

Aufbau eines serviceorientierten Unternehmens

Matthias Suthe
ii4838

Übersicht

1. Grundlagen

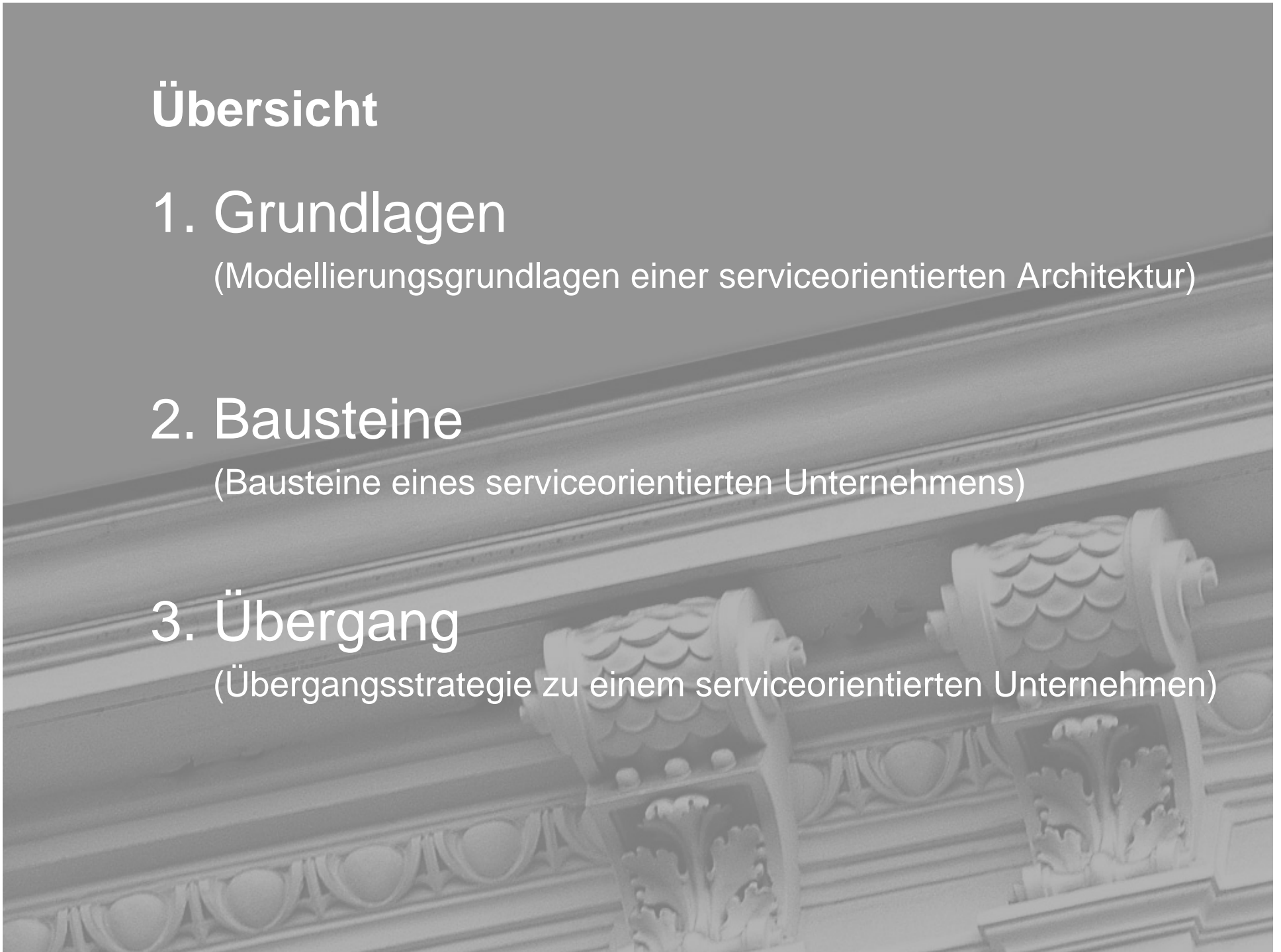
(Modellierungsgrundlagen einer serviceorientierten Architektur)

2. Bausteine

(Bausteine eines serviceorientierten Unternehmens)

3. Übergang

(Übergangsstrategie zu einem serviceorientierten Unternehmen)



Modellierungsgrundlagen einer serviceorientierten Architektur



**Elemente der Geschäftslogik können
verschieden bezeichnet werden, abhängig vom
Zusammenhang in dem sie betrachtet werden.**

| Drei Grundarten von Teilen einer | Geschäftslogik

Jede kann eine andere Sichtweise auf dasselbe Element der Geschäftslogik sein

1. Aktivitäten

2. Dienste

3. Prozesse

Grundlagen | Bausteine | Übergang

Aktivitäten | Dienste | Prozesse

Aktivitäten

- ◎ Teil einer Geschäftslogik welcher, mit anderen Teilen zusammenarbeitet um eine Geschäftsfunktion zu erfüllen
- ◎ Spannweite und ausgeübte Funktion variieren stark

1. primitive Geschäftsaktivität

2. Prozessaktivität

3. Geschäftsaktivität



Grundlagen | Bausteine | Übergang

Aktivitäten | Dienste | Prozesse

| primitive Geschäftsaktivität

- ⊙ Grundlegender, kleinster Baustein einer serviceorientierten Architektur
 - ⊙ Typischerweise gruppiert zu einem einfachen Geschäftsprozess
-



Grundlagen | Bausteine | Übergang

Aktivitäten | Dienste | Prozesse

| Prozessaktivität

- ⊙ Aktivität innerhalb eines Prozesses
 - ⊙ Stellt einen ausführbaren Schritt innerhalb der Ablauflogik dar
-



Grundlagen | Bausteine | Übergang

Aktivitäten | Dienste | Prozesse

| Geschäftsaktivität

- ⊙ Beschreibt üblicherweise einen groben Teil einer Geschäftslogik
 - ⊙ Kann Dienste und Prozesse umfassen
-

Grundlagen | Bausteine | Übergang

Aktivitäten | **Dienste** | Prozesse

Dienste

- ⊙ geschlossene, autonome Teile einer Geschäftslogik
- ⊙ kann eigenständig zum Erfüllen einer Aufgabe genutzt werden oder mit anderen Diensten zusammen
- ⊙ Benennung kommt auf den Zusammenhang an

1. primitiver Geschäftsdienst

2. Prozessdienst

3. Geschäftsdienst



Grundlagen | Bausteine | Übergang

Aktivitäten | Dienste | Prozesse

| primitiver Geschäftsdienst

- ⊙ Geschäftslogik welche eine bestimmte Geschäftsfunktion beschreibt
 - ⊙ enthält keine anderen Dienste
 - ⊙ oft Teil eines zusammengesetzten Dienstes
-



Grundlagen | Bausteine | Übergang

Aktivitäten | **Dienste** | Prozesse

| Prozessdienst

- ⊙ nicht Teil eines Prozesses
 - ⊙ enthält die Ablauflogik eines Prozesses
 - ⊙ auf die Logik eines einzigen Geschäftsprozesses beschränkt
-



Grundlagen | Bausteine | Übergang

Aktivitäten | **Dienste** | Prozesse

| **Geschäftsdienst**

- ⊙ kann einen beliebigen Grad an Funktionalität besitzen
 - ⊙ kann eigenständig einen Teil der Geschäftslogik oder einen/mehrere Prozesse darstellen
-

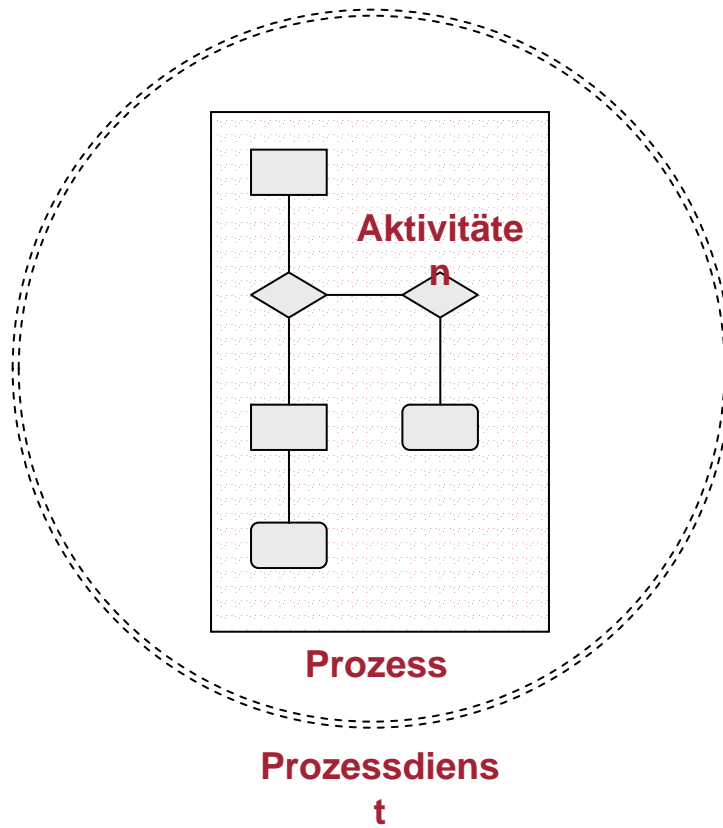
Grundlagen | Bausteine | Übergang

Aktivitäten | Dienste | Prozesse

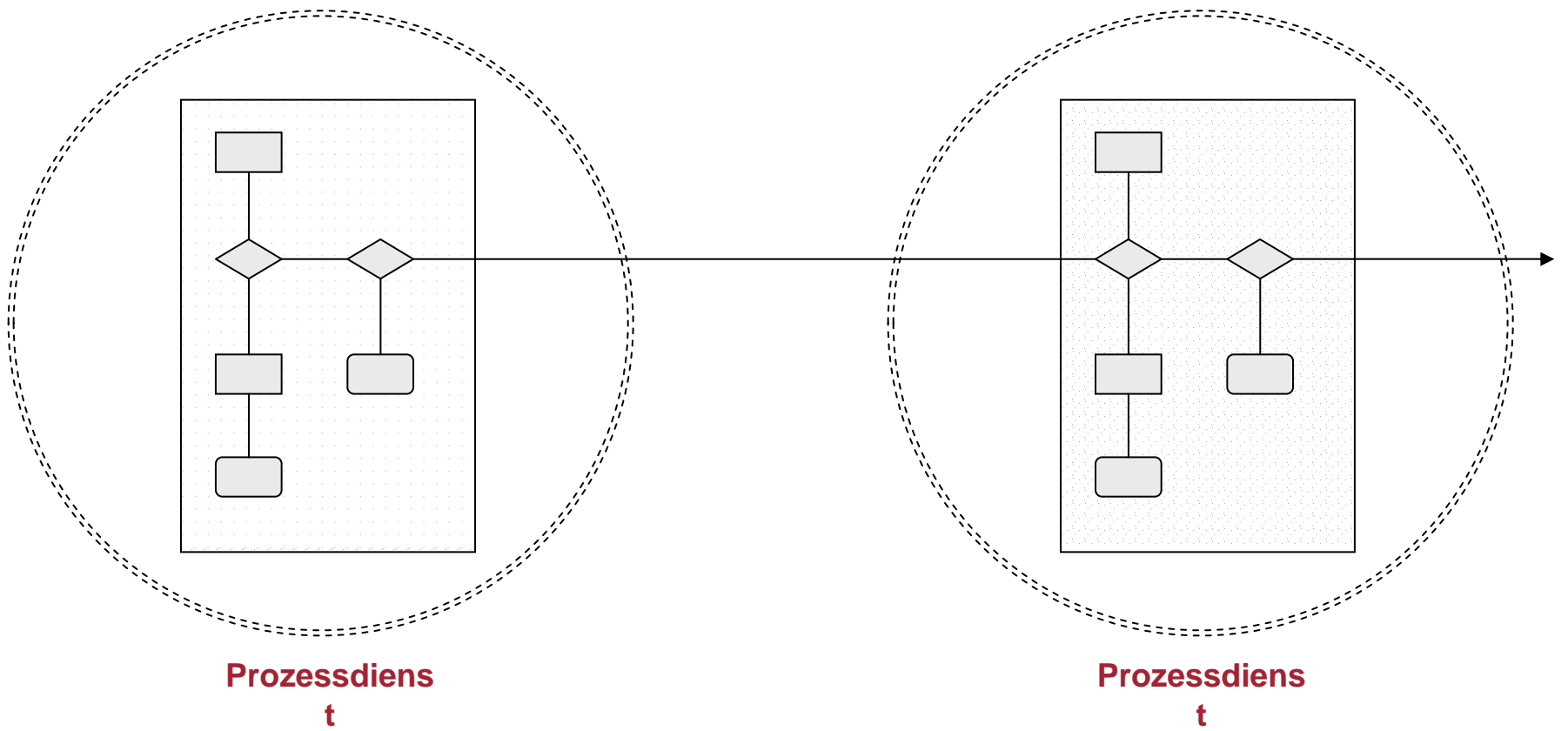
Prozesse

- ⊙ Ansammlung von Aktivitäten und Diensten
- ⊙ besteht aus Prozessaktivitäten, gesteuert von einem Prozessdienst
- ⊙ jede Prozessaktivität kann ein beliebiger Dienst sein, auch ein Prozessdienst
- ⊙ können mit Hilfe ihrer Prozessdienste verknüpft werden
- ⊙ mehrere Prozesse können in einem Geschäftsdienst zusammengefasst werden

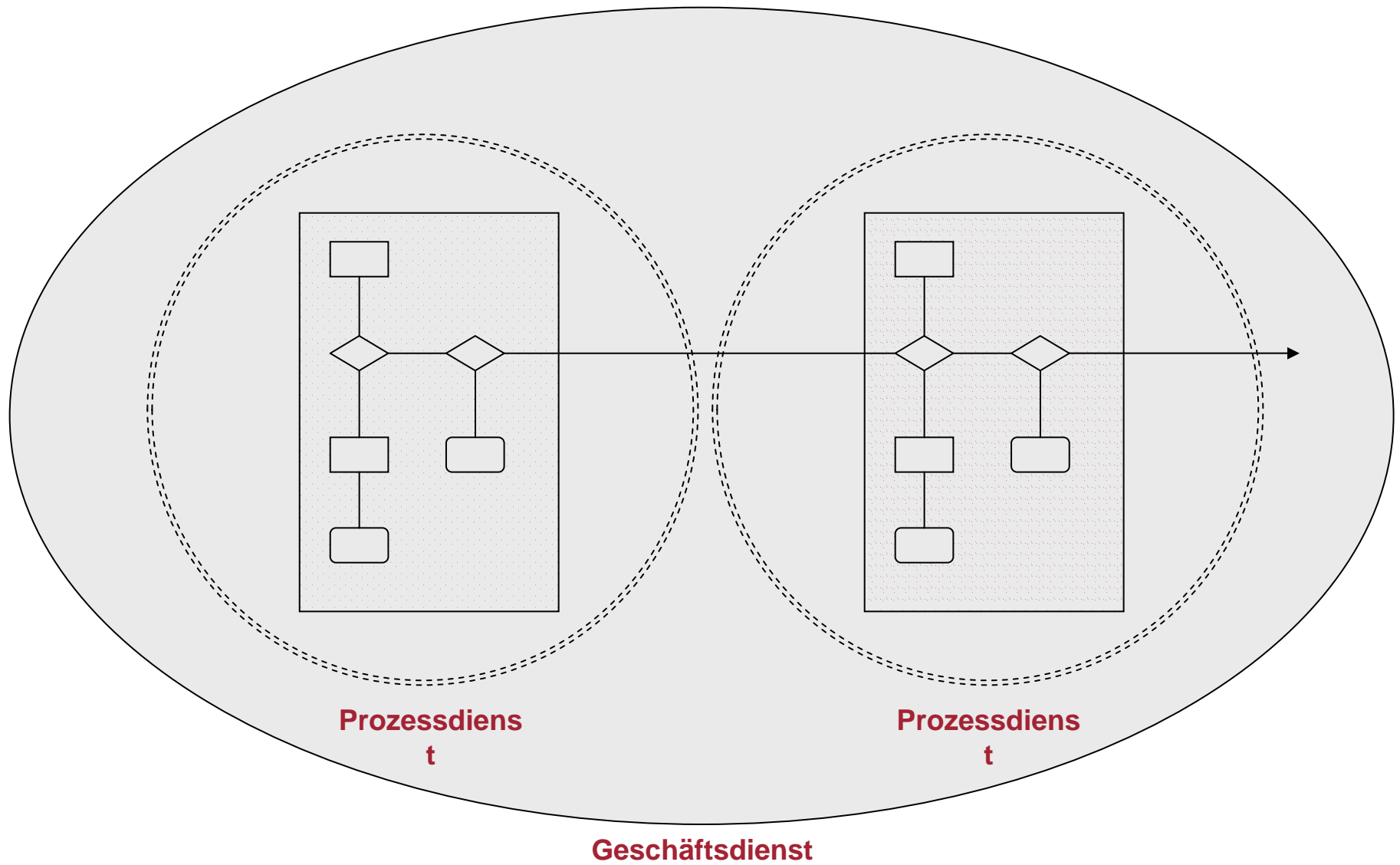
Grundlagen | Bausteine | Übergang



Grundlagen | Bausteine | Übergang



Grundlagen | Bausteine | Übergang



Bausteine eines serviceorientierten Unternehmens



Sechs Unternehmensbausteine

Jeder steht für ein bestimmtes Maß an Funktionalität und übertrifft den vorherigen an Umfang und Komplexität.

einfache
Geschäftsaktivität

einfacher
Geschäftsdienst

einfacher
Geschäftsprozess

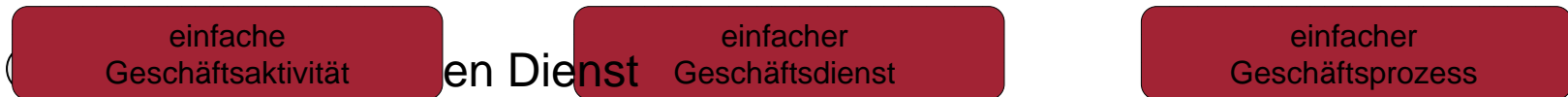
erweiterter
Geschäftsprozess

Unternehmensumgebungs
Geschäftsprozess

Unternehmens
Geschäftsprozess

⊙ einfacher Arbeitsschritt

⊙ kleinste Einheit von Geschäftslogik

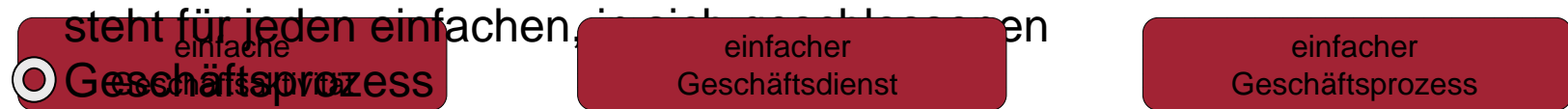


Beispiel: Batterie in einer Uhr wechseln

1. Uhr öffnen
2. Batterie tauschen
3. Uhr schließen

⊙ wichtigster Baustein eines serviceorientierten Geschäftsmodells

⊙



⊙ Zusammenschluss von mehreren einfachen Geschäftsaktivitäten



beinhaltet Geschäftslogik in offener Natur aus autonom

2. Batterie tauschen

3. Uhr schließen

- Verbund von logisch zusammenhängenden einfachen Geschäftsdiensten



durch Ablauflogik zusammengefasst und koordiniert



einfache
Geschäftsaktivität

einfacher
Geschäftsdienst

einfacher
Geschäftsprozess

jeder enthaltene Dienst bleibt autonom



kann insgesamt als Prozessdienst angesehen werden

erweiterter
Geschäftsprozess

Beispiel:

Batterie für Kunden wechseln

interne Geschäftsprozess

Unternehmens
Geschäftsprozess

1. Uhr entgegennehmen
2. Batterie wechseln
3. Uhr zurückgeben

○ bietet Funktionalität, die mehr als einen Prozess umfasst

○ kann aus mehreren einfachen Geschäftsprozessen bestehen

○ kann aus einem Prozess bestehen, welcher einen Dienst von einem dritten „borgt“

○ Kann in einem Prozessdienst enthalten sein, welcher seine Logik veröffentlichen

Beispiel: Batterie wechseln und Uhr reinigen

1. Uhr entgegennehmen
2. Batterie wechseln
3. Uhr reinigen
4. Uhr zurückgeben

○ stellt umfangreiche Ablauflogik zur Verfügung

○ besteht aus und koordiniert einfache und erweiterte Geschäftsprozesse

○ verbindet mehrere einfache Geschäftsprozesse Abteilungen oder sonstige logische einfacher Geschäftsprozesse Geschäftsbereiche einfacher Geschäftsprozesse

erweiterter Geschäftsprozess **Beispiel:** Unternehmensumgebungs Geschäftsprozess **wechseln,** Unternehmens Geschäftsprozess
reinigen, Ersatzteil einbauen und Rechnung erstellen

1. Batterie wechseln und reinigen
2. Ersatzteil einbauen

⊙ Größter Baustein

⊙ Zusammensetzung aller anderen Bausteine

⊙ verbindet üblicherweise unterschiedliche Unternehmensbereiche



Beispiel: erweiterter Geschäftsprozess, Unternehmensumgebungs Geschäftsprozess, n, reinigen, Ersatzteil, Geschäftsprozess

bestellen und einbauen

1. Uhr kontrollieren und reinigen
2. Ersatzteil bei externem/internen Partner bestellen

Sechs Technologiebausteine

Gleiches Baukastenprinzip wie beim Unternehmensmodell

Webservice
Operation

Webservice

serviceorientierte
Anwendungsarchitektur

serviceorientierte
Integrationsarchitektur

serviceorientierte EAI
Architektur

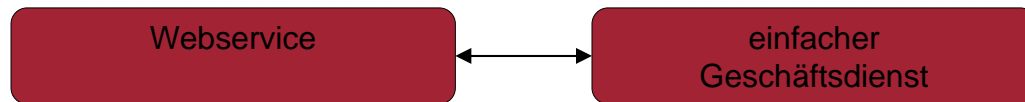
serviceorientierte Unternehmens-
Integrationsarchitektur



- Ⓒ einfache Aktion die ein Webservice ausführt
- Ⓒ kleinste Einheit von Anwendungslogik
- Ⓒ mehrere bilden einen Webservice

Beispiel: Bei Anfrage an eine Datenbank

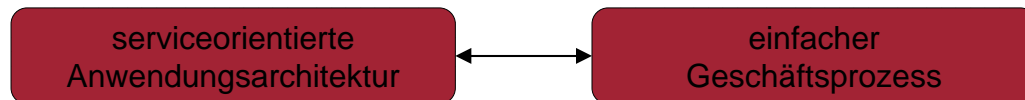
Daten ausgeben



- ⊙ wichtigster Baustein einer serviceorientierten Anwendung
- ⊙ steht für jede in sich geschlossene Programmlogik
- ⊙ Zusammenschluss von mehreren einfachen Webservice Operationen

Beispiel: Anfrage an eine Datenbank

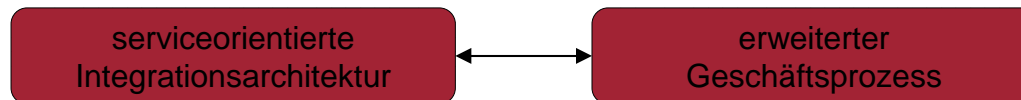
1. Anfrage erhalten
2. in der Datenbank suchen
3. Daten ausgeben



- ⊙ Verbund von logisch zusammenhängenden einfachen Webservices
- ⊙ durch Anwendungslogik zusammengestellt und koordiniert
- ⊙ jeder enthaltene Webservice bleibt autonom

Beispiel: Daten anfordern und bearbeiten

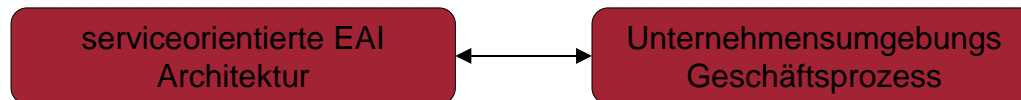
1. Daten von WS 1 anfragen (Ortsdaten)
2. Daten von WS 2 bearbeiten lassen (Entfernungen)
3. Daten ausgeben



- ⊙ bietet Funktionalität die mehr als eine Anwendung umfasst
- ⊙ kann aus mehreren serviceorientierten Anwendungen bestehen

Beispiel: Daten anfordern, bearbeiten und ausgeben

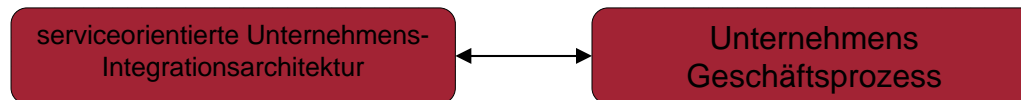
1. Daten von SOAA 1 anfordern (Entfernungen)
2. Daten von WS 2 bearbeiten lassen (grafische Darstellung)
3. Daten ausgeben



- ⊙ integriert mehrere serviceorientierte Anwendungen durch einen gesonderten Geschäftsprozess (meist ein Webservice)
- ⊙ hierauf basierende Lösungen werden oft als Service Bus bezeichnet

Beispiel: Daten anfordern, verknüpfen und ausgeben

1. Daten von SOAA 1 anfordern (Entfernungen)
2. Daten von SOAA 2 anfordern (Höhe über NN)
3. Daten verarbeiten und ausgeben (3-D Karte)



- ⊙ Größter Baustein
- ⊙ Zusammensetzung aller anderen Bausteine
- ⊙ verbindet üblicherweise unterschiedliche Integrationsumgebungen

Beispiel: 3D Karte erstellen

1. Geokoordinaten, Höhenangaben, Flüsse etc. Anfragen
2. Grafische Karte erstellen
3. Ortsnamen in die Karte eintragen

Übergangsstrategie zu einem serviceorientierten Unternehmen



Webservices und andere XML Technologien sind sehr vielseitig, daher kann ihr Einsatz einen großen Einfluss, sowohl organisatorisch als auch technisch, haben.

Da Webservices und andere XML Technologien recht willkürlich und unabhängig eingesetzt werden können, muss man aufpassen, dass man nicht einfach eine heterogene Umgebung durch eine andere ersetzt.

Um das zu vermeiden wurde das Layered Scope Model (LSM) entwickelt.

Das Layered Scope Model (LSM)

Wohl am besten mit „geschichtetes Anwendungsbereichsmodell“ zu übersetzen.

Das LSM erläutert wie in mehreren Phasen die Standardisierung innerhalb einer Organisation kontrolliert werden kann.

Mit Hilfe des LSM kann man:

1. eine Umgebung erschaffen, in der der Gebrauch von XML und Webservices standardisiert ist
2. die negativen Auswirkungen des Umstiegs mildern
3. die technischen Ressourcen so schnell entwickeln wie ihre Umgebung
4. eine Reihe von Standardarchitekturen aufbauen
5. ein durchdachtes serviceorientiertes Sicherheitsmodell entwickeln
6. Schritt für Schritt ein serviceorientiertes Unternehmen aufbauen

Das Layered Scope Model (LSM)

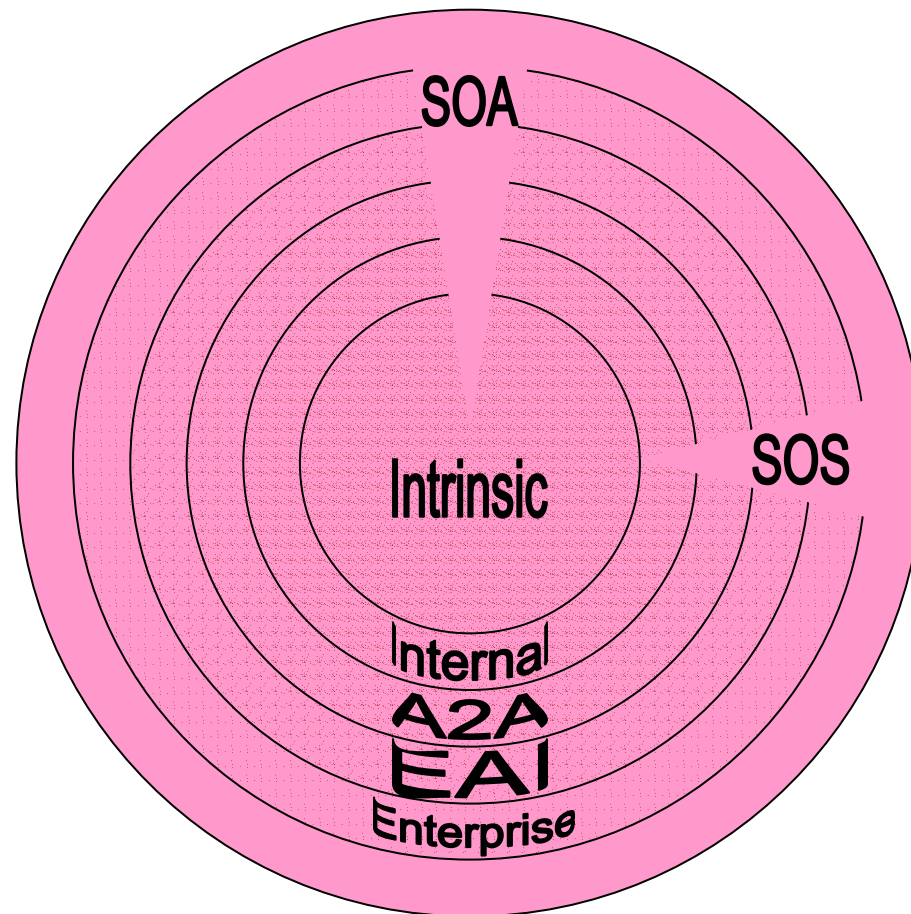
Jede Schicht steht für eine Phase im Übergang zu einem serviceorientierten Unternehmen.

Jede Phase hat zwei Ziele:

1. ein Set von Standards abhängig vom Anwendungsbereich in der Architektur bereitstellen
2. Webservices und XML so einzusetzen und zu nutzen, dass die Anwendungsbereiche der inneren Schichten begrenzt werden

Das Layered Scope Model (LSM)

Die verschiedenen Schichten



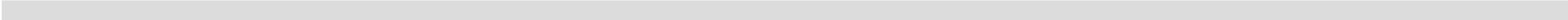


Grundlagen | Bausteine | **Übergang**
Intrinsic | Internal | A2A | EAI | Enterprise

Intrinsic Layer (eigentliche/wesenhafte Schicht)

Hier befinden sich Daten die nur eine Anwendung nutzt und die nie nach außen gelangen müssen, zum Beispiel Statusinformationen. Hier anzusetzen ist optional.

Technologien:

- XML zur Datenrepräsentation
 - XSD um XML Schemata zu definieren
 - WSDL und SOAP für utility services
- 

Internal Layer (firmeninterne Schicht)

Hier befinden sich Informationsarchitekturen die Firmenwissen beinhalten.

Ab dieser Schicht sollte begonnen werden Sicherheitsstrukturen einzubinden, da Anwendungen in dieser Schicht meist weniger Sicherheitsvorkehrungen haben als in der darunter liegenden

Technologie

n:

- WS-Security
- XML
- WS-Attachments
- XSD
- WSDL
- SOAP
- XHTML
- XQuery/XPath
- XSL T

A2A Layer (Application-to-Application)

Diese Schicht ist die aufwendigste und wichtigste, da hier ein standardisierter Rahmen für den Datenaustausch zwischen allen Anwendungen im Unternehmen erstellt werden muss.

Technologie

n:

- XML
- XSD
- WSDL
- SOAP
- XQuery/XPath
- XSLT
- WS-Security
- WS-Coordination
- WS-Transaction
- WS-Attachments

EAI Layer (Enterprise Application Integration)

Diese Schicht umfasst sämtliche Anwendungen, welche durch einen gesonderten Geschäftsprozess verbunden sind. Hier wird vollkommen auf das in der A2A Schicht erstellte Framework aufgebaut.

Technologie

n:

- XML
 - XSD
 - WSDL
 - SOAP
 - XQuery/XPath
 - XSLT
 - RDFI
- WS-Security
 - WS-Policy
 - WS-ReliableMessaging
 - WS-Coordination
 - WS-Attachments

Enterprise Layer

In dieser Schicht werden Architekturen zwischen den verschiedenen Geschäftsprozessen aufgebaut.

Diese Schicht komplettiert die Umsetzung des gesamten Unternehmens nach außen hin, daher können hier auch Schnittstellen und die dazugehörigen Sicherheitsvorkehrungen erstellt werden, für die Zusammenarbeit mit zukünftigen externen Partnern.

Technologien:

- alle bisher kennen gelernten

Das Layered Scope Model (LSM)

**Für detailliertere
Informationen zum LSM
besuchen Sie:**

www.layeredscopemodel.com





| Fragen? Unklarheiten?



**Vielen Dank für Ihre
Aufmerksamkeit !**
