produktdaten mit Suchfunktion
- Eine Anwendung zur Personalverwaltung
- Eine Auswahl von Funktionen im Bereich der Gehaltsabrechnung

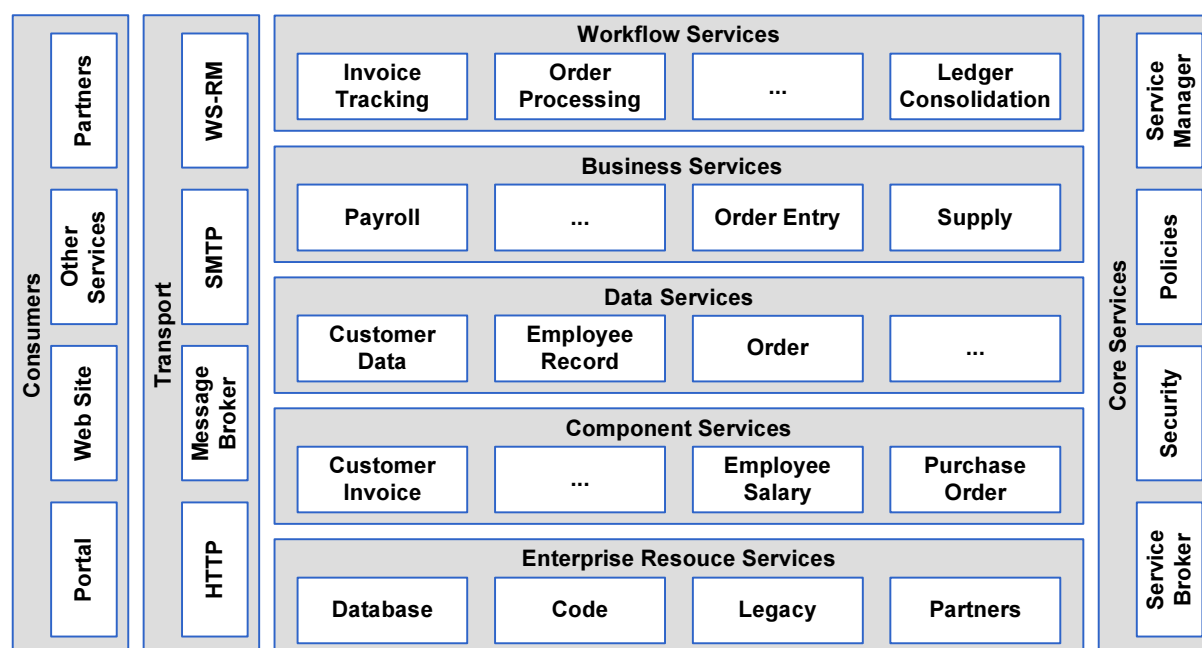


Abbildung 9: Die SOA-Referenzarchitektur, basierend auf dem Reference Beispiel

Die Servicelandschaft wurde im Rahmen der Spezifikation in verschiedene Bereiche aufgeteilt [Wilkes, Harby 2004]. Aus dieser Spezifikation ergibt sich ein Service Layering, das Core Services, Enterprise Resource Services, Component Services, Data Services, Business Services und Workflow Services umfasst. Zudem sind Transportmechanismen und eine Reihe von Consumer Systems vorgesehen. Der Definition des Blueprints folgte ein so genanntes SOA Blueprints Reference Example, das vor allem Services und deren Interaktion für die Personal-Verwaltung und den Katalog der Produktdaten genau spezifiziert und die Realisierung des Portals für die Angestellten mit seinen Sicherheitsmechanismen erläutert.

### 1.6.2 Real-Time Enterprise

Ein weiteres generisches Unternehmensmodell ist das Echtzeit-Unternehmen, welches sich aus den entsprechenden Modellen der ERP (Enterprise Resource Planning)-Systemherstellern entwickelt hat [Alt, Österle 2003]. Das so genannte Echtzeit-Unternehmen oder auch Real-Time Enterprise (RTE) ist ein idealisiertes Modell eines Unternehmens. Es geht davon aus, dass sich eine Firma sofort auf die Bedürfnisse seiner Kunden einstellen kann und die entsprechenden Produkte auch sofort liefert [Scheer et al., 2003].

Real-Time Enterprise basiert den Prinzipien Anyhow, One-Stop, Everything, Non-Stop, Everywhere und One-To-One. Damit Real-Time Enterprise als Unternehmensmodell umgesetzt werden kann, sind die betrieblichen Informationssysteme so zu gestalten, dass die „richtige Information zum richtigen Zeitpunkt am richtigen Ort zum richtigen Zweck verfügbar ist“ [Nussdorfer, Martin, 2003].

Die Umsetzung von RTE besteht aus Sicht der IT immer aus zwei grundlegenden Bestandteilen:

- **Flexible Prozesse:** Sämtliche Geschäftsprozesse sind vollständig modelliert und auf die entsprechenden IT-Systeme abgebildet. Eine Änderung am Geschäftsprozess zieht eine sofortige Änderung des unterstützenden IT-Systems nach sich. Diese Änderungen können in Echtzeit durchgeführt werden. Diesen Aspekt wird mit einer SOA abgedeckt.
- **Real-Time Information:** Sämtliche Informationen eines Unternehmens sind immer auf dem aktuellen Stand und zeigen die momentane Situation des Unternehmens auf operativer, taktischer und strategischer Ebene. Dieser Aspekt wird mit einer guten Business Infrastruktur abgedeckt (siehe Vorlesung „Operative Systeme“ – Anhang).



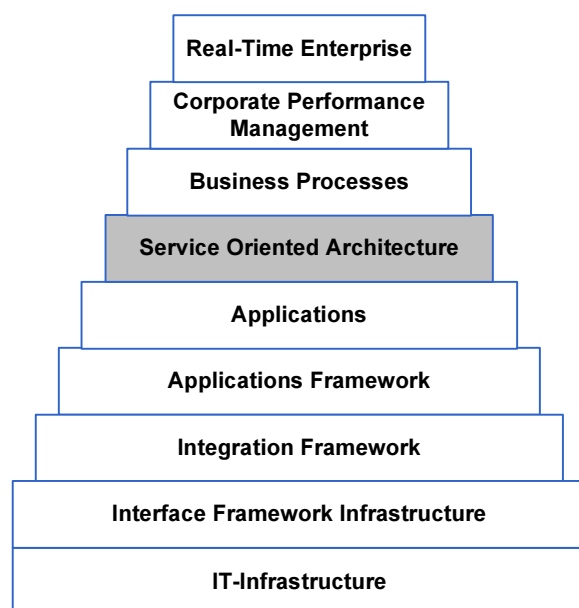


Abbildung 10: Die Ebenen der RTE-Architektur (nach [Schmale 2004])

SOA ist ein zentraler Bestandteil der Gesamtarchitektur eines RTE, der die beiden grundlegenden Eigenschaften eines solchen idealisierten Systems, Real-Time Information und flexible Prozesse, auf die Gesamtlandschaft der betrieblichen Informationssysteme eines Unternehmens abbildet. SOA ist damit eine Ebene der RTE-Architektur dargestellt.

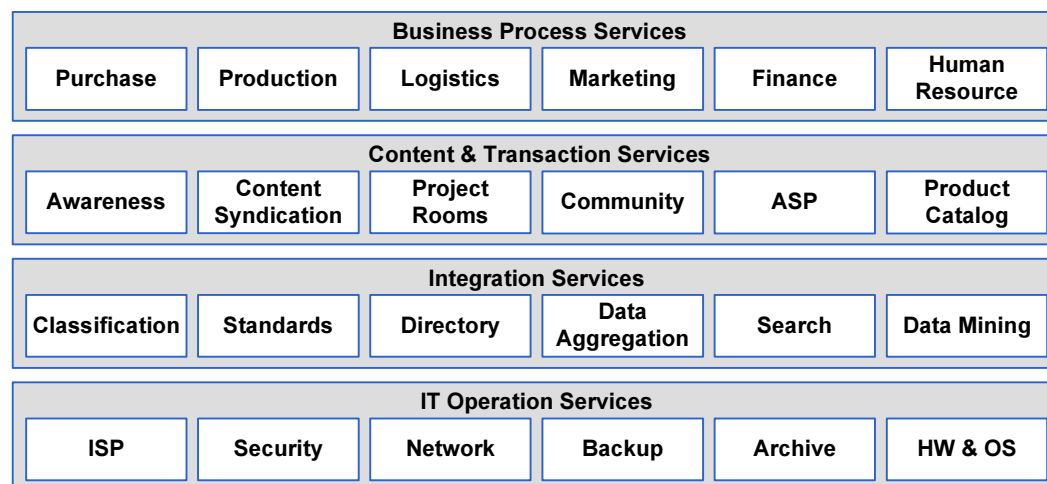


Abbildung 11: Web Service-Landkarte für ein RTE (nach [Alt, Österle 2003])

Die Servicelandkarte eines RTE hat vor allem Ordnungsfunktion. Es wird eine Zuordnung von Diensten auf verschiedenen Ebenen möglich. Obwohl die Landkarte ein idealisiertes Unternehmen darstellt, können solche Zuordnungen in der Praxis bei der Analyse bestehender Funktionalität sehr hilfreich sein. Die Landkarte sieht die Kategorien Business Process Services, Content & Transaction Services, Integration Services und IT Operation Services vor.

- **Business Process Services:** Die Services zur Unterstützung der Kernprozesse eines Unternehmens (Einkauf, Produktion, Logistik, Marketing, ...) werden auf dieser Ebene realisiert.

- **Content & Transaction Services:** Die „Information Worker“ Services, wie beispielsweise die Bereitstellung von internen und externen Informationen und die Unterstützung der Teamarbeit, werden durch Content & Transaction Services realisiert.
- **Integration Services:** Der Zugriff auf die verschiedenen Datenbereiche des Unternehmens erfolgt über diese Services.
- **IT Operation Services:** Die Basisdienste des IT-Betriebs.

### 1.6.3 Bedeutung

Generische Unternehmensmodelle entwickeln aus einem sehr detaillierten, jedoch fiktiven Beispiel eine Referenzarchitektur. Alle diese Modelle gehen von einer gesamtheitlichen Betrachtungsweise des Unternehmens und seiner Kernprozesse aus. Sämtliche Prozesse werden durch eine geeignete Architektur unterstützt. Viele Modelle – wie beispielsweise das GeneriCo-Modell der SOA-Blueprints-Initiative oder der Real-Time-Enterprise-Ansatz – resultieren in einer logischen Aufteilung der Services in verschiedene Bereiche, so genannte Service-Landkarten. Die Stärke dieser Referenzarchitekturen sind der illustrative Charakter der Lösung und die Gesamtsicht auf ein Unternehmen. Sie stellen daher eine wichtige Informationsquelle für ein Unternehmen der entsprechenden Branche dar. Die Schwäche dieser Modelle ist der fehlende Bezug zur Realität der bestehenden Systeme einer betrieblichen Informatik und somit die Illusion, dass SOA nur ein Ansatz für die „Enterprise Architecture“ eines großen Unternehmens ist. Spätestens bei der Ausbreitung einer SOA in einem Unternehmen wird die Service-Landkarte zu einem wichtigen Struktur- und Kommunikationsinstrument. Schon alleine deshalb sollten Sie wissen, wie Sie an die guten Beispiele herankommen.

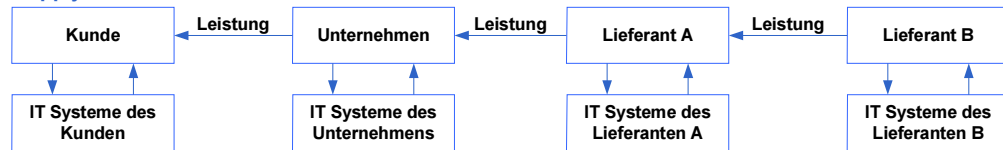
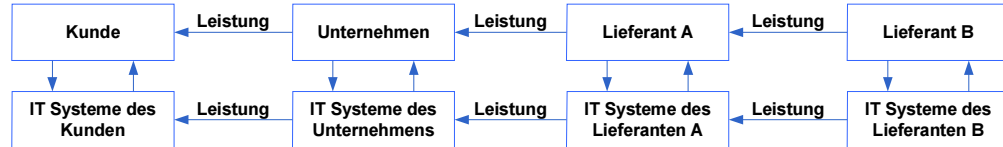
## 1.7 Beispiel: IT Supply Chain dank SOA

Innerhalb eines Unternehmens müssen sämtliche Primärprozesse (Einkauf, Produktion / Dienstleistung, Verkauf, Kundenbetreuung), sowie die Management Prozesse (Strategisches sowie Operatives Management) durch die Informationstechnologie unterstützt werden. Meist werden die Sekundärprozesse auch durch verschiedene Systeme unterstützt. Dies führt zu einer Vielzahl von Komponenten innerhalb eines Betrieblichen Informations-Systems und damit zu einer grossen Funktionstiefe. Diese Funktionstiefe reflektiert in den meisten Fällen die Unternehmenstätigkeit, stellt sie im Idealfall sogar 1:1 dar. Alle Unternehmen und alle Organisation arbeiten mit anderen Organisationen und Unternehmen zusammen. Die meisten von Ihnen haben einen oder mehrere Kunden. Die betriebliche Tätigkeit findet immer in einem Kontext anderer Unternehmen und Kunden statt. Die Informationssysteme der einzelnen Unternehmen arbeiten jedoch nur in seltenen Fällen nahtlos zusammen. Voraussetzungen für eine solche Zusammenarbeit sind eine globale Infrastruktur, Interoperabilität und die Möglichkeit des formatunabhängigen Datenaustauschs.

Service Oriented Architectures setzen auf einer bestehenden globalen Infrastruktur auf, dem Internet. Somit ist die sichere Verbindung von Funktionalität über Unternehmensgrenzen hinweg gewährleistet. Ein bestimmter Service, also eine bestimmte Umsetzung eines Primärprozesses eines Unternehmens ist damit genauso auf dem Markt verfügbar, wie der Leistungsprozess, respektive die Unternehmensleistung selbst.

Sämtliche Plattformen (Windows, UNIX, und Grossrechner) und sämtliche Keyplayers (IBM, Microsoft, SAP, HP, u.a.) unterstützen SOA. Technologieunterschiede zwischen Unternehmen sind also kein Hindernis für die Leistungserbringung mehr. Die Interoperabilität ist eine inhärente Eigenschaft von SOA.

Der Datenaustausch zwischen einzelnen SOA Diensten erfolgt in ASCII. Die verschiedenen Datenformate sind durch Standardmechanismen formal definiert. Damit ist eine wichtige Voraussetzung zum Datenaustausch zwischen Organisation gegeben. Transformationen zwischen verschiedenen Datenformaten, respektive semantische Informationen sind als Metadaten verfügbar und können von Software Convertern verwendet werden.

**Supply Chain ohne SOA****SOA Supply Chain****Abbildung 12:** Unternehmensübergreifende Supply Chain

Die Leistungserstellung eines Unternehmens erfolgt immer über eine Supply Chain (Lieferkette), die sowohl die Lieferanten als auch die Kunden des Unternehmens mit einbezieht. SOA erlaubt die Abbildung einer Geschäftsübergreifenden Supply Chain in die IT Systeme der beteiligten Unternehmen. So könnte beispielsweise der Verrechnungsprozess direkt von IT System zu IT System abgewickelt werden. Die Abfrage des Lagerbestandes sowie die Kontrolle von Lieferterminen sind weitere einfache Beispiele der möglichen Zusammenarbeit über Unternehmensgrenzen hinweg.

# Grundlegende SOA Modelle der grossen Hersteller

## 2.1 Einleitung

Will ein Unternehmen SOA einsetzen, sind in jedem Fall bestehende Systeme und Infrastruktur-Komponenten betroffen. Jedes Unternehmen hat eine Reihe von Standard Produkten bestimmter Hersteller im Hause. Es gibt kein Unternehmen, das nicht Software einsetzt, die entweder von IBM, SAP, Oracle oder Microsoft stammt oder auf Open Source-Technologie basiert. Jeder dieser Anbieter verfügt über einen eigenen mehr oder weniger vollständigen SOA-Stack, bestehend aus diversen Produkten, die gemäß den Angaben des Anbieters sämtliche Bedürfnisse eines Unternehmens bezüglich SOA abdecken. Es existieren nur sehr wenige Unternehmen und Organisationen, die lediglich Produkte eines Herstellers einsetzen. Ebenso unwahrscheinlich ist es, dass ein einziger Hersteller den idealen SOA-Stack für ein bestimmtes Unternehmen anbieten kann. Damit Sie die Angebote eines Herstellers richtig einschätzen und in Ihrem Kontext vernünftig einsetzen können, ist es notwendig, das jeweilige grundlegende SOA-Modell eines Herstellers zu kennen. Dieses Modell ist einer der Indikatoren, ob sich das konkrete Angebot des Herstellers mit Ihren betrieblichen Informationssystemen, Ihrer bestehenden „Installed Base“ also verträgt.

- **IBM:** SOA als Praxis, die davon ausgeht, dass die Architektur eines Informationssystems aus der Geschäftsarchitektur, einer Zusammenstellung von Services eines Unternehmens sowie den entsprechenden Richtlinien und Regeln für deren Einsatz abgeleitet werden kann.
- **SAP:** SOA als Instrument zur Abbildung betrieblicher Prozesse in die Informationssysteme eines Unternehmens, also als so genannte Enterprise SOA, die auf Enterprise Web Services aufbaut.
- **Oracle:** SOA als eine standard-basierte Plattform, die es erlaubt, Dienste (Web Services) bereitzustellen, aufzufinden und gegenseitig zu benutzen.
- **Microsoft:** SOA als Resultat bestimmter Grundsätze, Praktiken und Frameworks, die es erlauben, normierte Services bereitzustellen, sowie als Architektur einer Applikations-Plattform..

## 2.2 IBM

Die IBM versteht SOA als Praxis, die Architektur eines Informationssystems aus der Geschäftsarchitektur eines Unternehmens abzuleiten. Unter Geschäftsarchitektur versteht man im Kern eine Zusammenstellung von Services sowie die entsprechenden Richtlinien und Regeln für deren Einsatz. Ein Service ist ein wiederholbarer Arbeitsschritt innerhalb eines Geschäftsprozesses. Das SOA-Modell von IBM besteht aus mehreren Teilen, die SOA Foundation ist als Referenzarchitektur der Rahmen um das IBM-Produktangebot, die On-Demand Service-Oriented Architecture als standardisierte Lösung einer SOA und eine große Sammlung von SOA (Integration) Patterns.

## 2.2.1 IBM SOA Foundation

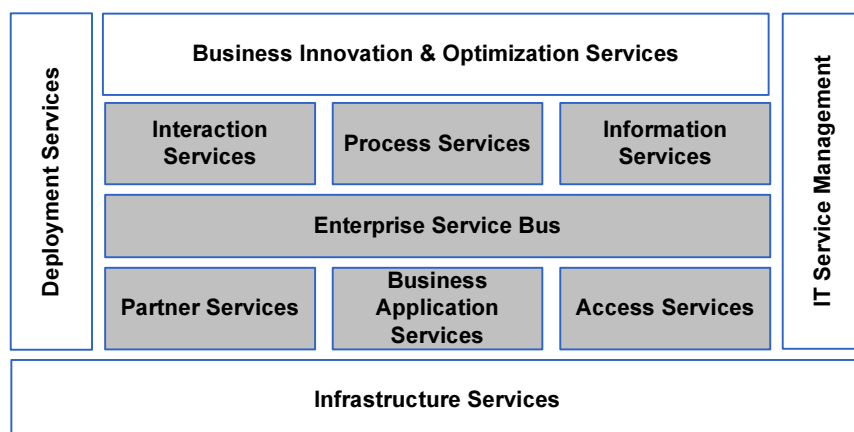


Abbildung 13: Die SOA-Referenzarchitektur von IBM (SOA Foundation)

Die SOA Foundation besteht aus dem Enterprise Service Bus, den verschiedenen Servicekategorien und einer Reihe unterstützender Bereiche. Die zentrale Komponente der Architektur ist der Enterprise Service Bus, der als Weiterentwicklung einer Messaging Infrastruktur und einer Enterprise Application Integration-Infrastruktur verstanden wird [High et Al. 2005].

Die Servicekategorien der SOA Foundation:

- **Interaction Services:** Diese Dienste sind die Schnittstelle zwischen Anwendungen und dem User (Mensch, Roboter, Sensoren, etc.). Die Sicht auf eine Anwendung kann je nach Bedarf auf den User zugeschnitten werden.
- **Process Services:** Die Services dieses Bereiches sind für den Ablauf einer bestimmten Logik zuständig. Sie werden entweder als Business Process Flows oder als Business State Machines realisiert. Business Process Flows werden mittels Process Server ausgeführt, während Business State Machines durch Rule Engines realisiert werden.
- **Information Services:** Die Information Services enthalten den Zugriff auf die persistenten Daten und die Kontrolle über den Datenfluss durch eine Organisation. Diesem Bereich werden das Data Management, die Information Integration, Business Intelligence, Content Management und Master Data Management zugeordnet.
- **Partner Services:** Diese Dienste sind für den Zugriff auf eventuell involvierte Services bestimmt, die nicht von der eigenen Organisation bereitgestellt werden. Auf dieser Ebene werden Industriestandards wie etwa RosettaNet [Damodaran 2004], EDIFACT [Salminen 1995], SWIFT [Weitzel, Martin 2002] oder andere unterstützt.
- **Business Application Services:** Diese Services realisieren die Business-Logik der Organisation.
- **Access Services:** Bestehende Systeme werden über Access Services in eine SOA integriert. Die Integration von Legacy-Systemen und die Realisierung von entsprechenden Adaptoren werden auf dieser Ebene umgesetzt.

Rund um den Enterprise Service Bus und die Services sind eine Reihe von unterstützenden Bereichen formuliert, wie das Monitoring der Services (Business Innovation & Optimization Services), die technische (Deployment Services) und organisatorische (IT Service Management) Bereitstellung von Diensten sowie die Dienste, die eine Infrastruktur repräsentieren (Infrastructure Services).

Der SOA Produkt-Stack von IBM besteht hauptsächlich aus den drei Produktfamilien WebSphere, Rational und Tivoli. WebSphere dient dabei als namensgebender Rahmen für eine Vielzahl von Produkten, die schon lange auf dem Markt sind. Das prominenteste Beispiel dafür ist MQSeries, das nun WebSphere Message Broker heißt. Eine IBM SOA Implementation ohne WebSphere-Produkte ist undenkbar.

## 2.2.2 On Demand Operating Environment

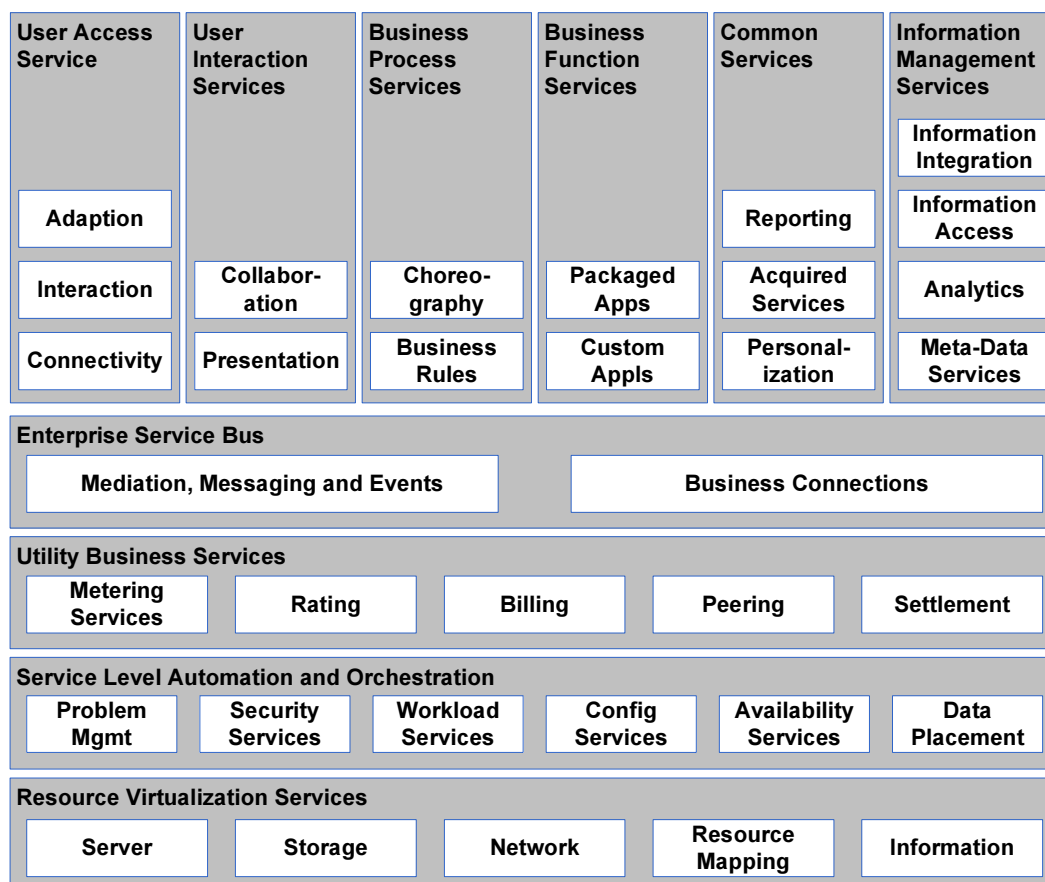


Abbildung 14: Architektur des IBM On Demand Operation Environments

Das IBM On Demand Operating Environment geht nach [Crawford et Al. 2005] von fünf Schichten aus:

- Die **Application Services** sind in die Bereiche User Access Services, User Interaction Services, Business Process Services, Business Function Services, Common Services und Information Management Services aufgeteilt. Unter ihnen befinden sich einzelne Dienste wie beispielsweise die Collaboration, die die Zusammenarbeit unterstützen.
- Der **Enterprise Service Bus (ESB)** oder auch Enterprise Integration Bus (EIS) besteht aus den Komponenten Mediation, Messaging and Events und den Business Connections und ist für die Kommunikation zwischen den diversen Diensten zuständig.
- Die **Utility Business Services** umfassen Dienste, wie die Metering Services, das Rating, das Billing, das Peering und das Settlement. Alle diese Dienste haben mit der Funktionalität des Systems als Ganzem nichts zu tun. Sie unterstützen lediglich die administrative Verwaltung der Dienste und die Darstellung der Qualitätsmerkmale. Außerdem wird ein Billing Service definiert, der die interne und externe Verrechnung von Diensten erlaubt.
- Die **Service Level Automation and Orchestration-Schicht** unterstützt den Betrieb der Services. Dabei sind die ITIL-Tätigkeiten Problem Management und Configuration Management als spezielle Gruppe von Diensten sichtbar. Workload, Availability und Data Placement Services reflektieren die Verwaltung der virtualisierten Ressourcen.
- Die **Resource Virtualization Services** stellen die für den Betrieb der Dienste unumgängliche Hardware eines Unternehmens dar. Dabei geht IBM von einer vollständig virtualisierten Infrastruktur aus, die je nach Bedarf Ressourcen hinzunehmen oder wegschalten kann.

„On Demand“ ist ein IBM-Marketingbegriff, der eine skalierbare und überall verfügbare Service-Infrastruktur suggeriert. Tatsächlich sind IBM-Produkte verfügbar, die eine Skalierung erlauben. So hat IBM Server im Angebot, die im

vollständigen und maximalen Ausbau geliefert werden. Der Kunde bezahlt jedoch nur, was er an Leistung (CPU, Memory, Disk) bezieht. Eine Zuschaltung von mehr Computing Power erfolgt lediglich durch das Freischalten bereits vorhandener Ressourcen.

### 2.2.3 Bedeutung

Der IBM SOA-Stack ist sehr differenziert. IBM hat als Hersteller die längste Erfahrung im Bereich Enterprise Service Bus und mit MQSeries (WebSphere Message Broker) jenes Produkt im Portfolio, das sich der größten Marktverbreitung erfreut. Eine Stärke des ESB von IBM ist das „Reliable Messaging“, welches eine garantierte Meldungsübermittlung erlaubt. In einem lose gekoppelten Umfeld ist diese Art von Kommunikation nicht immer die beste Lösung, vor allem dann, wenn Dienste nicht sehr oft und nicht zeitkritisch verwendet werden sollen. Der Kunde muss sich also fragen, ob er die entsprechende Komplexität auf sich nehmen will. Als weitere Besonderheit fällt die Differenziertheit des IBM SOA-Stack auf. Sicher hat IBM die größte Anzahl an Produkten in ihrem Stack und ist in den Bereichen Virtualisierung und ESB führend. Im Gegensatz zu den anderen Herstellern bietet IBM jedoch vor allem die Infrastruktur an, diese jedoch in einer besonders ausgereiften Form, da selbst betriebliche ITIL-Funktionen [Weston et Al. 2003] unterstützt werden. Das konkrete System muss darauf aufbauend realisiert werden. Die Problematik, eine heterogene Systemlandschaft zu einem flexiblen, funktionierenden Ganzen zusammenzufügen, bleibt.

## 2.3 SAP

SAP versteht SOA als Instrument zur Abbildung betrieblicher Prozesse in die Informationssysteme eines Unternehmens, also als so genannte Enterprise SOA. Konsequenterweise wird zwischen einem Web Service und einem Enterprise Web Service unterschieden. Der Enterprise Web Service erweitert den Standard Web Service dahingehend, dass von jeder Order-Entry-Anwendung (Auftragserfassung) darauf zugegriffen werden kann und der Service in der Lage ist, auf Daten verschiedener Quellen über einen einzigen zentralen Service zuzugreifen [Fritz 2004]. Eine Anwendung, die Enterprise Web Services verwendet, wird als Consuming Service oder auch als Composite Application bezeichnet. Enterprise Web Services und Composite Applications sind die Basis für die Enterprise Service Architecture (ESA), welche die SOA-Realisierung von SAP darstellt.

### 2.3.1 Enterprise Service Architecture (ESA)

Enterprise Service Architecture (ESA) wird als eine Reihe von Prinzipien definiert, die zum Ziel haben, eine Service-orientierte IT-Architektur für adaptive betriebliche Informationssysteme zu entwickeln.

Die wichtigsten Prinzipien von ESA sind folgende:

- Die Strukturierung der Anwendungsfunktionen in wieder verwendbare Komponenten, deren Realisierungskomplexität hinter standardisierten Schnittstellen versteckt wird.
- Die Bereitstellung von Mechanismen, um diese Komponenten über einen gemeinsamen Kontrakt zu verbinden, zusammenzustellen und zu konfigurieren. Solange dieser Kontrakt nicht verletzt wird, können sich die beteiligten Komponenten unabhängig voneinander verändern.
- Die Umsetzung von Standards, wie beispielsweise Web Services, Security und Prozessmodellierung, die ggf. mit Guidelines und Profilen erweitert werden.
- Die Unterstützung von Model Driven Software Development (MDSD [Mellor et Al. 2003]), um aus abstrakten Modellen den entsprechenden Code generieren zu können.
- Der Einsatz eines zentralen Service Repository, das unternehmensweit einheitliche Service Beschreibungen und ein einheitliches Programmiermodell bereitstellt.

ESA wird durch die Technologieplattform NetWeaver, die SAP xApps und durch die neuen Releases der SAP Business Suite unterstützt.

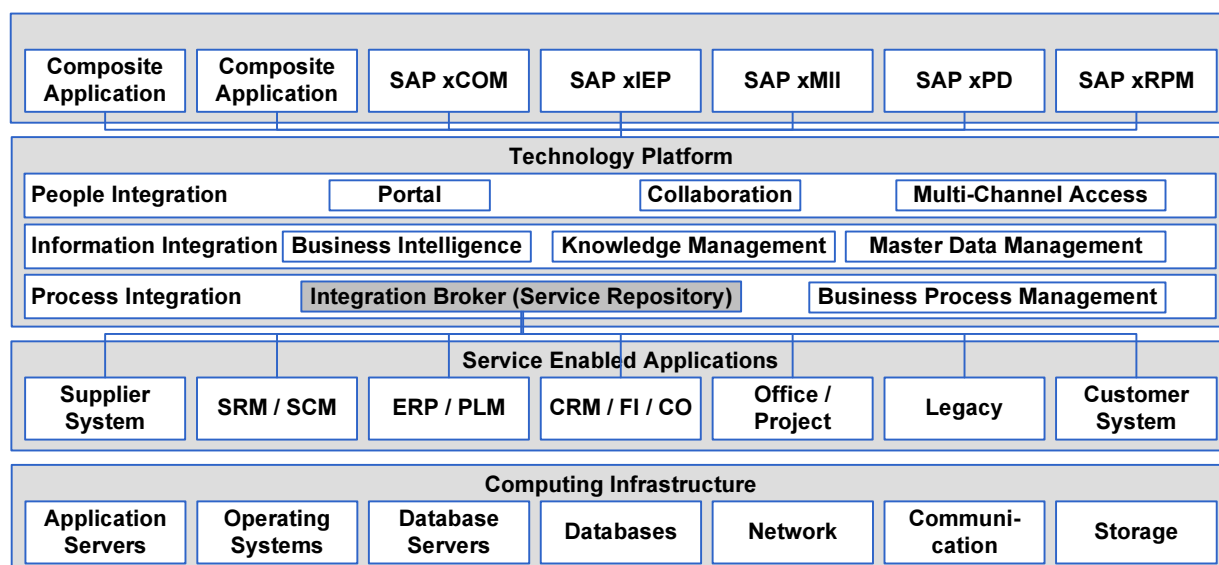


Abbildung 15: Enterprise Service Architecture (nach [Maloney 2004] und [Schumann 2007])

ESA ist in vier Ebenen aufgeteilt: Composite Applications (die Anwendungen), Technology Platform (NetWeaver), Service Enabled Applications (die bestehenden Anwendungen), Computing Infrastructure (die Basis).

- **Composite Applications:** Anwendungen werden als Composites (Consuming Services) mit dem Composite Application Framework (CAF) modelliert. Ausgangspunkt dieser Modellierung sind so genannte Business Scenarios, die eine Sammlung zusammengehörender Geschäftsvorfälle darstellen [Stotz 2003]. Diese Business Scenarios werden aus einem Process Architecture Model abgeleitet. SAP bietet eine Reihe von vorgefertigten Composite Applications an (xCOM – Cost and Quotation Management, xIEP – Integrated Exploration and Production, xMII – Manufacturing Integration and Intelligence, xPD – Product Definition, xRPM – Resource and Portfolio Management).
- **Technology Platform:** Die Technologieplattform von ESA ist der SAP NetWeaver, der als zentraler Integrationsmechanismus dient.
- **Service Enabled Applications:** Bestehende Systeme, SAP-Module und Office-Anwendungen, die servicefähig sein müssen, werden über ein zentrales Service Repository integriert.
- **Computing Infrastructure:** Die zum Betrieb notwendige Infrastruktur, bestehend aus Netzwerk, Speicher, Betriebssystemen, Datenbanken, Datenbank- und Applikations-servern.

ESA basiert auf der Business-Framework-Architektur, die mit SAP R/3 eingeführt wurde. Die Basis dieser Architektur ist ein Business Object, welches seine Funktionalität (Kernel) durch Konsistenzbedingungen (Integrity), eine Schnittstelle (BAPI) und standardisierte Zugriffsmethoden (ABAP – RFC) kapselt. Verschiedene Business-Objekte können in einer Business-Komponente, die eine betriebswirtschaftliche Funktionalität darstellt, gruppiert werden [Uhr 2004]. So ist auch zu verstehen, dass SAP zwischen einem Web Service und einem Enterprise Web Service unterscheidet. Der Enterprise Web Service ist nichts anderes als eine Business-Komponente mit einer Web Service-Schnittstelle.



### 2.3.2 SAP NetWeaver

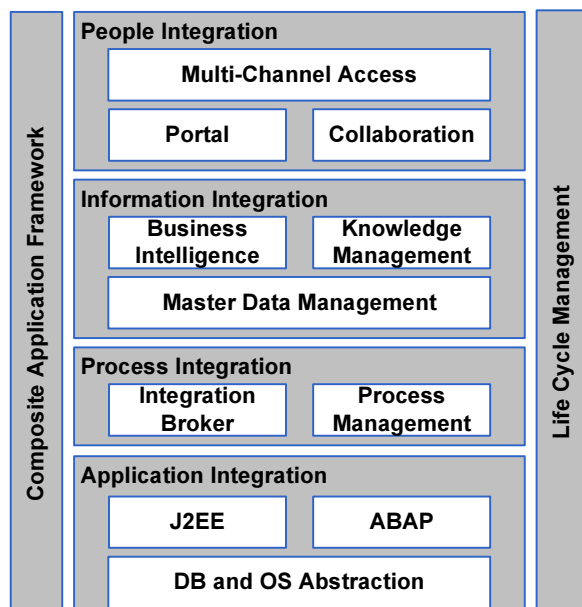


Abbildung 16: SAP NetWeaver

Die NetWeaver-Plattform ist das technologische Kernstück der SOA-Realisierung von SAP. Einerseits macht die Firma SAP damit ihre vielen Business-Komponenten als Enterprise Web Services zugänglich, andererseits hat NetWeaver den Anspruch, als Integrations-Plattform für andere Systeme die zentrale Komponente in einem Unternehmen darzustellen.

Die Architektur umfasst vier Integrationsschichten:

- **People Integration** – soll Personen die richtigen Informationen zur Verfügung stellen.
- **Information Integration** – fasst Komponenten zur Generierung von Mehrwert durch Integration von Informationen zusammen.
- **Process Integration** – ermöglicht das Zusammenspiel von heterogenen Komponenten innerhalb eines Geschäftsprozesses.
- **Application Plattform** – die gemeinsame Umgebung für ABAP- und J2EE-konforme Komponenten.

### 2.3.3 Die SAP XI Exchange-Plattform (heute SAP PI)

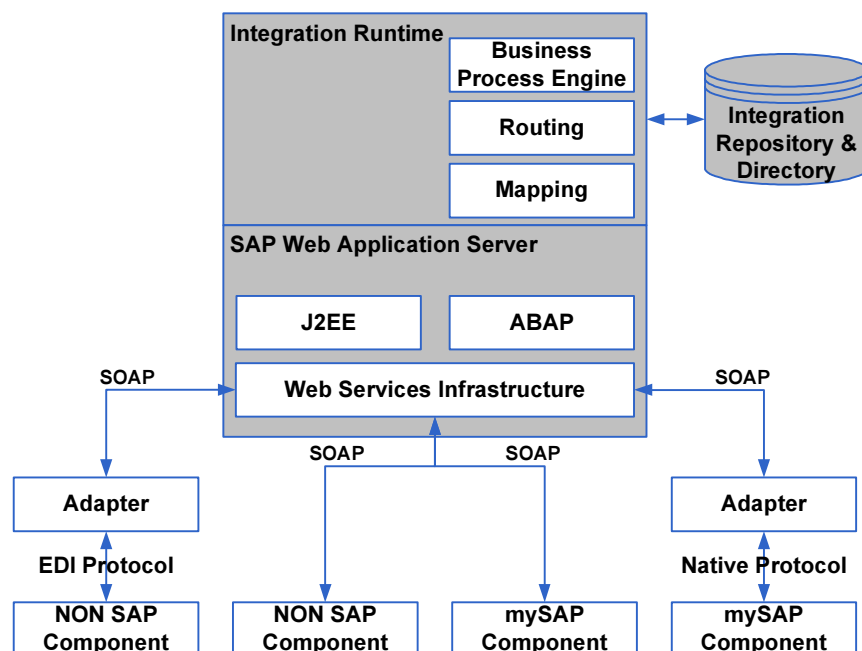


Abbildung 17: 14 SAP XI Exchange Infrastructure

Die SAP Exchange Infrastructure ist das Kernstück von SAP NetWeaver bezüglich der Anbindungen von Diensten an ein SAP-System. Es besteht aus einer Integration Runtime und dem SAP Web Application Server. Die Runtime-Komponente beinhaltet als wichtigste Komponente eine Business Process Engine, mittels derer einzelne Services zu einem Geschäftsprozess zusammengestellt und gesteuert werden können. Der SAP Web Application Server erlaubt das Nebeneinander von Programmen, die in Java oder in ABAP geschrieben worden sind. Die Web Services Infrastructure kommuniziert mittels SOAP mit allen SAP-Komponenten und auch mit anderen Web Services.

### 2.3.4 Bedeutung

Der SOA-Stack von SAP reflektiert die Stellung eines ERP-Systems in einem Unternehmen als Master. SAP geht davon aus, dass die Umsysteme zu SAP mittels SOA integriert werden können. Dies ist mit der NetWeaver-Plattform möglich. Die Verwendung einzelner SAP-Module als Enterprise Web Services in einer Umgebung, in der SAP nicht das Leading-System ist, gestaltet sich jedoch etwas umständlicher, da die Differenzierung Enterprise Web Service – Web Service und die Bereitstellung so genannter Consuming Services (Composite Applications) nicht allgemein üblich sind. Eine weitere Besonderheit des SAP SOA-Stacks ist die Tatsache, dass die Modellierung von Geschäftsprozessen durch Aris von IDS Scheer erfolgt. Aris ist seit Jahren im Bereich Modellierung von Geschäftsprozessen führend. Die Modellierung erfolgte ursprünglich jedoch mittels EPK (Ereignisgesteuerte Prozessketten) und nicht mittels BPEL. Heute werden BPEL und BPMN unterstützt.

## 2.4 Oracle

Oracle versteht SOA als eine standard-basierte Plattform, die es erlaubt, Dienste bereitzustellen, aufzufinden und gegenseitig zu benutzen. Ein Service wird als abstrakte Funktionalität definiert, die in jedem Fall als Web Service realisiert werden. SOA wird als Sammlung von minimal notwendigen Basisfunktionen definiert. Dies umfasst die Möglichkeiten der Weiterverwendung bestehender Systeme, die lose Kopplung von standardisierten und gekapselten Diensten, den Einsatz eines semantischen Frameworks und die Verwendung so genannter Business Events [Seymour 2005]. Das SOA-Modell von Oracle nennt sich Oracle Business Integration Architecture. Diese Architektur wird mittels Oracle Fusion Middleware als Technologieplattform realisiert. Oracle erweitert SOA mit Mechanismen zur Erzeugung und Verarbeitung von Events und nennt diese Event Driven Architecture (EDA).

### 2.4.1 Oracle Business Integration

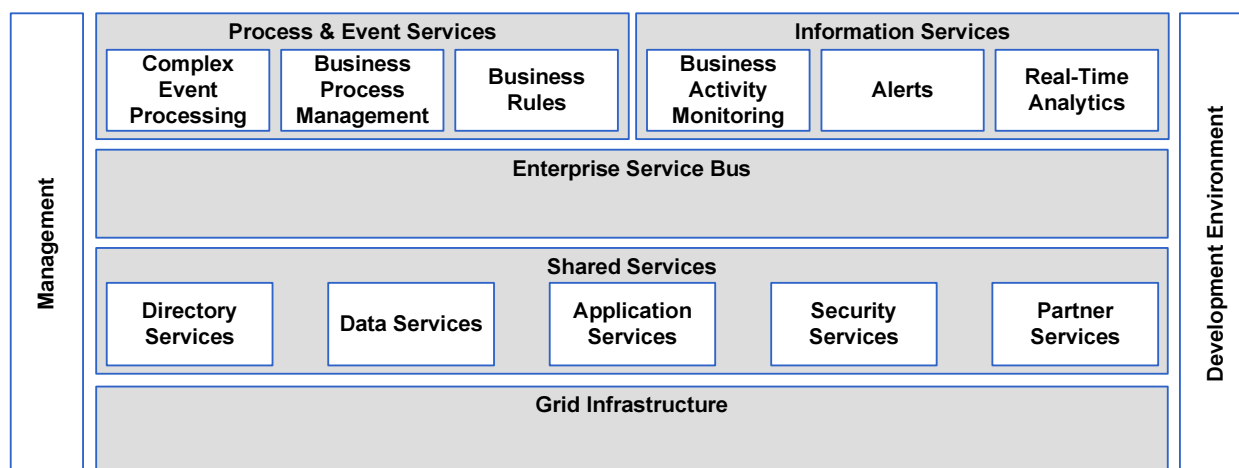


Abbildung 18: Oracle Business Integration Architecture

Die Oracle-SOA-Architektur ist in fünf logische Schichten aufgeteilt, die jede für sich eine Zuordnung von Services in verschiedene Kategorien gestattet [Oracle 2006]:

- **Process & Event Services:** Sowohl das Business Process Management als auch das Regelwerk zur Steuerung des Ablaufs von Geschäftsprozessen sowie die Verarbeitung von angereicherten Meldungen aus allen anderen Schichten ist Bestandteil dieser Schicht.
- **Information Services:** Die Analyse des Gesamtsystems durch Alerts, Monitoring und andere Mittel werden in dieser Schicht geregelt.
- **Enterprise Service Bus:** der zentrale Kommunikations-Mechanismus eines Gesamtsystems aus Sicht der Oracle-Architektur.
- **Shared Services:** umfassen alle Services in einer Schicht. Diese Dienste sind gruppiert nach Directory Services (User Verwaltung), Data Services (Bereitstellung von Daten), Application Services (Funktionale Dienste), Security Services und Partner Services.
- **Grid Infrastructure:** Dieser Layer umfasst sämtliche Hardware-Komponenten, die man für die Bereitstellung eines Systems benötigt.

### 2.4.2 Oracle Fusion

Oracle Fusion besteht einerseits aus der Technologieplattform Oracle Fusion Middleware und den Oracle Fusion Applications. Mit Oracle Fusion Applications sind diejenigen Systeme gemeint, die mittels Fusion Middleware integriert und vom Hersteller heute als eigenständige Produkte geführt werden (Oracle Applications, JD Edwards, Siebel, People-Soft, Retex, etc.).

Oracle Fusion Middleware ist die Technologieplattform von Oracle für die Realisierung von SOA, die auf einer Reihe von zentralen Designprinzipien basiert [Demarest, Heller 2006]. Anbei die wichtigsten Prinzipien:

- Modell Driven Software Development (MDSD [Mellor et Al. 2003]) wird mit graphischen Instrumenten zur Modellierung von Prozessen, Regeln, Ereignisabläufen und Datentransformationen, mit deklarativen Programmierungsumgebungen und Tools für das Lifecycle und Change Management unterstützt.
- Die Plattform erlaubt die Darstellung von Funktionen als standardisierte Web Services mit Mechanismen, die skalierbar, robust und sicher sind. Sie unterstützt außerdem Standards wie RosettaNet [Damodaran 2004], HL7 [Stevens 2003], HR-XML [Allen, Bartkus 2007], UN/CEFACT [Hofreiter et Al. 2004], um horizontale und vertikale Integration einzubeziehen.
- Ein globales Modell für die Erzeugung und Abarbeitung von Ereignissen wird lanciert, die so genannte Event Driven Architecture (EDA).
- Zentraler Schwerpunkt ist die Bereitstellung von operativen, analytischen und kollaborativen Informationen. Mechanismen für die Verwaltung von Meta-Daten, die Erzeugung standardisierter Datenmodelle und die Stammdatenverwaltung sind integraler Bestandteil von Oracle Fusion Middleware.

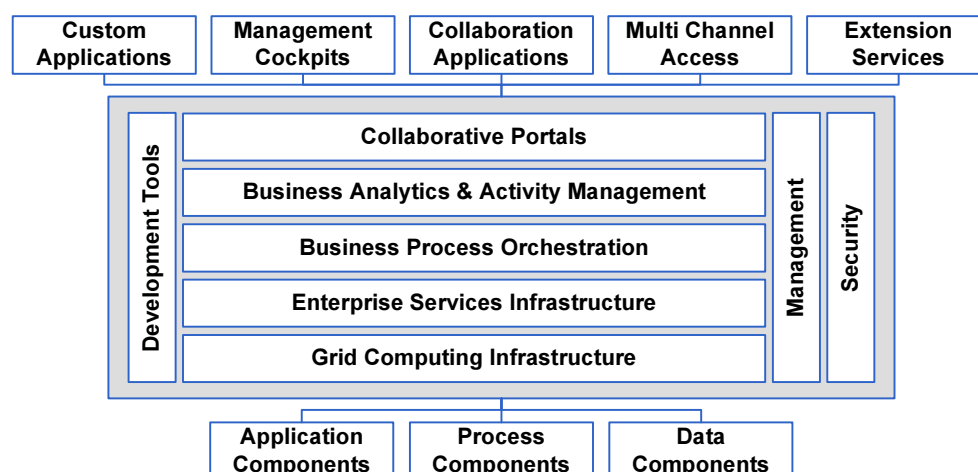


Abbildung 19: Die ursprüngliche Architektur der Oracle Fusion Middleware (nach [Kurian 2006])

Die Architektur der Oracle Fusion Middleware verbindet die Basiskomponenten bestehender Anwendungen (Application Components), Prozesse (Process Components) und Daten (Data Components) mit den Lösungen, die basierend auf Fusion Middleware realisiert werden (Custom Applications, Management Cockpits, Collaboration Applications, Multi Channel Access und Extension Services).

- **Development Tools:** Oracle stellt eine Vielzahl von Entwicklungsumgebungen für die Programmierung, die Modellierung von Workflows, Geschäftsprozessen, Regeln, Daten-transformationen und verschiedene deklarative Sprachen zur Verfügung.
- **Collaborative Portals:** Der Collaborative Portals Layer wird auch Collaborative Enterprise Portals genannt und beinhaltet Services für den so genannten „Information Worker“, wie beispielsweise Groupware, Presence Services, Portlets Frameworks und Search-Mechanismen.
- **Business Analytics & Activity Management:** In dieser Schicht werden Instrumente zur Kontrolle und Überwachung der Prozesse, Datenflüsse und von anderen Aktivitäten der gesamten Infrastruktur bereitgestellt. Dabei realisiert Oracle „Business Activity Monitoring“ (BAM), ein Konzept aus dem Bereich Business Intelligence zum Abgleich der operativen Tätigkeiten eines Unternehmens mit den Planzahlen.
- **Business Process Orchestration:** Die Orchestrierung umfasst sowohl die Ausführung von Geschäftsprozessen mittels BPEL als auch die Steuerung von Business-Logik durch Business Rules.
- **Enterprise Services Infrastructure:** Diese Schicht ist für die Konnektivität aller Services auf Unternehmens-ebene und die Bereitstellung von Integrationsmechanismen vorgesehen. Der Enterprise Service Bus und die so genannten Data Hubs für den Abgleich verschiedener Datenquellen sind integrale Bestandteile.
- **Grid Computing Infrastructure:** Die beiden zentralen Infrastrukturkomponenten sind die Oracle Datenbank und der Oracle Application Server, der sich sowohl als Middle Tier Server als auch als Integration Server oder Development Server einsetzen lässt.
- **Service Management:** Das Service Management beinhaltet Instrument zur Verwaltung von Diensten.
- **Identity Management:** Diesem Bereich sind alle sicherheits- und Compliance-relevanten Komponenten zugeordnet. Dazu gehören das Identity & Access Management, die PKI Infrastruktur, Audit & Compliance Software etc.

Die Abbildung der Oracle-Fusion-Architektur auf die verschiedenen Produkte des Herstellers beinhaltet eine Vielzahl von verschiedenen Produkten und Produktfamilien, wie beispielsweise Oracle Portal, Data Integrator, Enterprise Messaging Service, Identity & Access Management etc. Für die Umsetzung einer SOA ist jedoch die Produktfamilie Oracle SOA Suite zentral.

### 2.4.3 Die SOA Suite

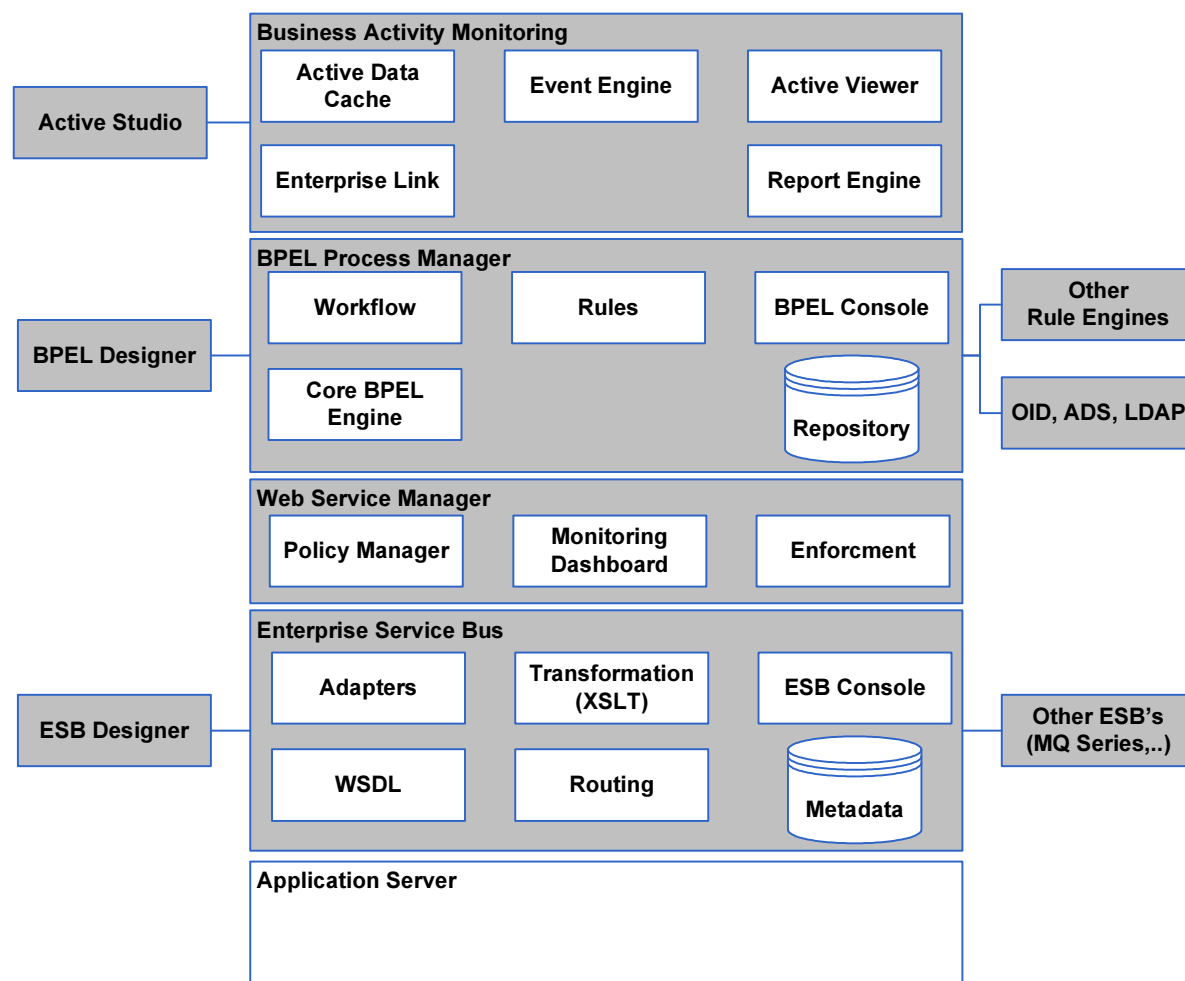


Abbildung 20: Die ursprünglichen Komponenten der Oracle SOA-Suite

Die Oracle SOA-Suite ist die Produktfamilie, bestehend aus dem Enterprise Service Bus, dem Web Service Manager, einer Process Engine und einer Monitoring-Komponente. Der Application Server ist aus Sicht von Oracle eine Infrastruktur-Komponente, auch wenn er für den Betrieb jedes der Produkte unverzichtbar bleibt. Die SOA-Suite hat alleine drei Entwicklungswerkzeuge, was die Umsetzung und die Konfiguration einfacher Umgebungen erschwert.

### 2.4.4 Bedeutung

Das Angebot von Oracle weist zwei Besonderheiten auf. Oracle weist in seinem SOA-Stack explizit Data Services aus, trennt also bewusst Funktionalität und Daten durch eine Abgrenzung verschiedener Servicetypen. Außerdem hat Oracle mit Business Activity Monitoring (BAM) und Real Time Analytics klassische BI-Funktionen in den SOA-Stack integriert. Oracle hat mit Fusion Middleware eine vielversprechende Komponente in seinem SOA-Stack. Im Gegensatz zu den anderen Herstellern ist Oracle gezwungen, SOA als Basis der eigenen Produkte zu verwenden, da es sonst kaum möglich sein wird, Siebel-Module mit Peoplesoft, Retex, JD Edwards oder anderen Modulen aus der großen Oracle Produktpalette zu einer Lösung zu kombinieren.

## 2.5 Microsoft

Microsoft versteht SOA als Architektur einer Applikations-Plattform. Diese Architektur ist das Resultat bestimmter Grundsätze, Praktiken und Frameworks, die es erlauben, normierte Services bereitzustellen. Kernelemente der SOA von Microsoft sind die Darstellung von Geschäftsanwendungen (Expose), die Bereitstellung von Services und Prozessen (Compose) und die Interaktion mit den bereitgestellten Anwendungen (Consume) [Sondermann 2007]. Der

Service selbst ist ein Konzept, das durch das Verhältnis zwischen dem Service Provider (Servicegeber) und Service Consumer (Servicenehmer) definiert ist und nicht zwingend mit Web Services umgesetzt werden muss, um im Rahmen einer SOA verwendet zu werden [Sprott, Wilkes 2004]. Zentral ist vielmehr die Integrierbarkeit und Interoperabilität eines Dienstes. Microsoft legt sehr großen Wert auf die Besonderheiten des Entwicklungsprozesses für die Bereitstellung einer SOA und der entsprechenden Services, der durch spezielle Modellierungstechniken, Frameworks und Composite Applications wie beispielsweise die Office Business Applications (OBA) oder Microsoft Dynamics unterstützt wird.

Die Umsetzung des SOA-Modells erfolgt auf drei Ebenen: einerseits durch eine Application-Plattform, die wiederum durch verschiedene Microsoft-Produkte und Produktfamilien implementiert werden kann. Andererseits durch die Microsoft Application Architecture, die den Rahmen für die Entwicklung von Software, basierend auf dem .NET Framework, vorgibt. Und schließlich durch ein Regelwerk für die Abbildung von Anforderungen in Systeme (Business Capabilities), für die Modellierung dieser Systeme und deren Komponenten (System Definition Model) und den Bau von Services (Software Factories und Domain Specific Language).

### 2.5.1 Die Service-Architektur der Application Platform

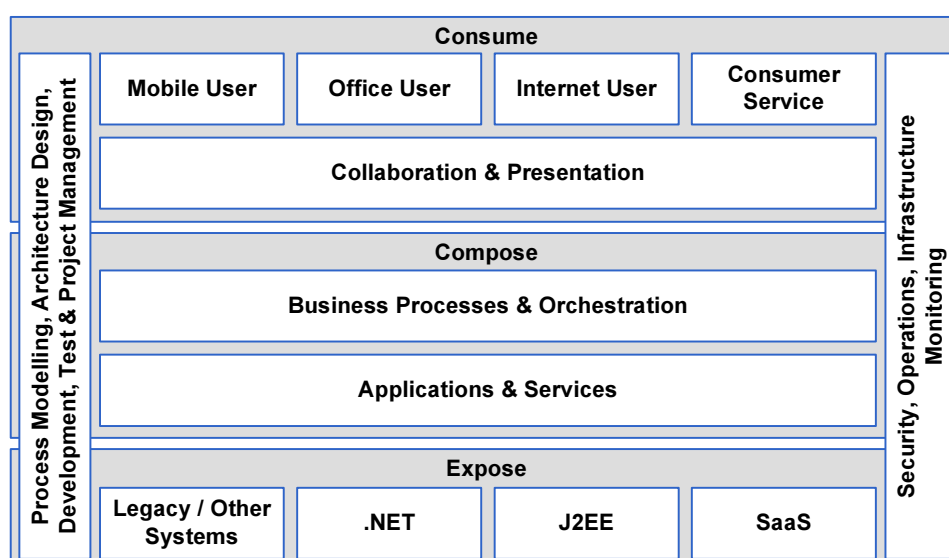


Abbildung 21: Service-Architektur der Application Platform

Die SOA-Architektur von Microsoft ist in drei logische Ebenen – Expose, Compose und Consume – aufgeteilt:

- **Expose:** Geschäftsanwendungen werden als technologie neutrale Dienste sichtbar gemacht. Darunter sind nicht einfach Web Services zu verstehen, sondern Services, die im Sinne des Users relevante Funktionalität darstellen, die abstrahiert wieder verwendbar, mit einer Schnittstelle versehen und formal beschrieben sind.
- **Compose:** Die Bereitstellung von standardisierten und „consumable“ Services und Geschäftsprozessen erfolgt durch Web Services, die so genannten Business Components (Komponenten, die Logik darstellen), Business Entities (Komponenten, die Daten darstellen) und durch Business Workflows. Business Workflows steuern den Ablauf der Anwendungslogik.
- **Consume:** Die Interaktion mit den bereitgestellten Anwendungen erfolgt durch so genannte Composite Applications und verwendet Collaboration und Presentation-Mechanismen, um den Zugriff auf Funktionalität sowie auf strukturierte und unstrukturierte Daten zu erleichtern. Auf dieser Ebene kommen Workflows zum Einsatz, die den Ablauf der Interaktion mit dem User steuern.

## 2.5.2 Application Architecture

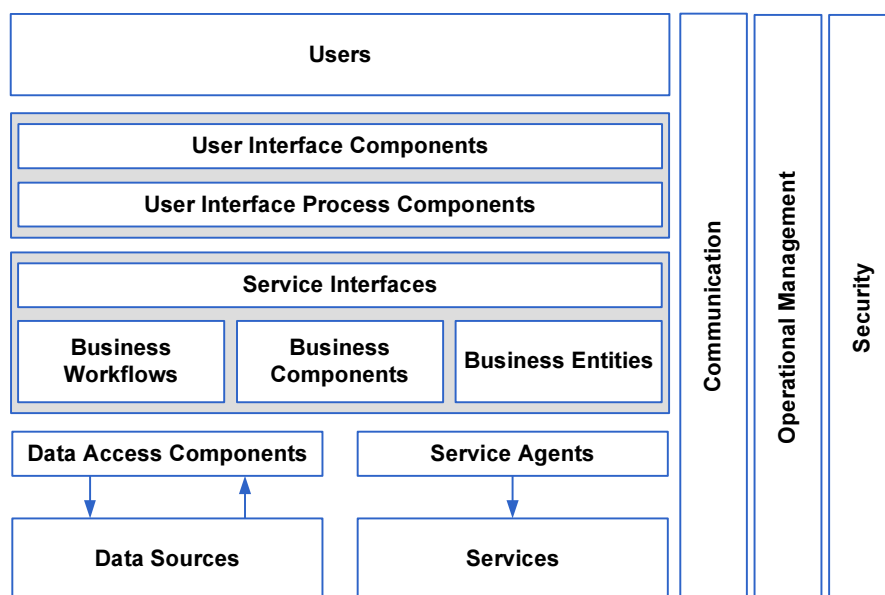


Abbildung 22: Microsoft Application Architecture

Die Microsoft Application Architecture besteht aus einer Reihe von Komponenten, die als logische Teile einer Gesamtlösung angesehen werden [Platt 2002]:

- **User Interface Components:** Bestandteile, die man zur Bereitstellung eines GUI verwendet. Darunter werden WinForms, Controls und ASP.NET-Seiten zusammengefasst.
- **User Interface Process Components:** Mechanismen zur Steuerung und Orchestrierung von Abläufen auf User-Interaction-Ebene.
- **Service Interfaces:** Um Business-Logik als Service zugänglich zu gestalten, sind Service Interfaces, die den „Communication Contract“ (Formate, Protokolle, Sicherheit, Exceptions etc.) einhalten, einzusetzen.
- **Business Workflows:** Mechanismen zur Steuerung und Orchestrierung von Abläufen auf Ebene eines Geschäftsfalles.
- **Business Components:** Die Business Components realisieren die Business-Logik einer Applikation.
- **Business Entity Components:** Diese Komponenten sind für den Austausch von Daten zwischen verschiedenen Komponenten zuständig. Sie können als DataSets, DataReaders, XML Streams oder „Custom Objects“ realisiert werden.
- **Service Agents:** Wird eine externe Ressource verwendet, so ist der Service Agent für die semantische Umsetzung und die korrekte Abfolge der Interaktion mit der entsprechenden Ressource zuständig.
- **Data Access Components:** Diese Komponenten abstrahieren die erforderliche Logik, um auf Daten zuzugreifen.

## 2.5.3 SOA-Grundsätze, Praktiken und Frameworks

Microsoft versteht SOA als Resultat eines wohl formulierten Entwicklungsprozesses, der auf einer Reihe von Grundsätzen, Praktiken und Frameworks basiert. Die wichtigsten Elemente zur Realisierung von SOA-Applikations-Plattformen sind die Art und Weise der Umsetzung von Anforderungen in Systeme (Business Capabilities), der Modellierung dieser Systeme und deren Komponenten (System Definition Model) und des Baus von Services (Software Factories).

Business Capabilities beschreiben eine geschäftliche Tätigkeit von außen als „Black Box“. Ziel dieser Beschreibung ist eine Trennung der Eigenschaften einer geschäftlichen Tätigkeit von den Eigenschaften eines betrieblichen Informationssystems, um die Einschränkungen der Abbildung durch die direkte Modellierung von Geschäftsprozessen in IT-Systeme zu überwinden [Homann 2006]. Business Capabilities sind ein Instrument auf der Ebene des Requirements Engineering, welches das traditionelle Business Process Engineering (wie wird ein Geschäft abgewickelt) erweitert. Eine Beschreibung, was ein bestimmtes Geschäft überhaupt ausmacht und wie genau verschiedene Aspekte einer geschäftlichen Tätigkeit zusammenhängen, werden zusätzlich erzeugt.

Das System Definition Model (SDM) soll die Vereinfachung und Automatisierung des Entwurfs, der Bereitstellung und der Ausführung unterstützen. Es modelliert Systeme auf abstrakter Ebene und ist ein Instrument, um Services als spezielle Implementierung eines Systems toolgestützt zu definieren und zu konfigurieren.

Eine Software Factory im weitesten Sinne ist eine vorgefertigte Lösung für eine bestimmte Klasse von Problemstellungen mit dem Ziel, den Umfang an Spezifika für die Realisierung eines bestimmten Services zu reduzieren [Santos 2006]. Zu diesem Zweck wird mit so genannten Software Factory Templates gearbeitet, die so konfiguriert werden können, dass sie den größten Teil einer Lösung bereits enthalten.

#### **2.5.4 Bedeutung**

Eine Besonderheit des SOA-Modells von Microsoft liegt in der Presentation-Ebene. Mit Office-Applikationen als User Interfaces für auf SOA basierende Lösungen und der damit verbundenen Idee, Anwendungen, basierend auf Services, einfach zusammenstellen zu können, erweitert Microsoft den Wirkungsbereich des Desktops. Des Weiteren versucht Microsoft, die Vorteile von SOA in die Entwicklung abzubilden. Dies geschieht durch die nahtlose Integration der Web Services-Technologien in die Entwicklungsumgebung und durch die konzeptionelle Realisierung von Software Factories und Domain Specific Languages. Diese Bestrebungen gehen von der Idee aus, mittels SOA-Anwendungen aus vor-gefertigten Komponenten (Services) sehr schnell und produktiv zusammenzustellen. Die Aufteilung der Office-Produktfamilie in eine Vielzahl von Serverprodukten und die Erweiterung von SharePoint als Portal und Collaboration Environment unterstützen diese Um-setzung.



# SOA Komponenten

## 3.1 Einleitung

Der Bau eines konkreten Systems, also die Lösung eines anstehenden Problems, ist der Ausgangspunkt für die vorliegende Auswahl der Bestandteile einer SOA. Nur wenn der Bestandteil in einer konkreten Situation als nützliche Komponente eingesetzt werden kann und gleichzeitig einen wichtigen Teil der Gesamtarchitektur darstellt, lohnt sich der Aufwand für die Evaluation.

Ausgehend vom pragmatischen und herstellerunabhängigen SOA-Modell, können auf je-der Ebene der Architektur die für ein Unternehmen wichtigen Bestandteile isoliert werden.

Das SOA-Architekturmodell sieht eine Teilung zwischen Applikationen, Integrationsmechanismen, Diensten und der Orchestrierung vor. Die Applikationen (Applications) beinhalten bestehende oder auch neue Systeme und logische Datenspeicher. Die Dienste (Services) stellen Schnittstellen zu den einzelnen Anwendungen dar. Die Orchestrierung (Orchestration) dient zur Steuerung von Abläufen, die mittels Einbezug mehrerer Dienste durchgeführt werden. Die Kommunikation zwischen verschiedenen Diensten untereinander sowie zwischen Diensten und deren Implementierung (Applikation und Daten) erfolgt über eine Integrationsarchitektur (Integration Architecture).

Das SOA-Architekturmodell ist nicht als hierarchisch geschichtetes Modell zu verstehen. So ist die Integrationsarchitektur sowohl zwischen Services und Anwendung als auch zwischen Services und der Orchestrierung auch zwischen der Präsentationsebene und allen anderen Ebenen denkbar. Auch können Services direkt andere Services aufrufen, und Workflows, falls sie als ausführbare Prozesse in BPEL oder BPMN realisiert wurden, sind selbst wiederum Services. Dies bedeutet, dass sie sowohl andere Workflows als auch direkt Services aufrufen können. Ebenso ist denkbar, dass Anwendungen Services aufrufen und Datenbanken über Trigger Funktionen Workflows aufrufen.

Die Orchestrierung umfasst sowohl grobgranulare Geschäftsprozesse, die als Workflows realisiert werden, als auch feingranulare Geschäftsregeln. So ist beispielsweise der Ablauf einer Flugbuchung durch ein Unternehmen (Eingabe der Angaben, Prüfung der Platzverfügbarkeit, Vorreservation, Bestätigung, Endgültige Reservation) ein grobgranularer Prozess, während die Rabattkonditionen für das Unternehmen als feingranulare Geschäftsregeln formuliert werden. Das Modell geht davon aus, dass für die Modellierung von Prozessen und für die Modellierung von Regeln entsprechende graphische Tools eingesetzt werden, sodass sich Änderungen sehr einfach und schnell durchführen lassen. Sowohl die Regeln als auch die Prozesse müssen mit formalen Methoden definierbar sein, sodass der Output einer graphischen Modellierung als Input für die entsprechende Process oder Rule Engine benutzt werden kann.

### 3.2 SOA Komponenten auf einen Blick

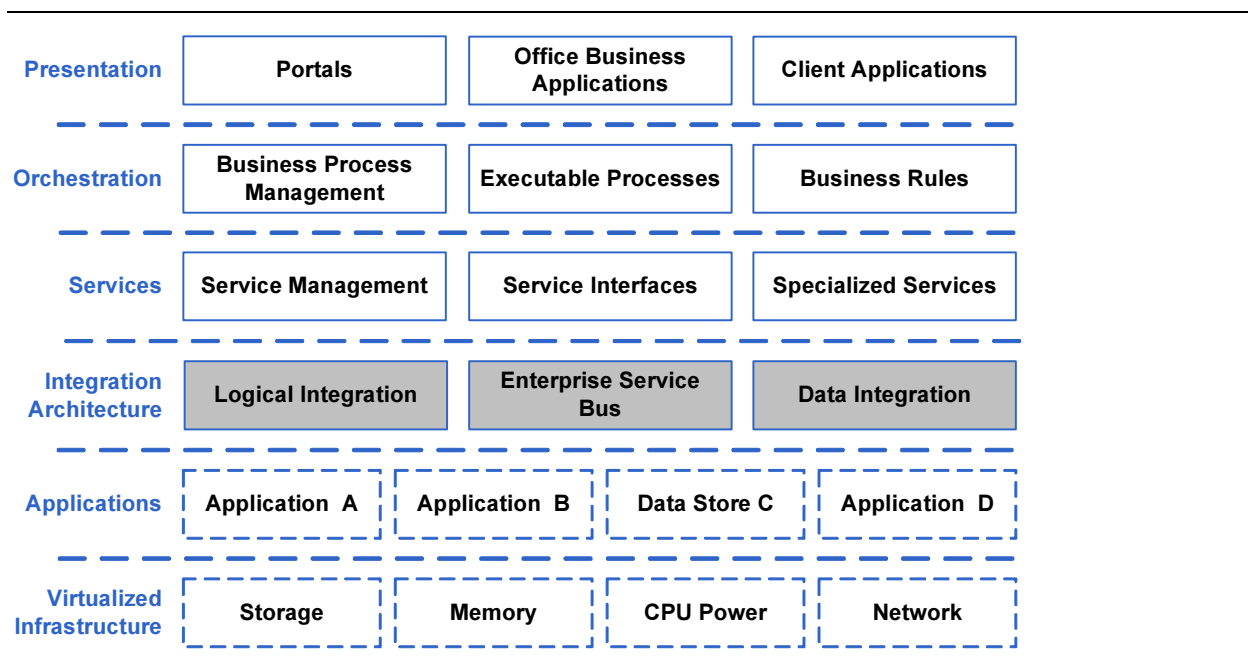


Abbildung 23: Die typischen Komponenten einer SOA

Die einzelnen Bestandteile befinden sich ausschließlich auf den Ebenen Presentation, Orchestration, Services und Integration Architecture:

- **Presentation-Ebene:** Das User Interface einer mittels SOA realisierten Anwendung wird entweder als Portal, als Office Application oder als Client Applications realisiert. Client Applications können sowohl als Rich Client oder auch als Web Client umgesetzt werden.
- **Orchestration-Ebene:** Die Orchestration-Ebene bildet Geschäftsprozesse und Geschäftsregeln in einer Service Oriented Architecture ab. Sie ist für den dynamischen Bereich der Business-Logik einer auf SOA basierenden Anwendung zuständig.
- **Service-Ebene:** In dieser Ebene sind die Mechanismen zur Verwaltung von Diensten, die standardisierten Serviceschnittstellen sowie die spezialisierten Dienste zu finden.
- **Integration Architecture-Ebene:** Die Infrastruktur zur Verknüpfung der verschiedenen Dienste und zur Verbindung von Diensten mit bestehenden Anwendungen oder Daten-banken sowie zur Koppelung von Services mit den Bestandteilen der Presentation Ebene.

In den verschiedenen SOA-Modellen der Hersteller und der Standard- und Normierungsgremien sind zusätzliche Bestandteile vorgesehen, wie beispielsweise Policy Manager, Monitoring, Activity Management, Deployment, IT Service Management, IT Governance Management und andere. Sie sind jedoch für die Funktionsweise einer auf SOA basierenden Lösung nicht zentral und überladen oft eine mögliche Einführung von SOA mit unnötiger Komplexität.

Spezielles Kopfzerbrechen bei der Einführung einer SOA bereitet der Bereich Security. Leider sind die Standards im Bereich Web Service Security sehr umfangreich und unternehmensintern wenig nützlich. Der Ansatz, Security als spezialisierten Service – beispielsweise als IAM (Identity und Access Management Service) – bereitzustellen ist vielversprechend, auch wenn in vielen Fällen die Ausbreitung eines solchen Service an den anwendungsspezifisch geltenden Sicherheitsmechanismen scheitern wird. Bei einer Weiterverwendung bestehender Systeme ist der Ausbau von Authentisierung und Autorisierung aus der Anwendungslogik oftmals viel zu teuer. Eine andere Lösung ist der Einsatz von spezialisierter Hardware, die Mechanismen wie beispielsweise die Verschlüsselung von SOAP Messages ausführt. Dies bedeutet jedoch, dass Security als Infrastrukturkomponente bereitgestellt wird und keinen direkten Einfluss auf den Aufbau und die Bestandteile der anderen Ebenen hat.

In den verschiedenen SOA-Modellen der Hersteller sowie der Standard- und Normierungsgremien sind zusätzliche Bestandteile vorgesehen, wie beispielsweise Policy Manager, Monitoring, Activity Management, Deployment, IT Service Management, IT Governance Management und andere. Sie sind jedoch für die Funktionsweise einer auf

SOA basieren-den Lösung nicht zentral und überladen oft eine mögliche Einführung von SOA mit unnötiger Komplexität.

### 3.3 Presentation Ebene

Das User Interface einer mittels SOA realisierten Anwendung unterscheidet sich nicht grundlegend von einem User Interface eines beliebigen Systems, obwohl die SOA Blueprints der meisten Hersteller Portlets als User Interface vorsehen. Dennoch bietet SOA durch die Bereitstellung von Services oder ganzen Prozessabläufen als Services eine Reihe von Möglichkeiten, große Teile der Logik einer Anwendung zu separieren und auf mehreren Ebenen von der Präsentationsebene aus auf verschiedene Dienste zuzugreifen.

#### 3.3.1 Portale

Portale und Portlets erlauben die Darstellung von Funktionalität und die Interaktion über standardisierte Erweiterungen von WSDL, den WSRP (Web Services for Remote Portlets). Darüber hinaus sind sie geeignet, im Rahmen eines ausführbaren Geschäftsprozesses die nicht sehr zahlreichen „Human Tasks“ als Webseite in einem Portal darzustellen.

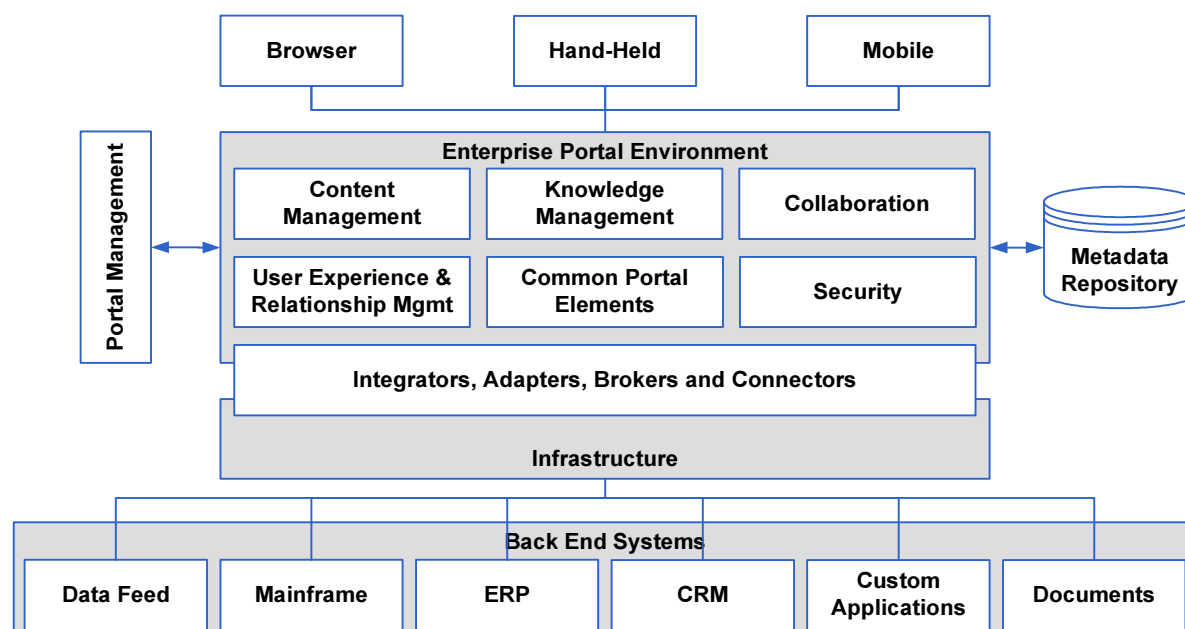


Abbildung 24: Enterprise Portal Architecture

Der klassische Ansatz für die Architektur eines Enterprise Portals, geht davon aus, dass ein Portal ein elektronischer Arbeitsplatz für Angestellte eines Unternehmens ist. Das Enterprise Portal stellt eine unternehmensweite Integration von Anwendungen dar, wobei es die Vorteile der Internet-Technologie nutzt [Hazra 2002]. Die zentralen Funktionen eines Unternehmensportals sind die Verwaltung der Portalinhalte (Content Management) und die Darstellung des Fachwissens eines Unternehmens (Knowledge Management), die Plattform für die Zusammenarbeit (Collaboration), die technischen Hilfsmittel zur Darstellung von Informationen (User Experience & Relationship Management) und die Sicherheit (Security).

- **Content Management:** Die Generierung, Zusammenführung, Personalisierung und Presentation von Inhalten.
- **Knowledge Management:** Die Kategorisierung von Informationen auf der Basis von User Profilen und die Speicherung dieser Kategorisierung als Meta Daten. Die Generierung von Reports und die Versionierung von Informationen.
- **Collaboration:** Mail, Chat und Videokonferenzen sowie die Möglichkeit, spezifische Ereignisse an bestimmte Nutzergruppen zu senden.
- **User Experience & Relationship Management:** Suchfunktionen, Navigationshilfen, gerätespezifische Visualisierung und die „Internationalisierung“ von Inhalten.

- **Security: User Management**, Single Sign On, die Identifikation und Authentisierung von Nutzern sowie die Validierung von speziellen Privilegien und die Erzeugung von Zugriffsprotokollen

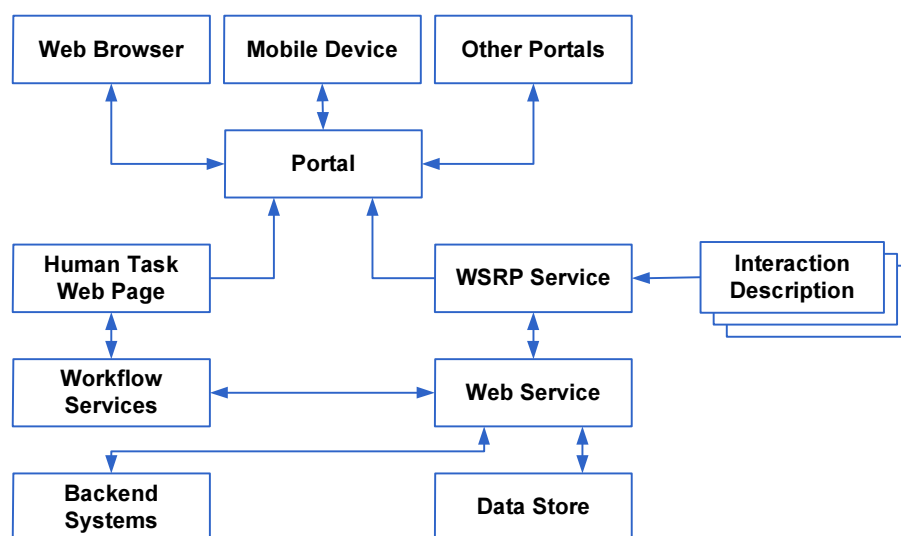


Abbildung 25: Portal als SOA GUI

SOA erlaubt die standardisierte Integration von Services über WSRP (Web Service for Remote Portlet) und erleichtert damit den Bau von Unternehmensportalen erheblich. Dies ist jedoch lediglich eine weitere Verwendung verschiedener Services, die zwar Kosten spart, mit der Präsentation von auf SOA basierenden Anwendungen jedoch nur indirekt zu tun hat. Ein anderer Fall sind Anwendungen, die nur relativ wenig Interaktivität benötigen. Typischerweise ist dies bei ausführbaren Geschäftsprozessen der Fall, bei denen eine Bestätigung durch eine autorisierte Person notwendig ist, bevor der Prozess weiterlaufen kann. Solche „Human Tasks“ sind einfache einzelne Webseiten, die sich sehr wohl im Rahmen eines Portals darstellen lassen. In beiden Fällen lohnt sich eine Darstellung in einem Unternehmensportal nur dann, wenn ein solches System bereits besteht oder auch für andere Zwecke eingesetzt werden soll. Für eine einzelne auf SOA basierende Anwendung lohnt es sich kaum. Eine einzelne Web-Anwendung reicht da völlig aus.

### 3.3.1.1 Bedeutung

Die Realisierung eines Portals als User Interface für eine auf SOA basierende Anwendung wird von vielen Herstellern vorgesehen. Die Bereitstellung einer Portalinfrastruktur nur für eine einzige auf SOA basierende Anwendung lohnt jedoch selten. Auch sollten Sie bei der Einführung einer SOA in Ihrem Unternehmen diese nicht noch mit einer eigens dafür aufgebauten Portaltechnologie überladen. Zudem ist ein Portal nur für solche Anwendungen das richtige User Interface, die relativ wenig Interaktivität, wie beispielsweise die Bestätigung eines Kreditantrages durch eine Fachperson, benötigen. Die Sache sieht jedoch anders aus, wenn Sie in Ihrem Unternehmen bereits ein Portal einsetzen. Dann können sie diese Technologie relativ einfach nutzen, um Ihre SOA-Anwendung darzustellen. Und sie ist erst noch an einem zentralen Ort im Unternehmen sichtbar. Falls Sie ein Portal einsetzen, empfiehlt es sich, die entsprechenden Standards zu nutzen, um aus bestehenden Web Services WSRP-fähige Dienste zu machen.

### 3.3.2 Office Business Applications

Office Business Applications: Office-Anwendungen wie beispielsweise Word, Excel, PowerPoint und Staroffice werden als User Interface für ein auf SOA basiertes System verwendet. Zu diesem Zweck wird einerseits die Anwendung selbst als so genannte Composite Application flexibler aufgebaut, andererseits wird das Dokumentformat so erweitert, dass die Beschreibung von Interaktionen eingebettet werden kann.

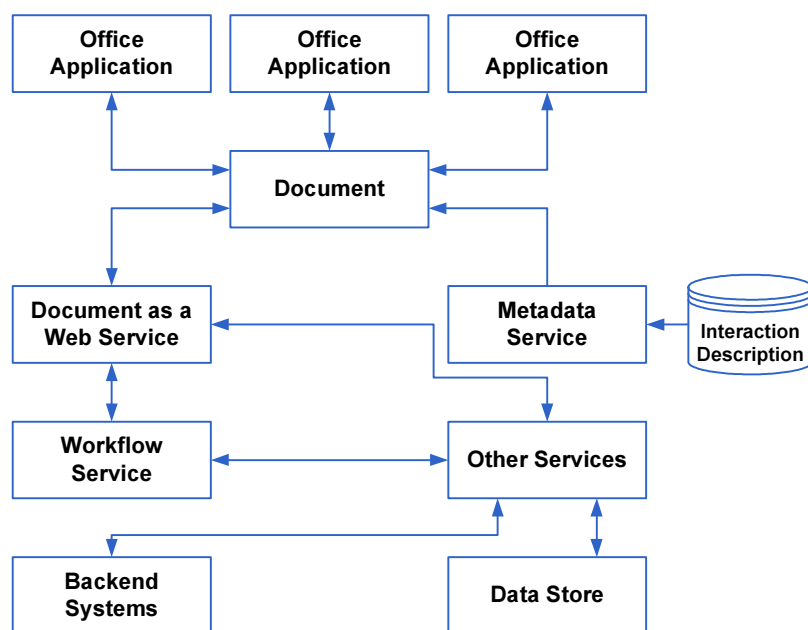


Abbildung 26: Prinzip der SOA-Präsentation mittels Office-Anwendungen

Für den Einsatz von Office-Anwendungen als Präsentationsebene für SOA spricht die Tatsache, dass Word, Excel, PowerPoint und Staroffice die am häufigsten benutzte Software ist. Diese Programme werden als User Interface für ein SOA-basiertes System verwendet. Zu diesem Zweck wird einerseits die Anwendung selbst als so genannte Composite Application flexibler aufgebaut, andererseits wird das Dokumentformat so erweitert, dass die Beschreibung von Interaktionen (actions) eingebettet werden kann [Banerjee 2007].

Eine Konsequenz dieser Konstruktionsweise ist die Betrachtungsweise eines Dokuments als Web Service. Die Daten des Web Service sind der Dokumentinhalt, die Methoden des Web Service sind die als Metadaten gehaltenen Interaktionsbeschreibungen. Diese Art von Web Service ist kaum verbreitet, entspricht jedoch der Funktionsweise des Internets als globale Informations-Architektur und könnte aus diesem Grund in Zukunft eine große Rolle spielen. Jede einzelne Webseite ist im Prinzip nichts anderes als ein Dokument, das als Service auf dem Web dargestellt wird. Die Referenz auf das Dokument ist die URL des Dokuments. Mit den Standard-HTTP-Methoden (GET, POST, PUT und DELETE) lassen sich Parameter mit dem Dokument austauschen. Die Funktionalität des Service ist durch die Struktur des Dokuments selbst gegeben. Die Service Implementation ist durch den Inhalt des Dokuments definiert. So wird beispielsweise ein täglicher Report aller Verkaufszahlen eines Unternehmens in seiner Realisierung einen oder mehrere Aufrufe der operativen Systeme, die die Zahlen liefern sowie eine Reihe von Kalkulationen beinhalten, umfassen. Die Schnittstelle zu einem Dokument als Web Service wird auch Content-Based Interface oder Resource Centric Interfaces genannt.

### 3.3.2.1 Bedeutung

Den Großteil unserer Arbeitszeit verbringen wir am Computer mit Office-Anwendungen. Die meisten Mitarbeitenden in Ihrem Unternehmen sind diese Anwendungen gewohnt. Bedenkt man nun, wie viele Einführungsprojekte in der Informatik große Probleme mit der Akzeptanz der Nutzer haben, ist die Vorstellung, die am häufigsten verwendeten Anwendungen als User Interface für betriebliche Informationssysteme zu nutzen, einfach bestechend. Bisher sind wir leider nie über die so genannte Office Integration hinausgekommen. Und auch diese bereiten große Probleme aufgrund der Freiheitsgrade, die Programme wie beispielsweise MS Excel erlauben. Alle Hersteller von Office-Software werden in nächster Zeit jedoch ihre Programme so gestalten, dass sie als Composite Applications sehr viel einfacher als User Interface eingesetzt werden können. Falls Sie also ein Rollout dieser neuen Versionen in nächster Zeit planen, lohnt es sich, den möglichen Einsatz als User Interface für Individuallösungen zu prüfen. Aber Vorsicht! Hier handelt es sich um eine neue Technologie, und der Einsatz eines Dokuments als Web Service ist gewohnungsbedürftig. Es lohnt sich aber auf jeden Fall, diese Technologie zu verfolgen.

### 3.3.3 Client Applications

Client Applications: Die Präsentation von SOA mit Client Applications entspricht der Präsentation der applikatorischen Logik in einer modernen geschichteten Anwendung. Die zwei wichtigen Technologien Java und .NET, die für die Realisierung von Anwendungen eingesetzt werden, unterstützen sowohl Rich Clients als auch Web Clients. Die Art und Weise der Präsentation unterscheidet sich zwar, die Art und Weise, wie mit SOA und den eingesetzten Diensten interagiert wird, jedoch nicht.

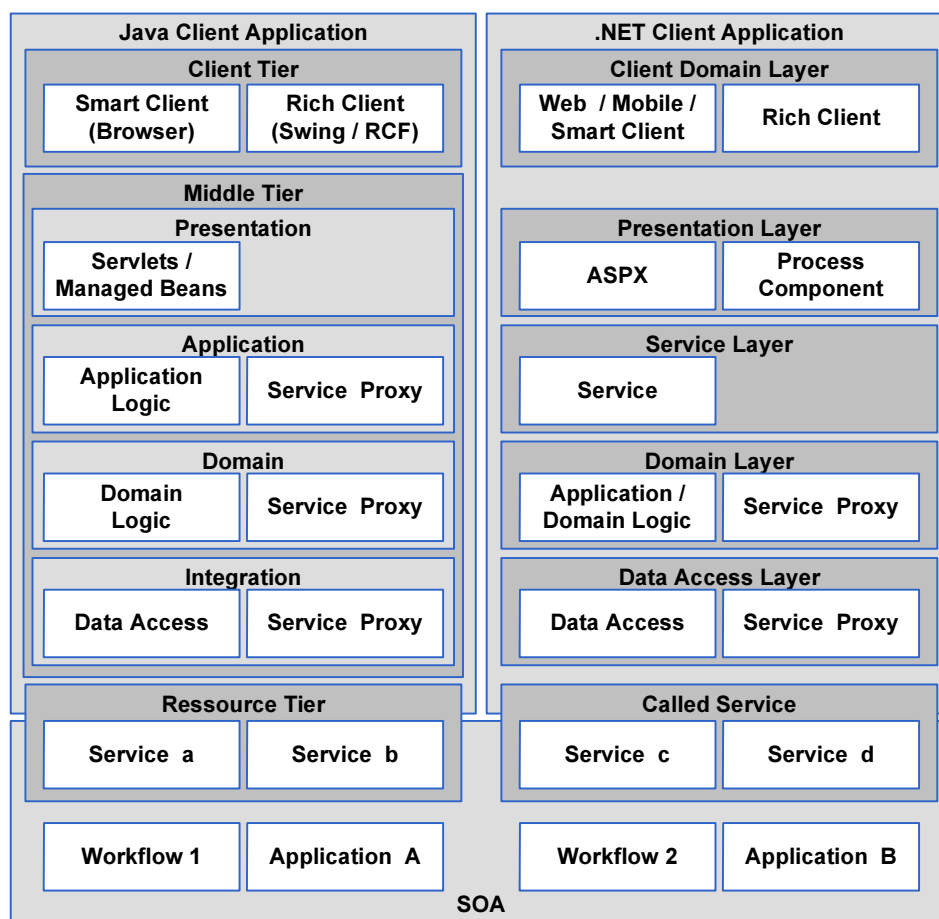


Abbildung 27: Szenarien für die Präsentation durch Client Applications

Es sind mehrere Szenarien für die Präsentation von Services in einer Client-Anwendung denkbar:

- **Java Application-Layer:** Dieser Layer enthält die applikations-spezifische Logik, die nicht auf dem Domain Layer realisiert worden ist. Die entsprechenden Dienste, die diese Logik umsetzen, werden über den Service Proxy auf dieser Ebene aufgerufen. Dies können sowohl statische Implementationen als auch Rule oder Process Services sein.
- **Java Domain-Layer:** Dieser Layer ist das Herzstück der geschichteten Architektur für die Java-Technologie. Sie enthält die Domain-Logik, diejenige Logik also, die Spezifika bezüglich des Anwendungsbereichs darstellt. Die entsprechenden Dienste, die diese Logik umsetzen, werden über den Service Proxy auf dieser Ebene aufgerufen. Dabei kann es sich sowohl um statische Implementationen wie um Rule oder Process Services handeln.
- **Java Integration-Layer:** Allgemeine technische Dienste zur Unterstützung des darüber liegenden Layers sind in dieser Schicht zu finden. Die meisten Dienste werden auf dieser Ebene integriert.
- **.NET Service-Layer:** Der Service Layer in einer .NET-Architektur stellt die verschiedenen Transportdienste zur Verfügung. Web Services stellen einen dieser Dienste dar.
- **.NET Domain-Layer:** Dieser Layer realisiert die Applikationslogik. Die Integration von Services auf dieser Ebene ist relativ selten. Voraussetzung ist, dass die Services sehr genau auf die Struktur der Anwendungslogik passen.
- **.NET Data Access Layer:** In den meisten Fällen wird ein Web Service auf dieser Ebene integriert.

### 3.3.3.1 Bedeutung

Der Einsatz von mit den modernen Technologien .NET und Java realisierten Client Applications ist die am weitesten verbreitete Variante eines User Interfaces für eine auf SOA basierende Anwendung. Die Integration von Services wird bereits von allen Entwicklungstools der entsprechenden Umgebungen sehr gut unterstützt. In Kürze wird es wohl kaum mehr neue Anwendungen geben, die nicht auf irgendeiner Ebene einen Service nutzen. Die Konsequenz davon ist, dass es keinerlei Einschränkung bezüglich der Interaktivität von Anwendungen, die auf SOA basieren, gibt. Ob Sie nun Ihre Anwendung als Rich Client, als Web Client, als Mobile Agent oder als Embedded System realisieren wollen, dem Einsatz von SOA und der systematischen Nutzung von standardisierten Services und ablauffähigen Prozessen steht nichts im Wege.

## 3.4 Orchestration Ebene

Die Orchestration-Ebene bildet Geschäftsprozesse und Geschäftsregeln in einer Service Oriented Architecture ab. Sie ist für den dynamischen Bereich der Business-Logik einer auf SOA basierenden Anwendung zuständig. Dabei kann zwischen der Dynamik auf Prozessebene und der Dynamik auf der Ebene einzelner Prozessschritte unterschieden werden. Geschäftsprozesse bilden Abläufe bestehend aus einer Sequenz von aufgerufenen Services, die gleichzeitig den Datenfluss zwischen diesen Services steuern, ab. Geschäftsregeln formulieren Steuerungsparameter für Funktionen und Abläufe einer Anwendung und kontrollieren das Verhalten der Geschäftsprozesse. Diese Regeln werden mit Rule Engines modelliert und zur Laufzeit ausgeführt. Sie repräsentieren den dynamischen Bereich der Business-Logik auf der Ebene des einzelnen Prozessschritts.

### 3.4.1 Business Process Management

Business Process Management (BPM) betrachtet ein Unternehmen als Ganzes, als unabhängige Sammlung von Geschäftsprozessen, die auf externe und interne Ereignisse reagiert. Die Geschäftstätigkeit wird als Ganzes modelliert und strukturiert. Man nennt diese Strukturierung auch Prozesslandschaft. Sie setzt sich aus Kern- und Unterstützungsprozessen sowie deren Makro-, Mikro- und Teilprozessen zusammen. Die Modellierung von Prozessen erfolgt mit verschiedenen Techniken. Die Prozesse werden mit geeigneten graphischen Werkzeugen definiert und getestet.

SOA stellt Standards und Technologien zur Verfügung, die eine direkte Umsetzung von graphisch modellierten Prozessen in ausführbaren Code erlauben. Allerdings sind die Mechanismen von SOA nicht für jede Art von Prozess geeignet. So stehen bis heute keine standardisierten Mechanismen zur Abbildung der menschlichen Tätigkeit im Rahmen eines Prozesses bereit. Dennoch schließt SOA die Kluft zwischen der Struktur der betrieblichen Informationssysteme und dem Geschäftsprozessmodell eines Unternehmens.

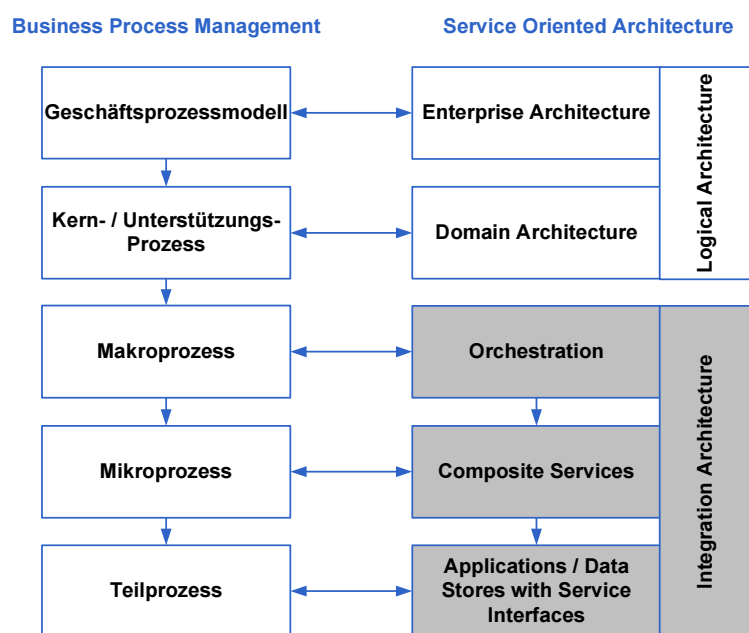


Abbildung 28: BPM und SOA

Werden BPM und SOA einander gegenübergestellt, so ergibt sich folgendes Bild; Auf der Ebene der Teilprozesse sind die einzelnen funktionalen Blöcke einer betrieblichen Informationstechnologie, also die Anwendungen und Daten zu finden. Sowohl Anwendungen wie auch die Datenbereiche sind mittels standardisierter Service-Schnittstellen als SOA-Basisdienste unternehmensweit sichtbar. Die Ebene der Mikroprozesse entspricht in einer SOA einer Gruppierung von Services (Composite Service), d.h. Services, die wiederum verschiedene Basisdienste aufrufen und als Ganzes eine erweiterte funktionale Einheit darstellen. Die Ebene der Makroprozesse entspricht der Orchestrierungsschicht, die Geschäftsprozesse in einer SOA als Sequenz von Dienstaufrufen modelliert. Die Steuerung der Anwendung erfolgt über den abgebildeten Prozessfluss und dessen einzelne Schritte, Entscheidungspunkte und Verzweigungen. Die Kombination von Web Services und Orchestrierung ist nur dann möglich, wenn man den Business Layer der Anwendungen eines Unternehmens in zwei Teile aufteilt: Funktion und Ablauf. Die einzelnen Funktionen einer Anwendung umfassen die statischen Komponenten, die eine bestimmte Business-Logik abbilden.

### 3.4.1.1 Bedeutung

Die Umsetzung von BPM wird durch SOA überhaupt erst strukturiert und standardisiert möglich. Ein Task eines Geschäftsprozesses ist nichts anderes als ein Aufruf einer Business-Funktion, die durch den entsprechenden Web Service standardisiert und unternehmensweit zugänglich ist. Der Prozess selbst wird als Workflow mittels einer ausführbaren Prozessmodellierungssprache modelliert und mittels einer Process Engine ausgeführt. Damit setzt SOA das bereits 1995 von der Workflow Management Coalition (WFMC) definierte „Workflow Reference Model“ mit den grundlegenden Standard-Web Services WSDL, SOAP und BPEL vollständig um. Jeder Hersteller einer SOA Suite hat die entsprechenden Modellierungstools und Process Engines im Programm. Die Umsetzung von BPM durch SOA erfolgt mittels einer Kombination aus Orchestrierung, Integration über einen ESB und der Verwendung von Web Services als Standardschnittstelle für jede Business-Logik und für jeden Datenbereich eines betrieblichen Informationssystems. Trotz fehlenden Standards und der damit verbundenen mangelnden Unterstützung der Ausführung von Prozess-Schritten durch den Menschen und den fehlenden Elementen für das Management ganzer Geschäftsprozessmodelle ist SOA das zentrale Instrument für die Realisierung von BPM.

### 3.4.2 Ausführbare Prozesse

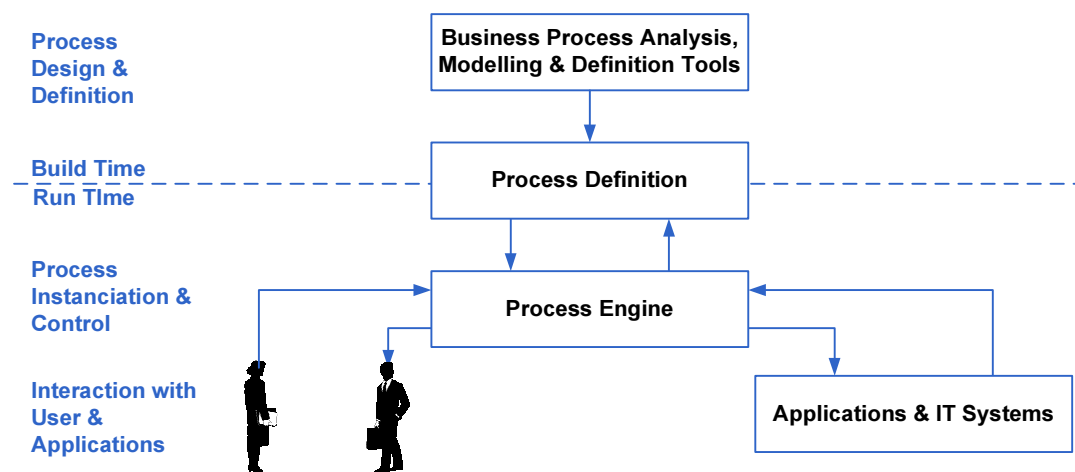


Abbildung 29: Das Prinzip der ausführbaren Prozesse

Die Modellierung von Prozessen in einer SOA mittels einer ausführbaren Modellierungssprache wie BPEL oder BPMN unterscheidet sich nicht von den üblichen BPM-Modellierungstechniken. Die Einschränkung von SOA gegenüber BPM erfolgt erst auf der Ebene der Ausführbarkeit. SOA versteht einen Prozess als automatisierbaren Prozess. Die Steuerung der Anwendung erfolgt über den abgebildeten Prozessfluss und dessen einzelne Schritte, Entscheidungspunkte und Verzweigungen. Damit wird Business Process Modelling zu einem zentralen Element für die Spezifikation einer Anwendung. Diese Spezifikation wird als Geschäftsprozess gestaltet und mit einem Modellierungstool als Ablaufdiagramm graphisch dargestellt. Aus dieser graphischen Darstellung des Prozesses wird die Prozessbeschreibung als Datei generiert, die generierte Prozessbeschreibung dann in eine Process Engine geladen und ausgeführt. Auf diese Weise entsteht aus einem Ablaufdiagramm ein konkreter Workflow, der sich auf jeder Workflow Engine ausführen lässt.



### 3.4.2.1 Bedeutung

Es gibt keine sinnvolle Einführung einer SOA in einem Unternehmen ohne den Einsatz einer Process Engine. Daher ist es notwendig, die Eigenschaften, die Struktur und auch den Aufbau dieser Werkzeuge gut zu kennen. Sie können davon ausgehen, dass auch Ihr Unternehmen ein solches Werkzeug einzusetzen gedenkt oder es bereits einsetzt. Ein wichtiger Aspekt ist die Tatsache, dass mit ausführbaren Prozessen eigentlich Workflows gemeint sind. Und ein Workflow ist die Automatisierung eines Geschäftsprozesses durch den Einsatz der Informatiktechnologie. Außerdem ist als ausführbarer Prozess nicht ganz dasselbe wie ein allgemeiner Geschäftsprozess. Der Workflow ist immer auf Automatisierbarkeit ausgerichtet. Also eignen sich vor allem diejenigen Geschäftsprozesse für eine Abbildung in Informationssysteme, die einen gewissen Automatisierungsgrad ermöglichen.

### 3.4.3 Geschäftsregeln

Geschäftsregeln (Business Rules) sind ein integraler Bestandteil des Geschäftsprozessmanagements. Sie steuern das Verhalten von Prozessen. Eine Business Rule ist eine Aussage, die einen Aspekt eines bestimmten Geschäftsfalles definiert oder begrenzt. Er soll die Struktur des Geschäftsfalles zur Geltung bringen und sein Verhalten beeinflussen oder kontrollieren. Business Rules können mittels Rule Engines umgesetzt werden. Diese unterstützen sowohl die Modellierung als auch die Verwaltung und Kontrolle der Geschäftsregeln in einer SOA. Sie sind geeignet, Regeln für die Validierung von Daten, für die Steuerung komplexer Aktionen und für die Steuerung des Verhaltens von Prozessen zu modellieren und auszuführen.

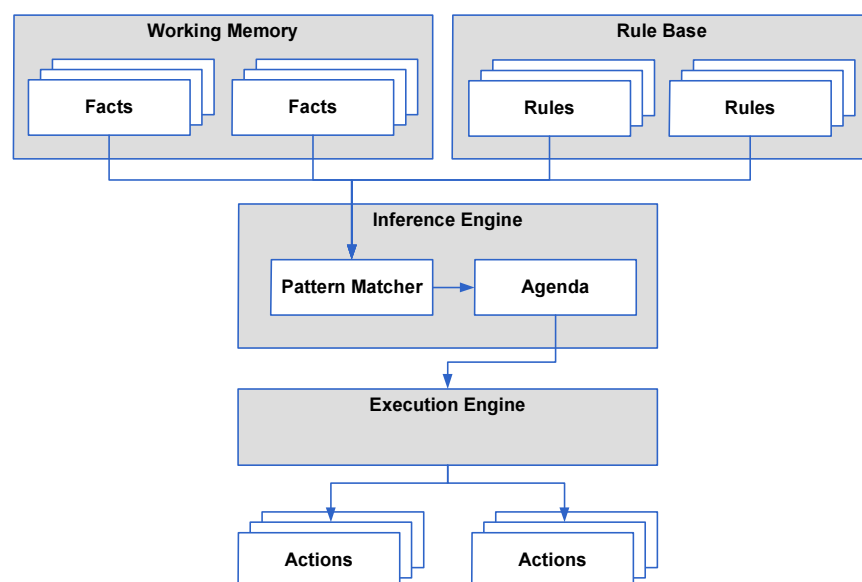


Abbildung 30: Aufbau einer Rule Engine

Der Einsatz von Rule Engines im Rahmen einer SOA erlaubt die Formulierung von Regeln zur Steuerung von einzelnen Prozessschritten, getrennt von der übrigen Business-Logik. Diese Regeln steuern einzelne Funktionen und Abläufe einer Anwendung. Im einfachen Fall gleichen Regeln einer „IF-THEN-ELSE“-Sequenz. Nur dass sowohl die Bedingungen als auch die Konsequenzen als Beschreibung mit graphischen Tools oder mit einfachen Sprach Konstrukten getrennt modelliert werden können. Die Rule Engine benutzt diese Regeln, um basierend auf den eingehenden Daten (Fakten), die vorgesehenen Aktionen auszulösen. So kann der Ablauf einer Anwendung zur Laufzeit gesteuert werden. Ändert sich nun eine bestimmte Regel, so ist lediglich ein Nachführen der Modellierung erforderlich. Die Rule Engine lädt anschließend die neuen Regeln, und die Anwendung verhält sich entsprechend. Der Einsatz von Rule Engines eignet sich daher für Geschäftsregeln, die sich häufig ändern, wie beispielsweise die Validierung von Konsistenzregeln, die Steuerung komplexer Aktionen oder als Steuerung von Reaktionen auf bestimmte Ereignisse.

Rule Engines kommen vor allem in drei Bereichen zum Einsatz. Als Mechanismus zur Validierung von Konsistenzregeln, als Steuerung komplexer Aktionen, basierend auf Fakten, oder als Steuerung von Reaktionen auf bestimmte Ereignisse [Beck 2005].

- **Konsistenzregeln** (Constraint Rules) sind Rahmenbedingungen, innerhalb derer Daten als konsistent oder gültig angesehen werden. Typischerweise werden damit Konsistenzprüfungen für die Eingabe von Werten durch Nutzer eines Systems durchgeführt. In einem auf SOA basierenden System sind jedoch vor allem Konsistenzprüfungen vor oder nach Datenkonvertierungen oder -Transformationen vorzusehen. Kritische Services können mit einem gut formulierten Regelwerk vor inkonsistenten Systemzuständen geschützt werden.
- Die Steuerung komplexer Aktionen, basierend auf **Fakten**, erfolgt aufgrund so genannter Produktionsregeln (Derivation Rules, Producer Rules oder Inference Rules). Im Gegensatz zur einfachen Berechnung von Werten (Computation) hängt die Berechnung eines Wertes durch Produktionsregeln vom Kontext der entsprechenden Fakten ab (Inference). Typische Anwendungen solcher Regeln sind die Prüfung für die Kreditwürdigkeit oder die Erzeugung von Rabattkonditionen.
- Die Steuerung von Aktionen als Reaktion auf bestimmte Ereignisse wird mit so genannten **Aktionsregeln** (Action/Event Rules, Behavior Rules) durchgeführt. So wird das Verhalten eines Prozesses gesteuert. Ein Prozess respektive Teilprozess kann über Aktionsregeln gestartet, angehalten oder auch abgebrochen werden. Diese Anwendung ist die bei weitem häufigste Anwendung einer Rule Engine im Rahmen einer SOA. Die Kombination einer BPEL Engine mit einer Rule Engine ist aus diesem Grund von den meisten Herstellern vorgesehen [Rosenberg, Dustdar 2005].

Eine Rule Engine erlaubt die zentrale Formulierung von Geschäftsregeln. Diese Regeln werden in einem Rule Repository gespeichert. Dieses Repository ist wiederum eine gute Quelle für Revisionen und Audits, um die Einhaltung von Compliance-Regeln zu prüfen. Geschäftsregeln werden so nicht mehr implizit formuliert, sondern können transparent gepflegt, verwaltet und geprüft werden.

Allen Einsatzgebieten gemeinsam ist die Tatsache, dass die Änderung der entsprechenden Regeln relativ häufig geschehen kann oder aber eine schrittweise Verfeinerung der Regelwerke aufgrund der tatsächlichen Faktenlage notwendig ist. In beiden Fällen ist der Einsatz von Rule Engines auf jeden Fall einer ausprogrammierten Lösung vorzuziehen.

### 3.4.3.1 Bedeutung

Jedes Unternehmen verwendet Geschäftsregeln. Sie beschreiben im Rahmen eines Geschäftsfalles, unter welchen Bedingungen welche Tätigkeit zu erfolgen hat, steuern also die einzelnen Schritte eines Geschäftsprozesses und sind damit für jedes betriebliche Informationssystem, das den entsprechenden Geschäftsprozess unterstützt, wichtig. Es ist kein Zufall, dass ein guter Teil der Analyse- und Designphase in einem Informatikprojekt dafür aufgewendet wird, die verschiedenen Regeln zu finden, wann was genau zu geschehen hat und wie es in Software zu gießen ist. Üblicherweise werden diese Regeln dann fix eingebaut. Sie werden zum impliziten Bestandteil eines Systems. Die Trennung der Regeln von der übrigen Logik einer Anwendung hat einerseits den Effekt, dass diese Regeln explizit sichtbar sind, und erlaubt andererseits, die Regeln zu ändern, ohne die ganze Anwendung ändern zu müssen. Aus diesem Grund sollten Sie den Einsatz einer Rule Engine in Betracht ziehen, auch wenn die Standardisierung für diese Werkzeuge weit weniger reif ist als diejenige für ausführbare Prozesse. Insbesondere, wenn sich die Geschäftsregeln in Ihrem Unternehmen oft ändern.

## 3.5 Service Ebene

Die Service-Ebene des SOA-Modells umfasst neben den standardisierten Serviceschnittstellen die Mechanismen zur Verwaltung von Diensten sowie spezialisierte Dienste. Die standardisierten Serviceschnittstellen sind das Herzstück einer SOA. Das Interface zu einem Service muss in jedem Fall eine Web Service-Schnittstelle sein, die mittels WSDL beschrieben ist. Werden andere Schnittstellentechniken oder Protokolle eingesetzt, geht ein wichtiger Vorteil von SOA verloren, und der Einsatz der Instrumente für die Orchestrierung, die Interoperabilität und die Verwendung von Diensten wird stark eingeschränkt. Die Mechanismen zur Verwaltung von Diensten unterstützen den Betrieb einer SOA auf Unternehmensebene. Aus betrieblicher Sicht sind Verantwortlichkeiten, Klassifizierung, Lebenszyklen, Dienstplanung, Versionsverwaltung, Testszenarien, Service Level Agreements und Verrechnungsmöglichkeiten für Dienste wichtige Elemente des Service Managements. Spezialisierte Dienste im Rahmen einer SOA sind Services, die auf Unternehmensebene Funktionalität zur Verfügung stellen, die weder einen bestimmten Fachbereich noch eine bestimmte Anwendung betreffen. Typische Vertreter sind Printing Services für ein unternehmensweites Output Management, Konvertierungs-Services für die Umwandlung von Formaten sowie Transformation Services für die Anreicherung respektive Filterung von Daten.

### 3.5.1 Service Management

Der Betrieb einer SOA wird durch Service Management unterstützt. Dieser Betrieb kann wie der Betrieb der IT mittels ITSM (IT Service Management) betrachtet werden. In diesem Sinne unterscheidet sich eine auf SOA basierende Lösung nicht von einem beliebigen Standardprodukt oder einer Individuallösung. SOA kann jedoch durch eine Reihe von unter dem Begriff „Service Management für SOA“ zusammengefassten Spezialmaßnahmen auf Unternehmensebene besonders unterstützt werden. Die Anforderungen an SOA Service Management sind die Erfassung der organisatorischen Verantwortlichkeiten von Service Providern und Service Consumern, die Klassifizierung von Services, die Bereitstellung von nachvollziehbaren Lebenszyklen, die Planung der Einführung verschiedener Dienste, die Verwaltung von verschiedenen Versionen desselben Service, die Instrumente zum Testen von Diensten, die Zuordnung von Service Level Agreements und unternehmensweite Verrechnungsmöglichkeiten für Dienste.

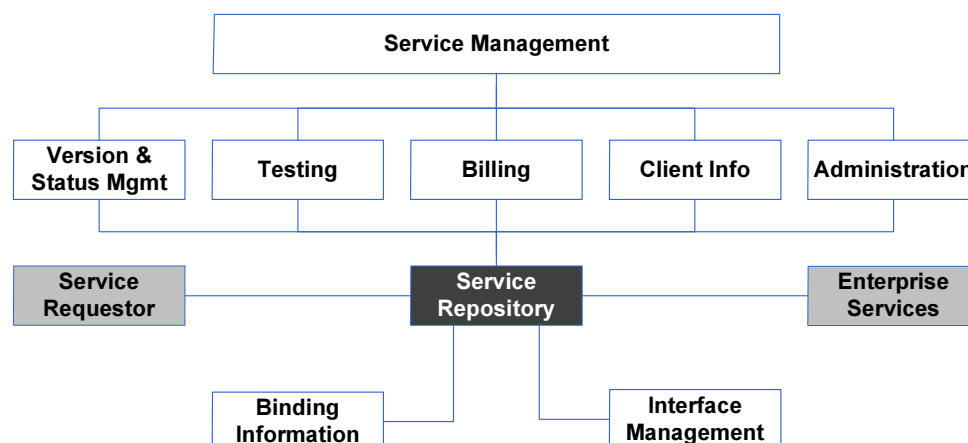


Abbildung 31: Service Management Tools

Die Besonderheiten des Service Managements im Rahmen einer SOA werden anhand der SOA Service Management Tools sichtbar. Ziel der Toolbox ist die Unterstützung der organisatorischen und technischen Maßnahmen, die eine Verwaltung von Diensten erleichtern.

Die Anforderungen an eine SOA Service Management Toolbox sind die Erfassung der organisatorischen Verantwortlichkeiten von Service Providern und Service Consumern, die Klassifizierung von Services, die Bereitstellung nachvollziehbarer Lebenszyklen, die Planung der Einführung verschiedener Dienste, die Verwaltung verschiedener Versionen desselben Service, die Instrumente für das Testen von Diensten, die Zuordnung von Service Level Agreements und unternehmensweite Verrechnungsmöglichkeiten für Dienste. Eine solche Toolbox ist auf die Nutzung innerhalb eines Unternehmens ausgerichtet und deckt nicht alle Anforderungen einer globalen Web Service-Infrastruktur ab.

Eine Service Management Toolbox stellt das logische Bindeglied zwischen Servicenehmer und den verschiedenen Enterprise Services dar. Es übernimmt damit Funktionen, die in einem Unternehmen den optimalen Einsatz von Diensten unterstützen können.

- **Service Management:** das User Interface des Service Repositories.
- **Version & Status Management:** die Verwaltung diverser Versionen eines Dienstes und der möglichen Status.
- **Testing:** die Verwaltung und Ablage von Testscripts, Testbeschreibungen und Test-Utilities.
- **Billing:** für die Verrechnung der Dienste zuständig.
- **Client-Info:** Die Servicenehmer und deren Organisationseinheit werden erfasst und regelmäßig informiert.
- **Administration:** Die Zuständigkeit für einen bestimmten Dienst wird über das Administrationsmodul erfasst und verwaltet.

#### 3.5.1.1 Bedeutung

Bei einer Einführung von SOA in Ihrem Unternehmen ist Service Management sicher nicht Ihr vorrangiges Problem, das Sie lösen müssen. Spätestens jedoch bei der Ausbreitung von SOA im Unternehmen spielen Fragestellungen, etwa wie die Verantwortung für einzelne Dienste zu organisieren ist, wie der Rollout verschiedener Services mit den laufenden Projekten koordiniert werden kann und wie der systematische Test, die Überwachung und die interne Ver-

rechnung von Diensten sichergestellt werden können, eine große Rolle. Selbstverständlich lassen sich diese Fragestellungen im Rahmen einer ITSM (IT Service Management)-Organisation lösen, sofern Ihr Unternehmen über eine solche Organisation und die entsprechenden Tools verfügt. In vielen Fällen wird es jedoch sinnvoll sein, eine SOA Service Management Toolbox einzusetzen. Sie bietet eine gute Unterstützung und kann Ihnen dabei helfen, Services in Ihrem Unternehmen bekannt zu machen und für viele Anwendungen als Alternative zum Selberbauen von Anwendungslogik einsetzbar zu machen. Die Verbreitung der Services steht und fällt mit ihrer Akzeptanz. Dienste können qualitativ noch so hochstehend realisiert sein, wenn sie niemand kennt und wenn sie sich nicht im Rahmen wichtiger Vorhaben planen lassen, werden sie sich nicht durchsetzen.

### 3.5.2 Service Interfaces

Die standardisierten Serviceschnittstellen sind das zentrale Basiselement einer SOA. Sie garantieren die nahtlose Interaktion von Diensten und somit das Paradigma von SOA, dass Services das fundamentale Element für die Erstellung von Anwendung sind. Obwohl verschiedene Standards für die Realisierung dieser Schnittstellen denkbar wären, garantieren allein der Einsatz von WSDL und SOAP die erforderliche Interoperabilität und insofern die Integrationsfähigkeit von Services in die verschiedenen Bestandteile einer SOA, wie beispielsweise diejenigen der Presentation- oder der Orchestration-Ebene.

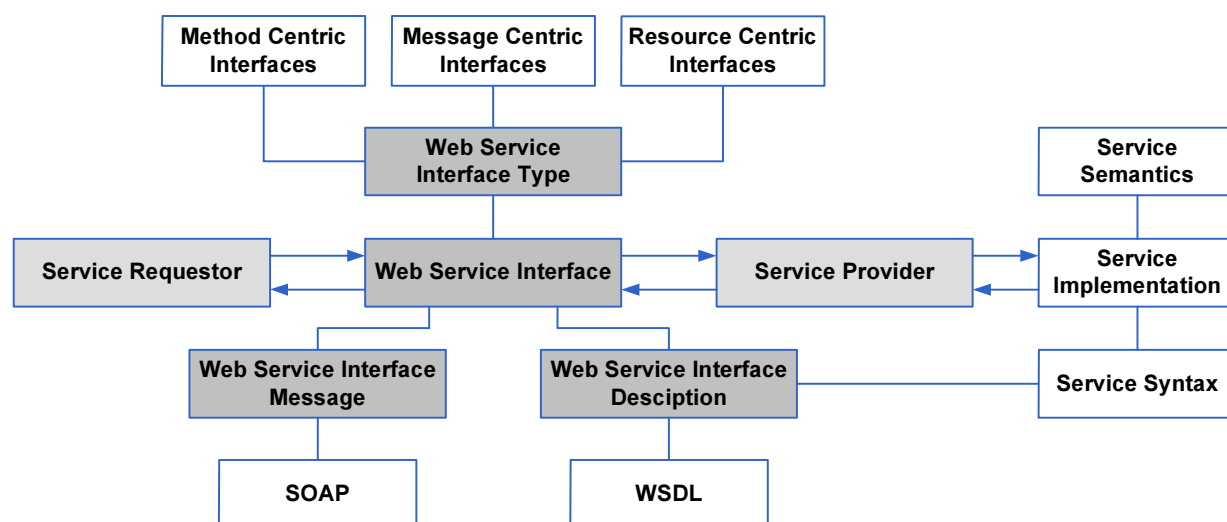


Abbildung 32: Service Interface und seine Bestandteile

Das Service Interface eines Web Service besteht aus der Beschreibung des Service, aus der Beschreibung der Meldungen, die der Service verarbeiten kann, sowie aus einer Klassifizierung des Service.

- **Web Service Interface Description:** Die semantische Beschreibung der Schnittstelle erfolgt mittels WSDL (Web Service Description Language). Sämtliche Operationen, die ein Service Provider ausführen kann, sowie sämtliche Meldungstypen, die als Parameter ausgetauscht werden können, sind in einem WSDL-Dokument erfasst.
- **Web Service Interface Message:** Der Aufruf eines Service durch einen Service Requestor und die Antwort des Service Providers – also der gesamte Datenaustausch – erfolgen als SOAP-Meldung.
- **Web Service Interface Type:** Service-Schnittstellen können verschiedene Charaktere aufweisen, die man als Interface-Typen bezeichnet. Methodenzentrierte Schnittstellen beschreiben einzelne Operationen und deren Eingangsparameter und Rückgabewerte. Meldungszentrierte Schnittstellen verarbeiten komplexe Meldungen mit einer einzigen Operation. Ressourcenzentrierte Schnittstellen stellen Inhalte dar und verbergen die Operationen.

In einer SOA kommen drei grundlegend verschiedene Interface-Typen zum Einsatz; Method-Centric Interfaces (Methodenorientierte Schnittstelle), Message-Centric Interfaces (meldungorientierte Schnittstelle) und Resource-Centric Interfaces (ressourcenorientierte Schnittstelle).

#### 3.5.2.1 Bedeutung

Das standardisierte Service Interface ist der Dreh- und Angelpunkt einer SOA. Ein Dienst ist nur dann im Rahmen einer SOA sinnvoll einsetzbar, wenn er als Web Service mittels SOAP und WSDL realisiert wird. Leider behaupten

allzuoft IT-Organisationen und Anbieter, sie würden seit Jahren SOA umsetzen, und verwenden in Wirklichkeit proprietäre oder andere Service-Schnittstellen wie beispielsweise CORBA. Das ist weder korrekt noch vernünftig. Achten Sie also darauf, was genau mit Services gemeint ist, wenn in Ihrem Unternehmen von Services gesprochen wird. Falls bereits Services bestehen, die nicht über standardisierte Web Service-Schnittstellen verfügen, ist das ein guter Anknüpfungspunkt für eine SOA-Einführung. Denn oft ist es einfach, aus einer proprietären Serviceschnittstelle eine standardisierte zu machen. Vorhandene Dienste sind also in jedem Fall Kandidaten für die Realisierung von Funktionen, die eine auf SOA basierende Anwendung einsetzen könnte.

### 3.5.3 Spezialisierte Services

Spezialisierte Dienste stellen im weitesten Sinn Querschnittsfunktionen für eine SOA dar. Im Speziellen handelt es sich dabei um Querschnittsfunktionen, die nicht schon durch andere Komponenten des Architekturmodells abgedeckt sind, in einer konkreten Umsetzung aber immer wieder realisiert werden. Beispiele sind Mechanismen zur Transformation und Konvertierung von Daten und unternehmensweite Output Management-Infrastrukturen. Es sind weitere Querschnittsfunktionen für eine SOA denkbar, die als spezialisierte Dienste unternehmensweit angeboten werden können. Die Auswahl dieser Dienste hängt stark von der Struktur der Informationssysteme und der Organisation einer Unternehmens-IT ab.

Spezialisierte Dienste sind mehr als technische Hilfsmittel, die die Bereitstellung und den Betrieb einer auf SOA basierenden Lösung vereinfachen. Sie sind ein Mittel zur organisatorischen Zentralisierung bestimmter Aufgaben in einer IT und erlauben damit die Kontrolle der entsprechenden Funktionalität. So kann man beispielsweise ein Audit Service als Garant der Einhaltung gesetzlicher Regulatoren einsetzen. Betreuen würden ihn sinnvollerweise die zuständigen Auditoren.

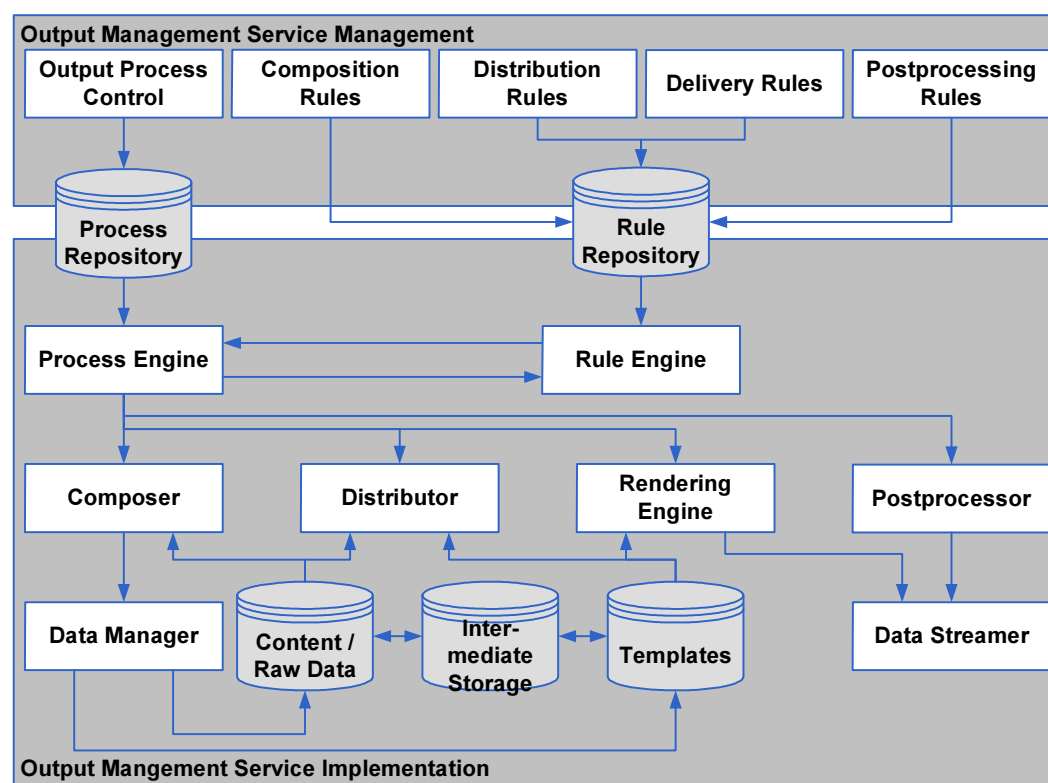


Abbildung 33: Beispiel eines spezialisierten Services: Output Management

Output Management umfasst die standardisierte Erstellung und Verteilung von elektronischen Dokumenten. Ein Output Management Service in einer SOA ist dann sinnvoll, wenn ein Unternehmen eine große Anzahl von Dokumenten regelmäßig an viele verschiedene Adressaten zu versenden hat.

Ein Output Management Service hilft diversen Bestandteilen von SOA, den gesamten Prozess zu steuern. Ausführbare Geschäftsprozesse, die als BPEL-Prozesse modelliert werden, steuern den Gesamtprozess. Die Assemblierung von Nutzdaten und Steuerdaten, die Zuordnung von Format-Informationen zu einzelnen Dokumententypen, die

Formatierung und die Sortierung des Outputs werden auf jeder Stufe – Composition, Distribution, Delivery und Post-processing – mittels Geschäftsregeln formuliert.

### 3.5.3.1 Bedeutung

Die Realisierung von unternehmensweit zentral zugänglichen Diensten ist ein wichtiger Pfeiler jeder SOA-Ausbreitungsstrategie. Gerade komplexere, jedoch für das ganze Unternehmen wichtige Querschnittsfunktionen bieten sich da für eine auf SOA basierende Lösung an. Ob das nun die so oft gebrauchte Konvertierung und Transformation von Daten betrifft oder das Output Management oder jedoch andere zentrale Dienste, hängt vom Bedarf Ihres Unternehmens ab. In jedem Unternehmen existieren bereits solche zentralen Dienste. Sie verwenden jedoch meist proprietäre Schnittstellen oder ganz spezielle Standards und verhindern oder behindern dadurch oft die Auswahl und Einführung einer nützlichen Standard Software oder einer guten Individuallösung, weil Letztere die entsprechende Schnittstelle nicht unterstützt. Vorhandene zentrale Dienste auf SOA umzustellen, ist in jedem Fall eine vernünftige Strategie, die dabei hilft, SOA einzuführen und zu verbreiten. Der zentrale Dienst kann einfacher genutzt werden, und die Investition zahlt sich aus.

## 3.6 Integration Architecture Ebene

Die Integration Architecture-Ebene ist für die Verknüpfung diverser Dienste und zur Verbindung von Diensten mit bestehenden Anwendungen oder Datenbanken sowie zur Koppelung von Services mit den Bestandteilen der Presentation-Ebene zuständig. Die entsprechenden Mechanismen können entweder als logische Integration (ohne ESB), als Enterprise Service Bus oder als Datenintegration realisiert werden. Die logische Integration als Minimallösung setzt lediglich ein Netzwerk als Infrastruktur voraus. Für die Realisierung vieler auf SOA basierender Lösungen reicht völlig aus. Der Einsatz eines ESB oder einer Data Integration-Plattform als Integration Architecture ist vor allem für größere SOA-Realisierungen sinnvoll. Ein ESB stellt die Kommunikation zwischen verschiedenen Services sicher, transformiert, validiert und aggregiert Meldungen, übernimmt das Routing von Meldungen und kann proprietäre Service-Schnittstellen als Web Service-Schnittstellen darstellen. Außerdem stellt jeder ESB weitere nützliche Instrumente zur Verfügung. Eine Data Integration Plattform ist ein spezialisierter ESB, der die unternehmensweite Harmonisierung und Konsolidierung von Daten unterstützt.

### 3.6.1 Logische Integration

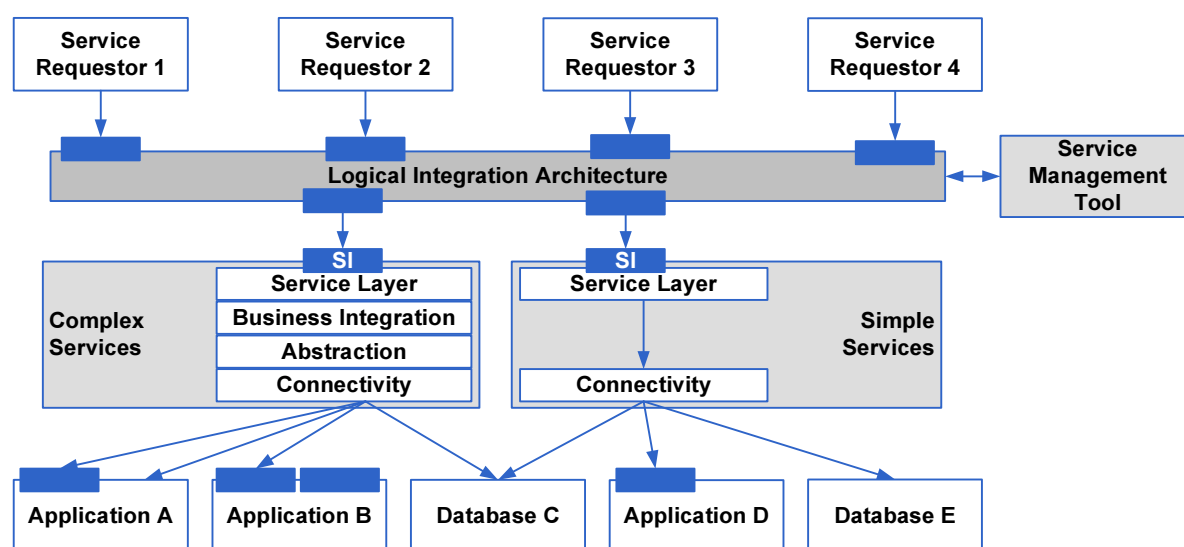


Abbildung 34: Logische Integration mit SOA

Die Umsetzung der Integration Architecture-Ebene von SOA durch die logische Integration, also ohne ESB, setzt als Basis diejenigen Infrastruktur-Bestandteile ein, ohne die eine SOA überhaupt nicht funktionieren würde. Dies sind ein vorhandenes Netzwerk mit dem Web Service Protocol-Stack. Konkret bedeutet dies, dass ein TCP/IP-Netzwerk obligatorisch ist und dass Application Server sowie die dazugehörigen Protokolle (HTTP und DNS) für die Bereitstellung

der Web Services vorliegen müssen. Ein Servicenehmer (Service Requestor) verwendet verschiedene Web Services, indem er die entsprechenden in WSDL definierten Service Endpoints aufruft. Die Services selbst sind entweder als einfache Dienste (Simple Services) oder als Dienste mit zusätzlicher Logik (Complex Services) realisiert. Beide kapseln bestehende Anwendungen und Datenbanken. Dabei wird direkt von der Service Implementation aus auf die entsprechenden Systeme zugegriffen.

Eine solche Realisierung ist in vielen einfacheren Lösungen, die auf SOA basieren sollen, völlig ausreichend. Nichtfunktionale Anforderungen, wie beispielsweise die Skalierbarkeit, werden auf der Services-Ebene (im Application Server) realisiert.

### 3.6.1.1 Bedeutung

Falls in einem Unternehmen die Einführung einer SOA kurz bevorsteht, ist es in den meisten Fällen nicht notwendig, gleich im Rahmen der Einführung einen ESB einzusetzen. Es geht auch ohne. Sie ersparen sich dabei unnötige Kosten und überladen Ihre Einführung nicht mit unnötiger Komplexität. Allerdings bedeutet auch die Realisierung der Integration Architecture-Ebene, dass Sie darüber nachdenken müssen, wie genau die notwendigen Funktionen umgesetzt werden sollen.

## 3.6.2 Enterprise Service Bus

Der Einsatz eines Enterprise Service Bus (ESB) oder auch Enterprise Integration Bus als zentraler Komponente zur Verbindung von Diensten ist für komplexere auf SOA basierende Lösungen sinnvoll. Ein ESB enthält typischerweise eine Sammlung von Instrumenten zur sicheren und garantierten Meldungsübertragung, Routingmechanismen für die Verteilung von Meldungen, vorgefertigten Adaptoren für die Integration verschiedenster Systeme, Management- und Überwachungstools und vieler anderer Komponenten. Er stellt eine logische zentrale Anlaufstelle in einem verteilten System dar. Die meisten Hersteller sehen einen ESB als wichtigen Bestandteil ihres SOA-Stacks vor.

### 3.6.2.1 Bedeutung

Den Einsatz eines ESBs in einer SOA sollte vor allem in zwei Fällen in Erwägung gezogen werden:

- für größere SOA-Realisierungen mit vielen beteiligten Systemen unterschiedlichster Technologie;
- für die Darstellung bestehender Dienste als Web Services, falls ein ESB bereits im Einsatz ist.

Für größere SOA-Realisierungen sind vor allem die Konnektivität und die garantierte Meldungsübermittlung eines ESBs nützlich. Bei der Auswahl eines ESBs sollten Sie jedoch unbedingt darauf achten, dass der ESB die SOA-Standards (SOAP, WSDL und BPEL/BPMN) unterstützt. So verfügen beispielsweise viele kommerzielle ESBs über proprietäre Workflow Tools, die Sie auf keinen Fall für die Modellierung von ausführbaren Geschäftsprozessen einsetzen sollten. Dies muss in BPEL oder BPMN erfolgen. Sie können im besten Fall für das Routing von Meldungen verwendet werden. Ein weiterer Aspekt ist die Gefahr der All-in-One-Lösung. Viele Unternehmen tendieren dazu, einem ESB viele Aufgaben zuzuordnen. In einer SOA ist der ESB eine Ausführung der Integration Architecture, also einer einzigen Ebene. Aufgaben, die in einer SOA auf den Ebenen Service, Orchestration und Presentation gelöst werden sollten, lassen sich zwar mit einem ESB realisieren, doch ist Letztere dafür nicht konzipiert worden. Dies hat zur Folge, dass die Unterstützung für eine bestimmte Aufgabenstellung, wie beispielsweise die zentralisierte Verwaltung von Diensten, immer eingeschränkt ist.

## 3.6.3 Data Integration

Der Einsatz von Data Integration oder auch Enterprise Information Integration (EII) als Integrationsarchitektur für eine SOA ist vor allem für größere Unternehmen sinnvoll. Die entsprechenden Produkte sind eine Kombination von spezialisierten ESBs mit spezialisierten Instrumenten aus dem Data Warehouse-Bereich. Sie erlauben eine generelle Sicht auf wichtige Unternehmensdaten und entwickeln sich zunehmend in Richtung Integration und Kombination von strukturierten mit unstrukturierten Daten. Sie stellen sehr ausgereifte Transformatoren und Konverter für die Integration verschiedener Datenquellen zu Verfügung.

### 3.6.3.1 Bedeutung

Enterprise Information Integration-Infrastrukturen sind meist nur in sehr großen Unternehmen im Einsatz. In diesem Fall liegt eine Einbindung einer EII in eine SOA auf der Hand. Die Infrastruktur muss lediglich um eine zusätzliche Abfrageschnittstelle, einen Web Service, erweitert werden. Auch könnten EII-Instrumente wie die Konvertierung, das Mapping und der inhaltliche Abgleich von Daten als spezialisierte Services im Rahmen einer SOA verwendet werden. Die meisten dieser Infrastrukturen sind jedoch auf strukturierte Daten beschränkt. Der Umgang mit unstrukturierten Daten im Rahmen einer SOA ist meistens ohne EII umzusetzen. Die Nutzung einer EII im Rahmen der Einführung

einer SOA in Ihrem Unternehmen ist nicht zu empfehlen, es sei denn, entsprechende Web Services sind bereits vorhanden.

In vielen Fällen werden ETL Tools als EII-Produkte angesehen. Auch diese Produkte bieten gute Tools für die Transformation und das Mapping von Daten an. Achten Sie jedoch darauf, dass Sie über eine standardisierte Web Service-Schnittstelle verfügen, da sie ansonsten nicht durch andere Bestandteile einer SOA wie beispielsweise einer Process Engine angesteuert werden können. Dann sind sie aber nicht mehr als Komponenten der SOA Ebene Integration Architecture zu betrachten.