

Zielformulierung

Die erste Lektion hat Ihnen einen Einstieg in die SOA gegeben und gezeigt, dass es in Abhängigkeit vom jeweiligen Hersteller, Analysten oder Berater ein sehr großes Spektrum unterschiedlicher SOA-Modelle gibt.

Die vorliegende zweite Lektion will Ihnen nun ein konsistentes Bezugssystem zur Arbeit mit der SOA an die Hand geben, mit dessen Hilfe Sie die im Zusammenhang mit der SOA diskutierten Begrifflichkeiten und Konzepte besser verstehen und einordnen können.

Mit dem Ziel der Identifikation eines Ordnungsrahmens wird die SOA in Kapitel 1 zunächst einmal „aus der Vogelperspektive“ betrachtet. Es wird dadurch klar werden, dass die SOA keine konkrete Architektur, sondern eine Referenz-Architektur darstellt, die prinzipiell mit unterschiedlichen Real-Architekturen verwirklichtbar ist. Zudem wird aus Vogelperspektive deutlich, dass bei der SOA sowohl Business Services als auch Implementation Services und Infrastructure Services von großer Bedeutung sind. Die Unterschiede zwischen diesen drei „Service-Arten“ sind für den Aufbau einer SOA von großer Bedeutung und nehmen in Kapitel 1 deshalb entsprechend Raum ein.

In Kapitel 2 sollen die „hinter“ der SOA liegenden „Key Concepts“ des SOA-Paradigmas vorgestellt werden. Solche Key Concepts sind: „Priorität für Business Services“, „Service-Vertragsversprechen“, „Interoperabilität verteilter Services“, „Lose Koppelbarkeit von Services“ und „Standards-Orientierung von Services“. Es handelt sich hierbei um Architektur-Richtlinien, von denen jede einzelne eine „conditio sine qua non“ beim Aufbau einer SOA darstellt.

Das 3. Kapitel widmet sich eingehend den in Kapitel 1 eingeführten Business Services, die die für ein Business Engineering notwendigen Funktionalitäten und Workflows zur Abbildung von Geschäftsprozessen bereitstellen. Das Kapitel wird verdeutlichen, dass Business Services in unterschiedlicher Granularität und hierarchisch geordnet vorliegen sollten. Die „Granularität“ sollte hierbei von fein-granularen „Component Business Function Services“ über Business Services mit mittlerer Granularität bis hin zu grob-granularen „Business Workflow Services“ als Repräsentanten von (Teil-)Geschäftsprozessen reichen. Während Component Business Function Services als *linking pins* zu den korrespondierenden „Implementation Services“ große Bedeutung beizumessen ist, erleichtern grob-granularere Business Services die Arbeit des Business Engineers.

Wie im Ordnungsrahmen von Kap. 1 eingehend erläutert, benötigen „Business Services“ korrespondierende „Implementation Services“. Kapitel 4 nimmt sich deshalb speziell der Implementation Services an, die de facto die von den Business Services benötigte Funktionalität liefern und deshalb an die Component Business Function Services angekoppelt werden müssen. Insbesondere soll hier auch auf die besonderen meist stiefmütterlich behandelten, aber in der Unternehmenspraxis oft zentralen Probleme von „Implementation Services bei Legacy-Systemen“ eingegangen werden. Natürlich wird auch zu „Implementation Services bei Komponenten-Software“ Stellung bezogen.

Kapitel 5 widmet sich den – ebenfalls in Kapitel 1 eingeführten – Infrastructure Services. Es wird gezeigt werden, dass der heute immer öfter diskutierte „Enterprise Service Bus“ (ESB) eine zentrale Rolle im Bereich der „Infrastructure Service Concepts“ einnimmt. Einem ESB können hierbei idealtypisch zehn unterschiedliche Core Capabilities – im Sinne von Infrastruktur-Aufgaben, die ein ESB zu bewerkstelligen hat – zugeschrieben werden. Kapitel 5 wird deshalb ausführlich auf diese Core Capabilities eingehen. Darüber hinaus wird auch auf die Relevanz von Quality-of-Service, Meta-Data- und SOA-Governance Concepts hingewiesen.

Die SOA wird immer wieder in Zusammenhang mit Begriffen wie „Web Services“, „EAI“, „Objekt-/Komponenten-Orientierung“, „Client/ Server“, „Web 2.0“ und „Open Source“ diskutiert. Kapitel 6 soll die entsprechenden Zusammenhänge systematisch herausarbeiten und somit dabei helfen, das Wesen der SOA noch besser zu verstehen und den Blick für SOA-relevante Bausteine zu schärfen.

Inhaltsverzeichnis

1	Ordnungsrahmen für Service-Orientierte Architekturen	1
1.1	Ziele des Kapitels	1
1.2	SOA im Spannungsfeld unterschiedlicher Engineering-Interessen	1
1.3	SOA als Referenz-Architektur	3
1.4	Zentrale Service-Arten einer SOA	4
1.4.1	Business Services	5
1.4.2	Implementation Services	7
1.4.3	Infrastructure Services	9
1.5	Ihr Lernerfolg aus diesem Kapitel	11
1.6	Übungsaufgaben zu diesem Kapitel	13
2	Key Concepts des SOA-Paradigmas	14
2.1	Ziele des Kapitels	14
2.2	Priorität für Business Services	14
2.3	Vertragsversprechen durch Business Services	16
2.4	Interoperabilität „verteilter“ Business Services	17
2.5	„Lose Koppelbarkeit“ von Business Services	19
2.6	Standards-Orientierung von Business Services	20
2.7	Ihr Lernerfolg aus diesem Kapitel	23
2.8	Übungsaufgaben zu diesem Kapitel	25
3	Business Service Concepts	26
3.1	Ziele des Kapitels	26
3.2	Probleme bei der Granularitäts-Festlegung	27
3.3	Hierarchisierung von Business Services	27
3.4	Business Function Services	29
3.4.1	Component Business Function Services	29
3.4.2	Composite Business Function Services	33
3.5	Business Workflow Services	35
3.5.1	Business Workflow Services als Repräsentanten von Geschäftsprozessen	35
3.5.2	Business Workflow Orchestration / Choreographie	37
3.5.3	Formale Sprachen zur Beschreibung von Workflows	39
3.5.4	Erschwernisse bei der Orchestrierung	39
3.6	Presentation Frontend für Business Services	41
3.7	Ihr Lernerfolg aus diesem Kapitel	42
3.8	Übungsaufgaben zu diesem Kapitel	43
4	Implementation Service Concepts	44
4.1	Ziele des Kapitels	44

4.2	Zusammenhänge zwischen Business Services und Implementation Services.....	44
4.3	Implementation Services bei legacy-Software.....	46
4.3.1	Besonderheiten von Altsystem-Code.....	46
4.3.2	Mapping von Business Services mit Altsystem-Code.....	46
4.3.3	Das Konzept der „anwendungs-orientierten ECA“.....	47
4.4	Implementation Services bei Komponenten-Software.....	49
4.4.1	Bestandteile offener Software-Komponenten.....	49
4.4.2	Mapping von fachlichen Business Services und komponenten-orientierten Implementation Services.....	50
4.4.3	Das Konzept der „objektorientierte ECA“.....	51
4.5	Ihr Lernerfolg aus diesem Kapitel.....	54
4.6	Übungsaufgaben zu diesem Kapitel.....	55
5	Infrastructure Service Concepts.....	56
5.1	Ziele des Kapitels.....	56
5.2	Enterprise Service Bus Concepts.....	56
5.3	Weitere Infrastructure Services.....	60
5.4	Ihr Lernerfolg aus diesem Kapitel.....	61
5.5	Übungsaufgaben zu diesem Kapitel.....	62
6	Einordnung und Abgrenzung der SOA.....	63
6.1	Ziele des Kapitels.....	63
6.2	Abgrenzung „SOA / Andere prominente Konzepte“.....	64
6.2.1	SOA und Web Service Infrastrukturen.....	64
6.2.2	SOA und EAI-Software.....	66
6.2.3	SOA und Objekt-/Komponenten-Orientierung.....	69
6.2.4	SOA und Client/Server.....	71
6.3	SOA meets Web 2.0 und Open Source.....	73
6.3.1	Web 2.0 und SOA.....	73
6.3.2	Open Source meets SOA.....	79
6.4	SOA – Die Normalität des Overhead.....	80
6.5	Ihr Lernerfolg aus diesem Kapitel.....	81
6.6	Übungsaufgaben zu diesem Kapitel.....	83
7	Stichwortverzeichnis.....	84
8	Literaturverzeichnis.....	85
9	Endnotenverzeichnis.....	88

1 ORDNUNGSRAHMEN FÜR SERVICE-ORIENTIERTE ARCHITEKTUREN

1.1 Ziele des Kapitels

Dieses Kapitel versucht „Ordnung“ in die Begriffs- und Konzepte-Welt der SOA zu bringen. Hierzu wird die SOA zunächst einmal aus der „Vogelperspektive“ betrachtet. Es wird dadurch klar werden, dass die SOA

- im Spannungsfeld von Business Engineering, Software Engineering und Infrastructure Engineering liegt und dass diese drei Engineering-Bereiche in der Unternehmenspraxis ineinander greifen müssen.
- keine konkrete Architektur, sondern eine Referenz-Architektur darstellt, die prinzipiell mit unterschiedlichen Real-Architekturen verwirklichtbar ist.
- sowohl Business Services als auch Implementation Services und Infrastructure Services kennt und dass die Unterschiede zwischen diesen drei „Service-Arten“ für den Aufbau einer SOA von großer Bedeutung sind.

1.2 SOA im Spannungsfeld unterschiedlicher Engineering-Interessen

„SOA im Nebel der Interessen“ – so könnte man etwas provokant die Vielzahl unterschiedlicher Angebote und Beiträge zu „Service-orientierten Architekturen“ überschreiben.

Eher software-technologisch interessierte Beiträge und Dienstleistungsangebote sehen in der SOA meist eine „verbesserte Komponententechnologie“ und setzen deshalb einen Service eher mit einer Software-Komponente gleich. „In these terms, a service is larger than a component, but smaller than an application“¹. Dies mag zwar eine gute Definition für jene Services sein, die wir „Implementation Services“ nennen wollen (siehe Abschnitt 1.4.2). Sie erschwert aber zunächst den Blick auf Services von einer Granularität, wie sie für die Abbildung von Geschäftsprozessen erforderlich ist.

Jeder der drei Bereiche stellt andere Anforderungen an eine „Lösung“. Nichtsdestoweniger bedarf es einer sinnvollen Verzahnung, damit eine optimale Gesamtlösung entsteht.

1.3 SOA als Referenz-Architektur

Die SOA ist inzwischen zu einem Paradigma (Denkmuster) für die geschäftsprozess-orientierte² Entwicklung, Bereitstellung und Nutzung verteilter bzw. verteilter Software geworden.³

Hierbei ist zu beachten, dass die SOA als paradigmatisch-methodischer Ansatz grundsätzlich technologie-neutral⁴ ist und eher als „Referenz-Architektur“ denn als „Konkrete Architektur“ zu bezeichnen ist.

Eine Referenz-Architektur stellt die Basis für eine „Konkrete Architektur“ dar, wie es nachfolgendes Beispiel verdeutlicht.

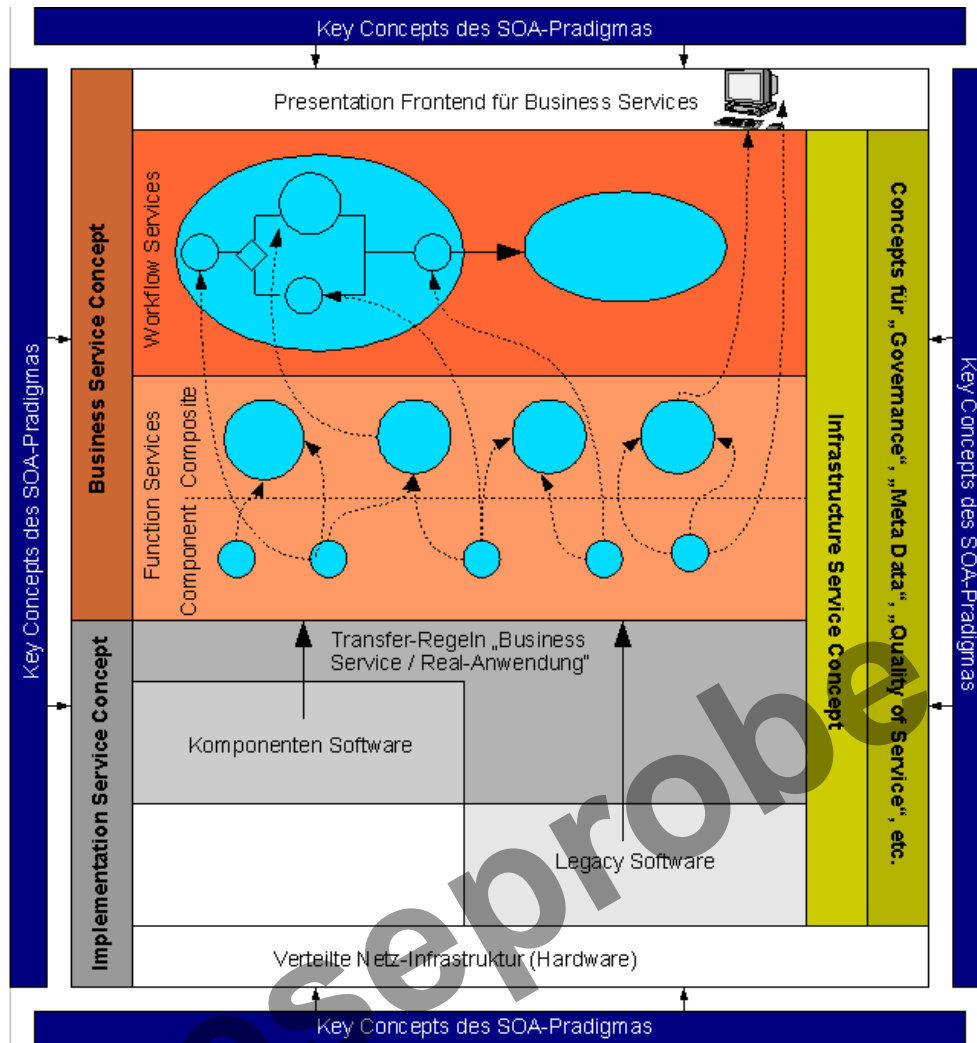


Abb. 2: SOA als Referenz-Architektur⁵

Demzufolge kann eine SOA im Grunde genommen durch unterschiedliche Konkrete-Architekturen und deren produktspezifische Ausprägungen realisiert werden.⁶

Wie jede Architektur⁷ beschreibt auch die SOA lediglich die grundlegenden System-Elemente⁸ und deren prinzipielles Zusammenspiel.⁹

Abbildung 3 zeigt auf, dass die grundlegenden System-Elemente einer SOA „Services“¹⁰ sind, die es als „Business Services“, „Implementation Services“ und „Infrastructure Services“ gibt. Abbildung 3 zeigt auch, dass diese Services in einer SOA auf eine bestimmte Art „zusammenarbeiten“ und sich hierbei an den „Key Concepts“ einer SOA ausrichten.¹¹

Abb. 3: SOA aus der Vogelperspektive¹²

1.4 Zentrale Service-Arten einer SOA

Generell bezeichnet der Begriff „Service“ eine Leistung, die ein Service-Anbieter¹³ für einen Service-Nehmer erbringt.¹⁴ Das kann vieles sein.

„Gerade aber weil der Begriff ‚Service‘ in unserem Sprachgebrauch so vielfältig verwendet wird und hiermit insbesondere keine einheitliche Granularität assoziiert wird (unter den Begriff ‚Service‘ fällt sowohl eine aufwändige Inspektion als auch das Füllen von Einkaufstüten), ergibt sich daraus eine beeindruckende Vielfalt von Interpretationsmöglichkeiten.“¹⁵

Bereits in Abschnitt 1.2 wurde aufgezeigt, wie unterschiedlich der Service-Begriff in eher software-technologisch- bzw. eher consulting-orientierten Beiträgen verwendet wird.

Wie problematisch dies ist, wird deutlich, wenn man in einem größeren Kreis die Frage aufwirft, „mit wie vielen Services bei der SOA-Umsetzung z.B. eines Finanzdienstleistungsunternehmens denn wohl zu rechnen sei“.¹⁶ – Man muss sich dann bei den Antworten „auf die erstaunliche Bandbreite von wenigen Dutzend bis hin zu vielen tausend Services gefasst machen.“¹⁷

Unklarheiten bei der Verwendung des Service-Begriffs sind aber wegen der Zentralität von Services für die Umsetzung einer „Service“-orientierten Architektur fatal und müssen vermieden werden.

Bei der Realisierung einer SOA sollte deshalb von Beginn an sehr sorgfältig zwischen

- Business Services,
- Implementation Services und
- Infrastructure Services

unterschieden werden.¹⁸

1.4.1 Business Services

Im Mittelpunkt einer SOA stehen Business Services, weshalb mitunter auch recht treffend von Service Oriented Business Architecture (SOBA) gesprochen wird.¹⁹

Es handelt sich bei Business Services um „fachliche“ Services, wie sie speziell für die Geschäftsprozess-Abwicklung in Unternehmen gebraucht werden. Solche fachlichen Services müssen in unterschiedlicher Granularität vorliegen; ihr Spektrum reicht von elementaren Geschäftsfunktionen bis hin zu wiederverwendbaren Workflows.

Die Funktionalität eines Business Service wird – unabhängig von seiner Granularität – nach außen über standardisierte Beschreibungen und Aufrufmechanismen zur Verfügung stellt. Die Inanspruchnahme solcher Business Services ist im Bedarfsfall sogar zeit- oder nutzungsbasiert verrechenbar.

Ein Business Service ist insofern „gekapselt“, als für den Service-Nutzer nur sichtbar ist, was der Service kann und wie seine Input- und Output-Strukturen aussehen. Dies wird gemeinhin mit Hilfe einer IDL²⁰ beschrieben.

Konkret könnte ein in der Sprache der Fachanwender formulierter (relativ feinkörniger) Business Service aus dem Bereich des Börsenhandels z.B. „PlaceTrade“ lauten.²¹ Dass die entsprechende Funktionalität im Hintergrund durch einen Implementation Service „saveTradeOrder()“ codiert ist, interessiert weder Business Engineers noch Anwender (Black-Box-Prinzip).

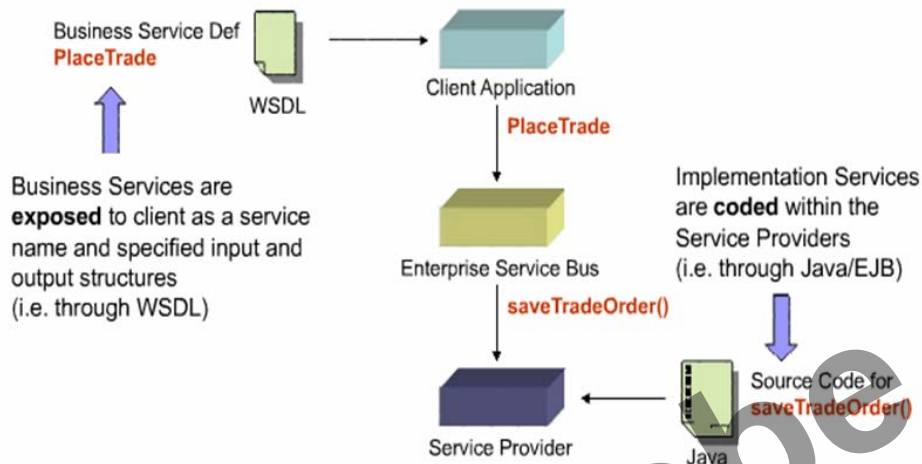


Abb. 4: Business Services vs. Implementation Services²²

Für ein Business Engineering werden Business Services unterschiedlicher Granularität gebraucht.

Wenn Business Engineers die Wahl haben, werden sie – zwecks Komplexitätsreduktion – beim Abbilden von Geschäftsprozessen natürlich gerne auf umfassendere relativ grob-granulare Business Services, die vielleicht sogar schon Teilprozesse abdecken, zurückgreifen.

Oftmals „passen“ die grob-granularen Business Services aber nicht exakt für die Abbildung eines spezifischen Geschäftsprozesses, so dass man zu „fein-granulareren“ Business Services zurückgreifen muss.

Außerdem setzen sich „grob-granulare“ Business Services aus „fein-granulareren“ Business Services zusammen, die quasi den „Rohstoff“ des Business Engineering bilden.

„Nach unten hin“ stellen Business Services ein „linking pin“ zu den „Implementation Services“ dar. Dies deshalb, weil sie auf der einen Seite auf die Belange des Business Engineers und auf der anderen Seite auf die Belange des Software Engineers – der ja die Funktionalitäten aus den real vorhandenen Plattformen bereitzustellen hat²³ – abgestimmt sein müssen.

Die Ausführungen zu den Business Services werden in Kapitel 3 vertieft.

1.4.2 Implementation Services

Business Services „beziehen“ ihre Funktionalitäten aus unterschiedlichsten Real-Systemen.²⁴ Diese können durch „Altanwendungen oder Web Services oder Komponenten implementiert sein oder durch eine Zusammenstellung dieser Bausteine.“²⁵

Es sind also die „real existierenden“ Applikationslandschaften, die (als Service Provider)²⁶ die „Implementation Services“ für die „Business Services“ (als Service-Nehmer) zur Verfügung stellen müssen.²⁷

Dies ist insbesondere im Falle großer Applikations-Landschaften nicht einfach.

Diese „bestehen zunächst einmal aus Altlasten (englisch: legacy). Das ist nicht despektierlich gemeint, denn im Grunde ist jedes in Betrieb befindliche System eine Altlast. Es bedeutet aber, dass man SOA nicht als Konzept für die Entwicklung eines neuen Systems oder Systemverbunds betrachten kann. Man muss sich der Tatsache stellen, dass die Systeme im Normalfall bereits verwendet werden und dies auch weiterhin der Fall sein wird. [...] Systemlandschaften sind eine wüste Zusammenstellung verschiedener Plattformen, Programmiersprachen und Programmierparadigmen. Selbst die Middleware kann heterogen sein.“²⁸

„Ein Grund für Heterogenität ist die lange Lebensdauer der Systeme und der darin verwalteten Daten. Es kommen zwar immer neue Funktionen und Systeme hinzu, aber es fällt schwer, existierende Systeme abzuschalten. Ein Problem dabei ist, dass das Abschalten keinen direkten Geschäftsnutzen hat. Es handelt sich vielmehr um eine Investition in die Wartbarkeit der Systeme. Derartige Investitionen haben in der Praxis gegenüber neuen konkreten Features mit fachlichem Nutzen einen schweren Stand. Sie werden daher oft erst dann getätigt, wenn die Systeme außer Kontrolle geraten sind und die Probleme offensichtlich werden.“²⁹

„In der Vergangenheit gab es oft den Ansatz, das Problem der Heterogenität durch eine Harmonisierung zu lösen. Es ist keine Frage, dass Harmonisierung – wenn sie denn funktioniert –, hilft. Wenn man veraltete Plattformen loswird, macht dies die Dinge im Normalfall einfacher. Das Problem ist, dass man den Zustand der Harmonisierung trotzdem nie erreicht. Kurz bevor das letzte Fünkchen Heterogenität eliminiert wird, gibt es eine Firmenfusion oder eine neue Technologie, und alles fängt von vorne an.“³⁰

Sicherlich ist also nicht davon auszugehen, dass die SOA einer „heilen komponenten-orientierten“ Software-Welt gegenübertritt, in der es „nur“ gilt, aus den existierenden Komponenten „bessere“ SOA-Komponenten zu machen.

Bei genauerer Betrachtung relativiert diese Tatsache auch die Bedeutung von Webservice-Infrastrukturen und stärkt die Notwendigkeit von Integrations-Plattformen als Infrastruktur auch für die Einbindung von Implementation Services aus Legacy-Systemen.

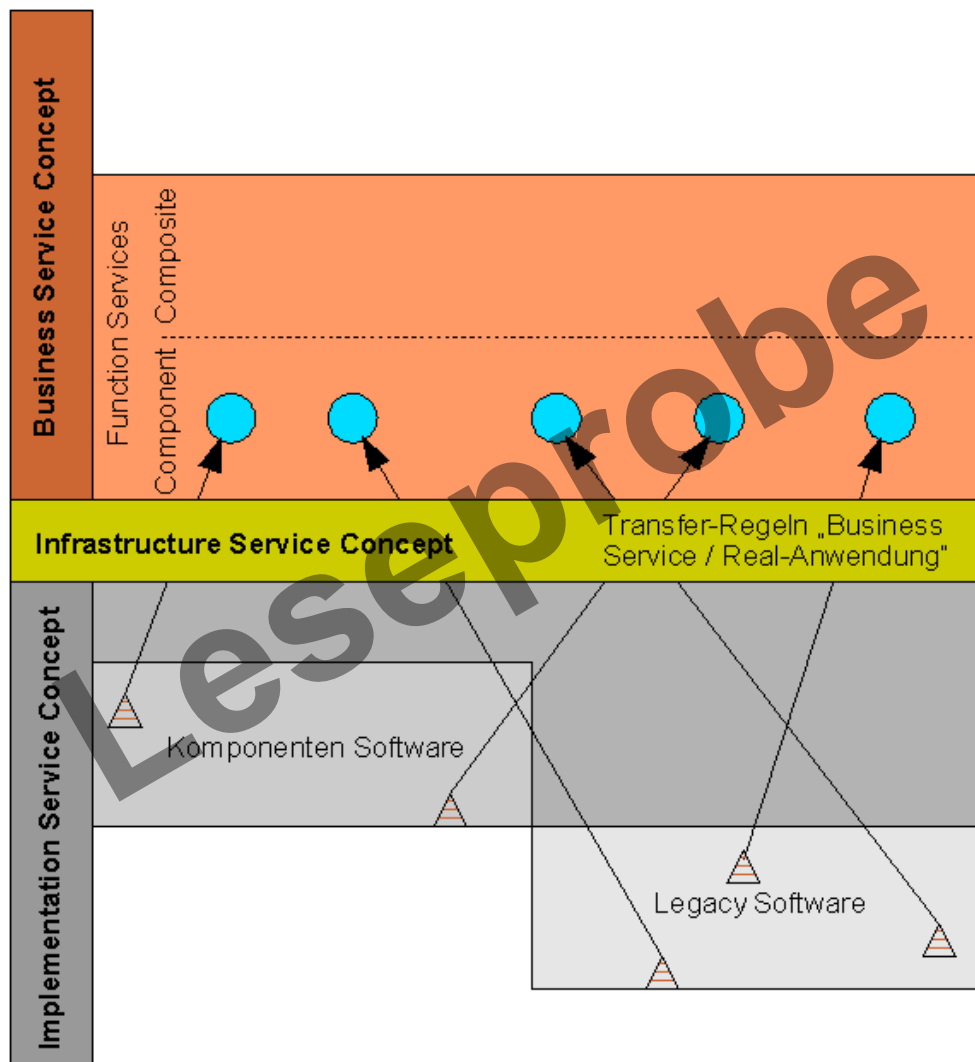


Abb. 5: Implementation Service in „real existierenden“ Software-Landschaften

Abbildung 5 macht dies deutlich, indem sie zwischen „Business Services“ und „Implementation Services“ explizit die Ebene der „Infrastructure Services“ einfügt und somit auch hierfür SOA Concepts einfordert.

1.4.3 Infrastructure Services

Um eine SOA in einem Unternehmen umzusetzen, bedarf es in aller Regel einer Vielzahl von Infrastructure Services.

Versteht man einen ESB (Enterprise Service Bus) als Obermenge relevanter Infrastructure Services – und nicht als konkretes Produkt –, so müsste ein idealtypischer ESB die in Abbildung 6 aufgeführten „10 Core Capabilities“ haben.

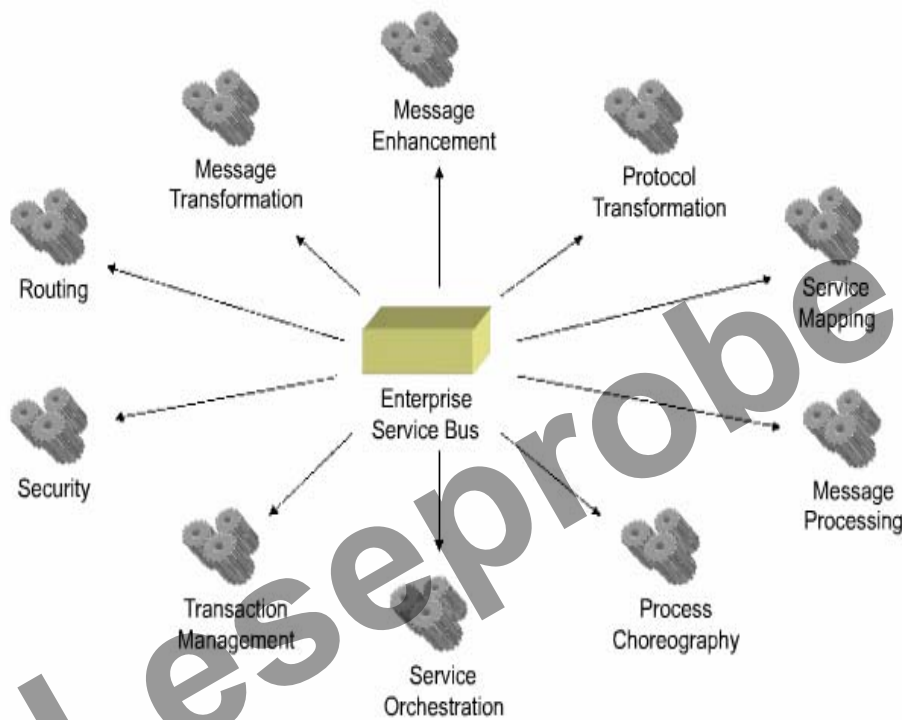


Abb. 6: „10 Infrastructure Core Capabilities“ eines ESB³¹

Die 10 Infrastructure Core Capabilities eines ESB werden in Kapitel 5 näher erläutert.

Wie wichtig „vor die Klammer“ gezogene Infrastructure Services sind, soll hier nur kurz und exemplarisch anhand der Capabilities „Service Orchestration“ und „Service Mapping“ aufgezeigt werden:

- **Service Orchestration** (Beispiel 1): Beim Business Process Engineering werden Infrastruktur-Services benötigt, die dabei helfen, dass „Component Business Function Services“ auf Geschäftsprozess-Ebene zu „Workflow Business Process Services“ zusammengefügt und ablauffähig gemacht werden können.³²
- **Service Mapping** (Beispiel 2): Beim Software Engineering werden unter anderem Hilfestellungen für ein flexibles Verzahnen von Business Services und Implementation Services bzw. Code gebraucht. Entsprechende Infrastructure Services können hierfür den „glue“³³ liefern.

Welche der „10 Core Capabilities“ ein ganz bestimmtes Unternehmen tatsächlich braucht, hängt davon ab, welche reale Applikationslandschaft in einem Unternehmen vorzufinden ist. Unterschiedliche Unternehmen können also unterschiedliche ESB benötigen.

Es sei bereits jetzt angemerkt, dass demzufolge auch unterschiedliche Anbieter von ESB-Produkten unterschiedliche Core Capabilities abdecken, so dass man sich nicht wundern muss, wenn zur Zeit mit so vielen core capability-spezifischen ESB-Definitionen geworben wird.

Leseprobe

1.5 Ihr Lernerfolg aus diesem Kapitel

- Die SOA stellt ein auf breiter Front anerkanntes „Denkmuster“ (ein Paradigma) für die Geschäftsprozess-orientierte Entwicklung, Bereitstellung und Nutzung verteilter bzw. verteilter Software dar.
- Die SOA ist eine technologie-neutrale Referenz-Architektur und keine Konkret-Architektur (wie z.B. CORBA oder die Web Service Architektur). Sie beschreibt auf relativ abstrakter Ebene die notwendigen Architektur-Elemente und -Zusammenhänge sowie „Key Concepts“, die beim Aufbau SOA-basierter Konkret-Architekturen zu beachten sind.
- Eine Gleichsetzung der „Referenz-Architektur SOA“ mit der „Konkret-Architektur Web Services“ ist nicht richtig. Aufgrund von in Unternehmen vorhandener Legacy-Systeme wird eine SOA oft sogar unter Beibehaltung unterschiedlicher, nebeneinander weiterexistierender Konkret-Architekturen verwirklicht, wofür dann umfangreiche „Infrastructure Services“ benötigt werden, die man in Summe als Enterprise Service Bus (ESB) bezeichnen kann.
- Zwar soll die SOA insbesondere zu einer Flexibilisierung der Geschäftsprozesse beitragen und damit den Business Engineer bei seiner Arbeit wirksam unterstützen. Nichtsdestoweniger muss sich die SOA aber gleichzeitig auch der Frage widmen, wie solche Business Services in der Praxis unserer Unternehmen software-technisch und infrastrukturbezogen umzusetzen sind. Sie steht also im Spannungsfeld von Business-, Software- und Infrastructure Engineering.
- Business Engineers arbeiten mit in Fachsprache formulierten Business Services und bilden damit letztlich Geschäftsprozesse ab. Business Services können von unterschiedlicher Granularität sein, was in Kapitel 3 näher erläutert wird.
- Software Engineers stellen gemäß vorher festgelegter Transferregeln die Implementation Services der realen Anwendungslandschaften (als Service Provider) bereit und liefern den Business Services (als Service-Nehmer) die von diesen angeforderten Funktionalitäten. Kapitel 4 wird hierzu ausführlich Stellung beziehen.

- Infrastructure Engineers kümmern sich um die Bereitstellung und Aufrechterhaltung der für eine SOA notwendigen Infrastructure Services. Solche Infrastructure Services können vielfältigster Art sein und werden durch Anbieter und die Literatur immer häufiger unter den Begriff Enterprise Service Bus subsumiert. Als Beispiele für Infrastructure-Aufgaben eines ESB wurden bereits „Business Service Orchestration“ und „Mapping von Business Services und Implementation Services“ genannt. Kapitel 5 wird zu den „10 Core Capabilities“ der ESB Stellung nehmen.

Leseprobe