

Engineering IT-basierter Services

Prof. Dr. Klaus-Peter Fährnich

Teil 13. Service-Technologien

21.06.2006

Engineering IT-basierter Services

1. Einführung
2. Typologisierung von Dienstleistungen
3. Grundlagen des Service Engineering
4. Vorgehensmodelle
5. Plattformstrategie: Produktmodelle und Modularisierung
6. Methoden und Werkzeuge I
7. Methoden und Werkzeuge II
8. Methoden und Werkzeuge III
9. Methoden und Werkzeuge IV
10. Werkzeuganwendung I
11. Werkzeuganwendung II
12. Zusammenfassung Werkzeuge
- 13. Service-Technologien**
14. Kundenintegration und Kundenmanagement
15. Fallstudie IT-Services
16. Standardisierung im Dienstleistungsbereich
17. Dienstleistungen im internationalen Wettbewerb
18. Praxisteil I
19. Praxisteil II

Motivation

Technik als Service-Enabler durch:

- Standardisierte Datenhaltung und -kommunikation
 - XML
 - Component Ware
 - Webservices
- Medienintegration
 - Unified Messaging
 - Multimedia
- Management „weicher“ Assets
 - Wissens- und Contentmanagement
 - Ontologien, RDF, OWL, SPDL
- Ubiquitäre Kommunikation
 - Teleservice
 - Mobile Anwendungen

Technologien: XML

- durch die Trennung von
 - Inhalt (Dokument, Datenbank)
 - Struktur (DTD, XML schema)
 - Layout (XML-XSL)
- kann ein XML-Dokument z.B. für
 - Druckversion
 - Webseite (HTML)
 - Seite für PDA/Handy (WAP)
 - Sprachausgabe (Voice-XML)
 - Direktkommunikation mit anderen Systemen
 - und mehr
- verwendet werden

Technologien: XML

- bereits vorgestellte Anwendungen, die auch XML unterstützen:
 - MindJet MindManager X5
 - EngCon (Produktkonfigurator)
 - MS Office 2003
 - SAP R/3
 - IDS Scheer ARIS
 - IBM Rational Rose
 - IBM Websphere Business Integration
 - MS Navision
 - ...
- **XML ist ein „Muss“!**
- aber: unterschiedliche Programme = unterschiedliche Formate
 - oft gibt es keine einheitlichen Standards, jede Firma hat eigene Schemata

Technologien: Component-Ware

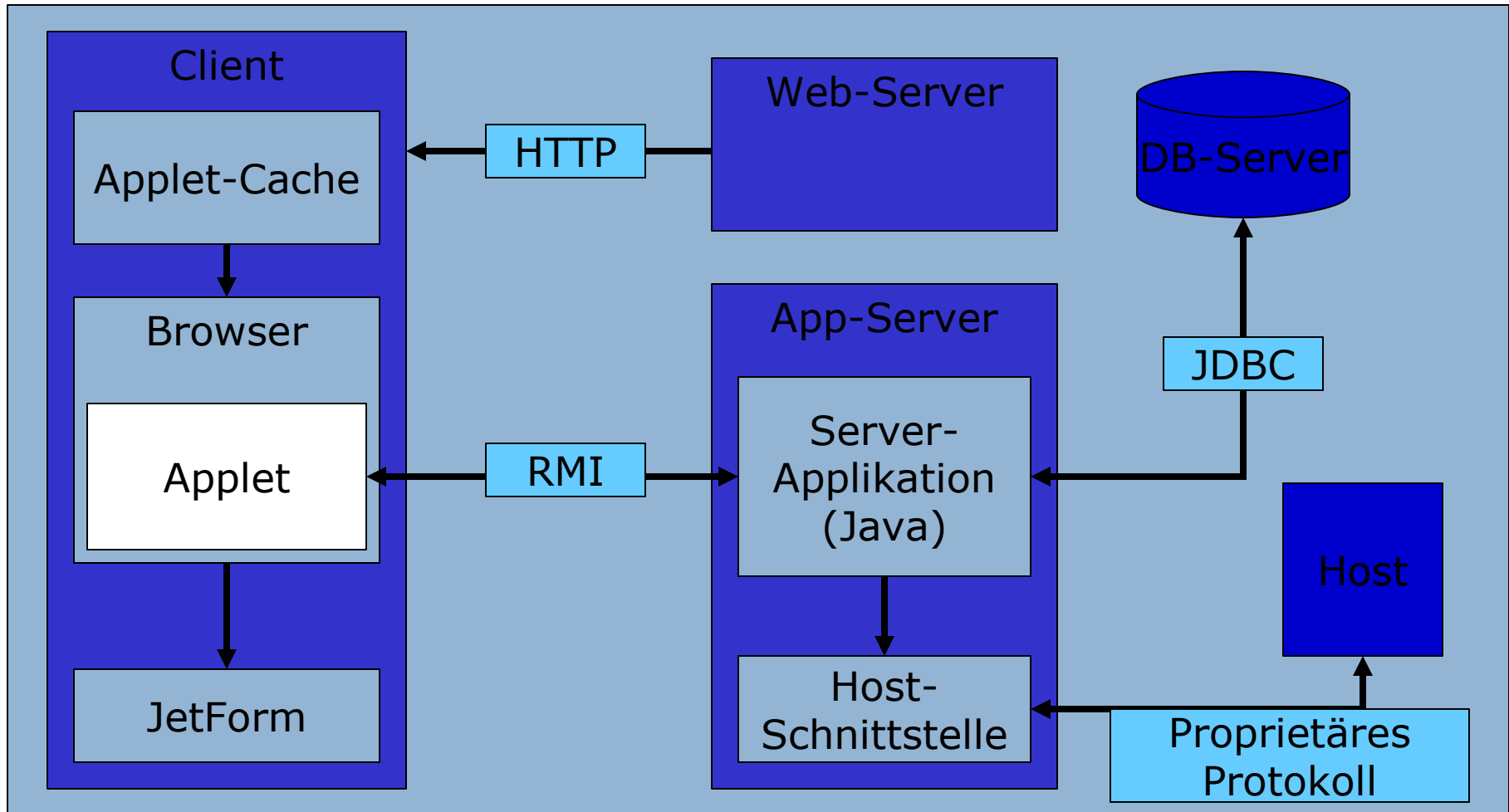
mittels Component-Ware lassen sich sowohl

- Hilfsmittel zur Dienstleistungsentwicklung als auch
- IT-Dienstleistungselemente selber einfach und modular entwickeln

Aufgabendomäne	Hauptkomponenten
Dokumentenmanagement	Imaging, Retrieval, Archivierung, Editoren
CSCW: Teamunterstützung	eMail, Arbeitslisten und Terminkalender, Konferenzsysteme, Telekooperation
Geschäftsprozess- Unterstützung	Vorgangsunterstützung und -steuerung, transaktionale Kammern, Optimierungs- und Simulationssysteme
Multimedia-Systeme	Objektarchive, Multimediaarchive, Animationssysteme, virtuelle Welten
Wissensmanagement	Agenten, Wissensaufbereitung, Klassifikation, Wissensrepräsentation und -extraktion
Kundenmanagement	Kundeninteraktionssysteme, Helpdesk, Support- Systeme, Relationship Management, Beschwerdemanagement

Technologien: Component-Ware

- Anwendungsbeispiel: JavaBeans
 - Client-Seite: Swing + erworbene Beans + eigene Beans

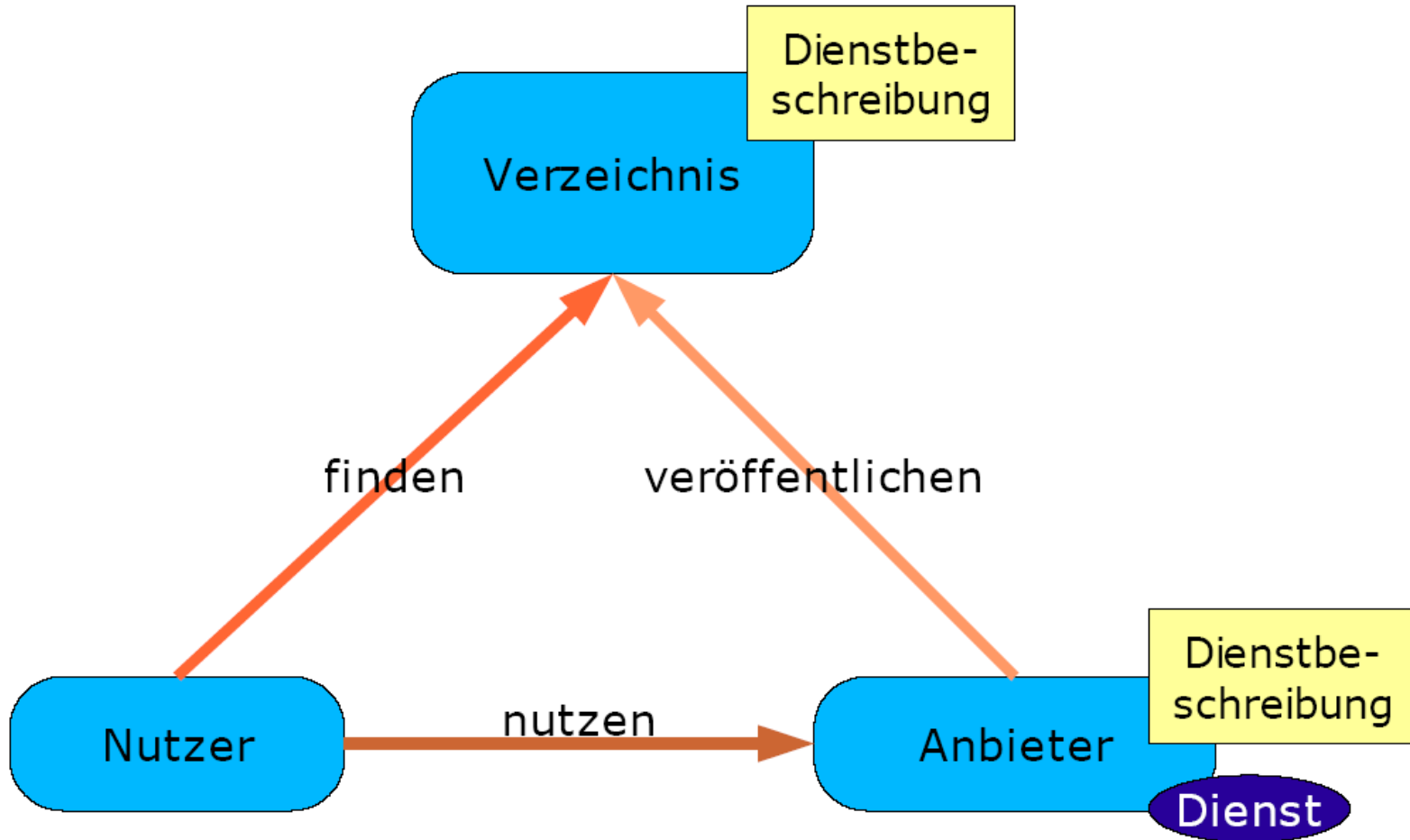


Technologien: Webservices

- Webservices sind
 - Softwareanwendungen
 - durch eine URI identifiziert
 - können über XML definiert, aufgerufen und gefunden werden
 - unterstützen direkte Interaktion mit anderen Webservices über XML-basierten Nachrichtenaustausch
- ein Standard des W3C-Konsortiums
- benutzen
 - SOAP (Simple Object Access Protocol)
 - WSDL (Web Service Description Language)
 - UDDI (Uniform Description, Discovery and Integration)
- basieren auf XML und HTTP

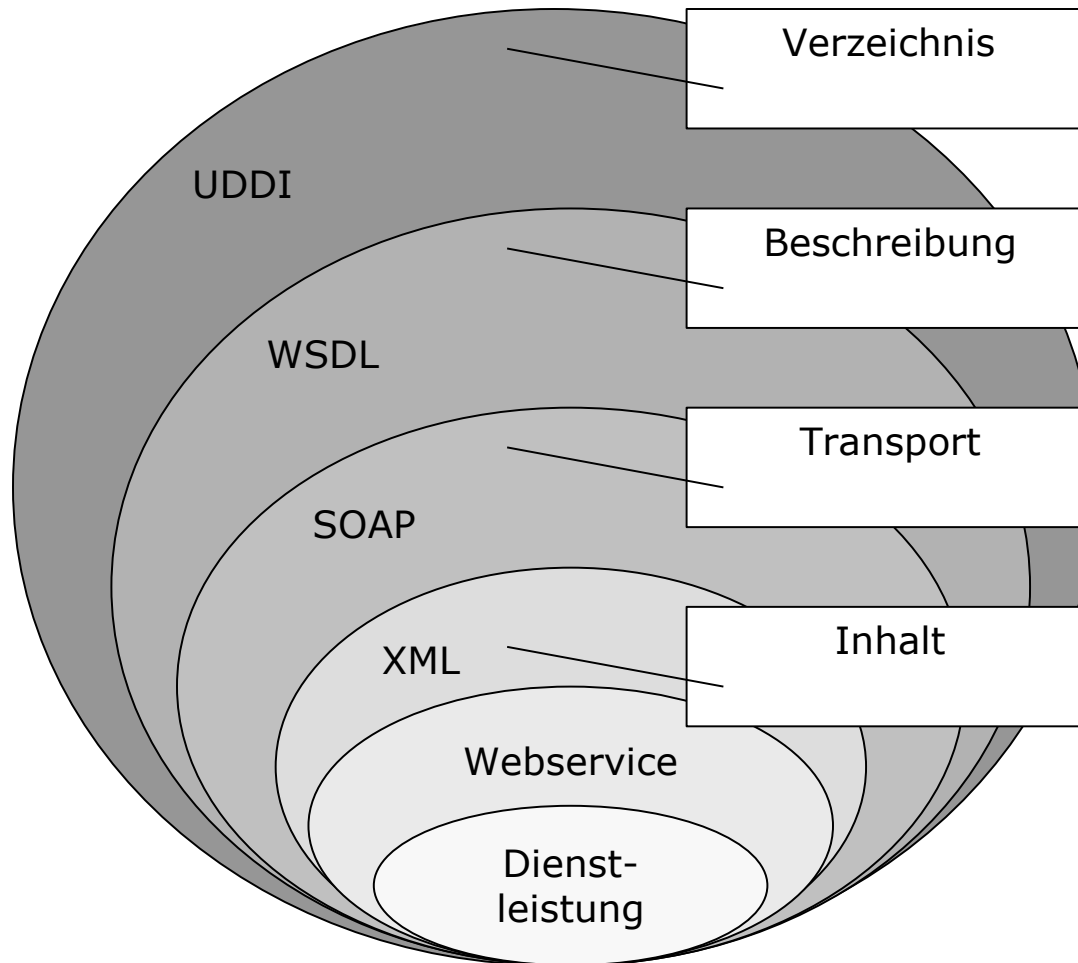
Technologien: Webservices

- Architektur von Webservices



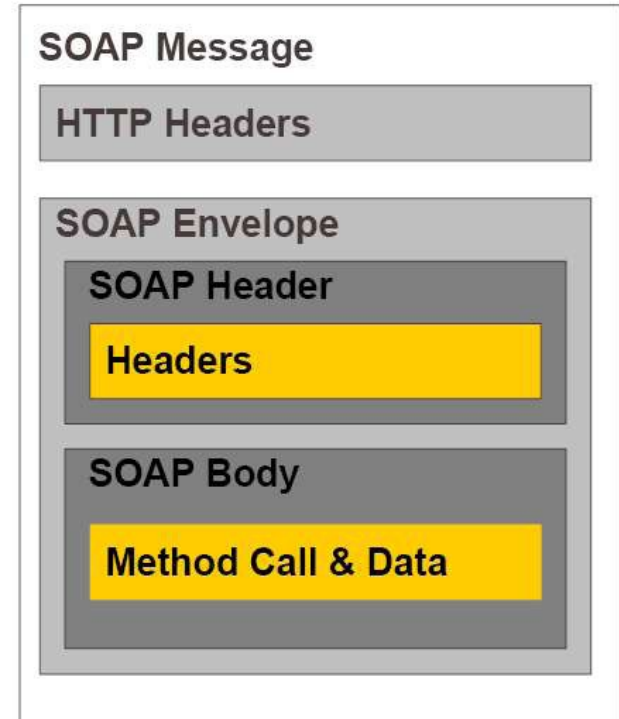
Technologien: Webservices

- Zwiebelschalenmodell:



Technologien: SOAP

- "Simple Object Access Protocol"
- Spezifikation eines Kommunikationsprotokolls, das es erlaubt, Funktionen (Methoden) von Servern, Diensten, Komponenten und Objekten aufzurufen
- verwendet HTTP und XML
- wurde für die bestehende Internet-Infrastruktur entwickelt
- plattformunabhängig
- sprachunabhängig
- kann in verschiedensten Systemen für Anwendungen von Nachrichtendiensten bis RPC (entfernte Funktionsaufrufe) verwendet werden
- einfach und erweiterbar
- im Grunde „nur“ ein Nachrichtenaustauschformat



Technologien: WSDL

- eine XML-Grammatik
- beschreibt Bindungen und Schnittstellen von Webservices
- besteht aus den grundlegenden Elementen
 - `<types>`: XML-Schema-Definition von Datentypen, die verwendet werden
 - `<message>`: definiert die möglichen Nachrichten, die der Webservice akzeptiert
 - `<portType>`: legt Operationen fest, die bestimmte Nachrichten als Ein- oder Ausgabe erhalten
 - `<binding>`: ordnet definierten Operationen ein Protokoll zu, das zur Kommunikation verwendet werden soll, also z.B. HTTP oder SMTP, und legt die Kodierung der Nachrichten fest
 - `<services>`: beschreibt alle externen Schnittstellen, die dieser Dienst hat

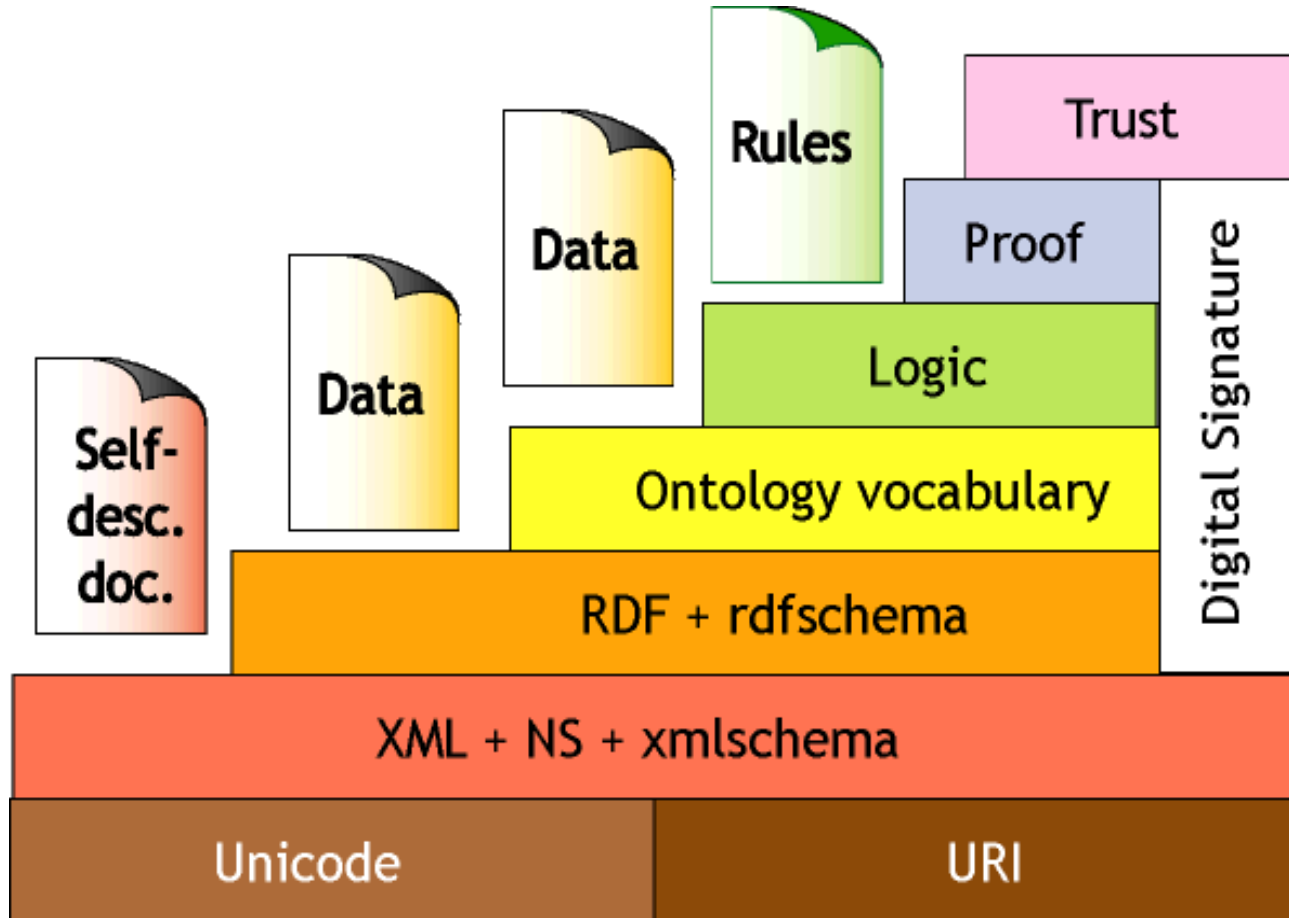
Technologien: UDDI

- ist selbst ein Webservice mit einer normierten Schnittstelle
- dient zum Registrieren und Finden von Webservices
- benutzt WSDL-Dokumente, um Webservices zu registrieren
- bietet ein zentrales Repository für Dienstanbieter
- bietet in Version 3 u.a.
 - Unterstützung für digitale Signaturen
 - Publish/Subscribe-Mechanismen
 - Multi-Registry-Unterstützung (root/affiliate)

Technologien: XML, RDF, OWL, ...

Knowledge Engineering und Semantic Web

Tim Berner-Lee's semantische Pyramide



Basistypen

RDF unterscheidet 2 fundamentale Grundtypen:

Ressourcen

- Komplexe abstrakte oder konkrete Entitäten
- Eindeutig durch URI charakterisiert

Literale

- Datentyp
- Sprache

Technologien: RDF**RDF Statements**

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

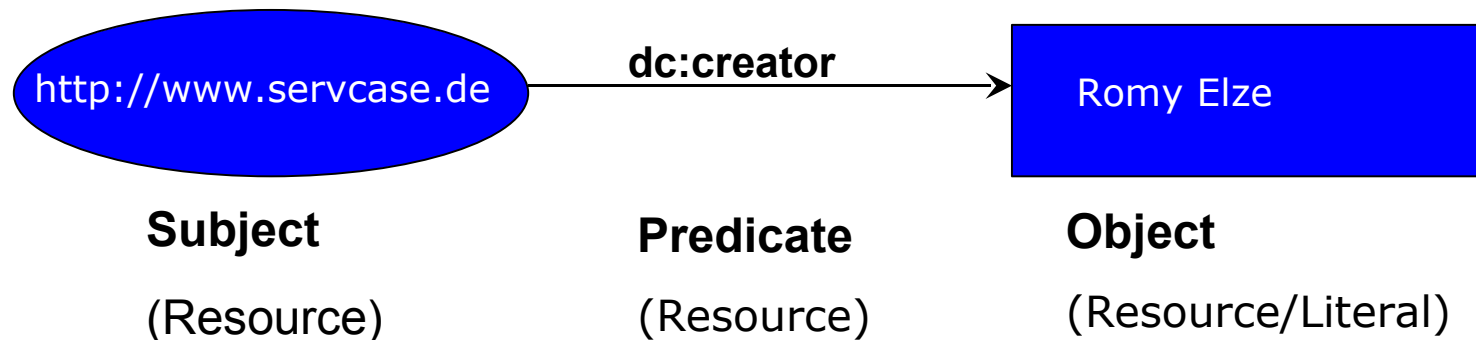
```
  xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/metadata/dublin_core#">
```

```
  <Description about="http://www.servcase.de">
```

```
    <dc:Creator> Romy Pfretzschner </DC:Creator>
```

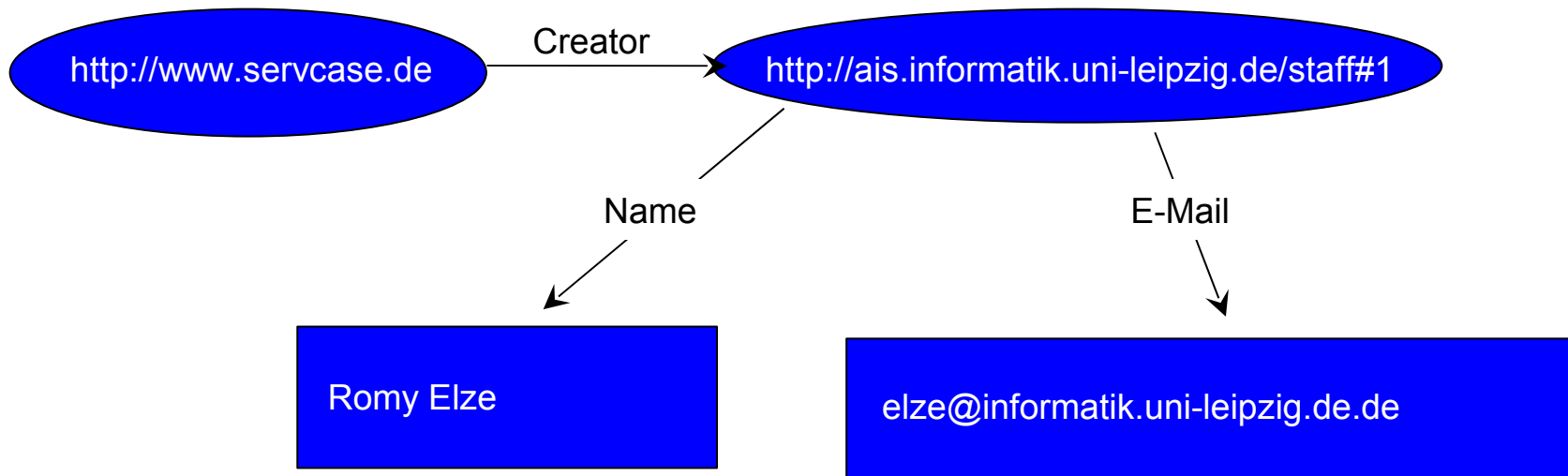
```
  </Description>
```

```
</rdf:RDF>
```



RDF Model

- Einfache Wissensbasis
- Kombiniert mehrere RDF Statements



Technologien: RDF – Beispiel

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/metadata/dublin_core#">
```

```
<rdf:Description about="http://www.servcase.de">
```

```
  <dc:Creator>
```

```
    <rdf:Description>
```

```
      <rdf:Description about="http://ais.informatik.uni-
leipzig.de">
```

```
        <v:Name>Romy Pfretzschner</v:Name>
```

```
        <v:Email>pfretzschner@informatik.uni-
leipzig.de</v:Email>
```

```
      </rdf:Description>
```

```
    </dc:Creator>
```

```
  </rdf:Description>
```

```
</rdf:RDF>
```

Anwendung

- „Eingebettet“
 - in HTML/XML bzw. beliebige andere Datenformate
 - z.B. PDF / OpenOffice / AVI / P3P
- „Standalone“
 - Beschreibung einer oder mehrerer „Ressourcen“
 - Beschreibung von Domainwissen
(z.B. zur weltweiten Nutzung im Web bereitgestellt)

Container

- Problem:
 - Dokument mit mehrere Autoren
 - eine Person hat mehrere Emailadressen
 - Anweisungen müssen in definierter Reihenfolge bearbeitet werden
- Lösung:
 - Zusammenfassung mehrerer Ressourcen / Literale in einem Container
- Drei Typen:
 - bag - ungeordnete Liste (mit Duplikaten)
 - sequence - geordnete Liste (mit Duplikaten)
 - alternative – Alternative (ohne Duplikate)

Container Beispiel

...

```
<DC:Creator>
```

```
<Bag>
```

```
<li>Romy</li>
```

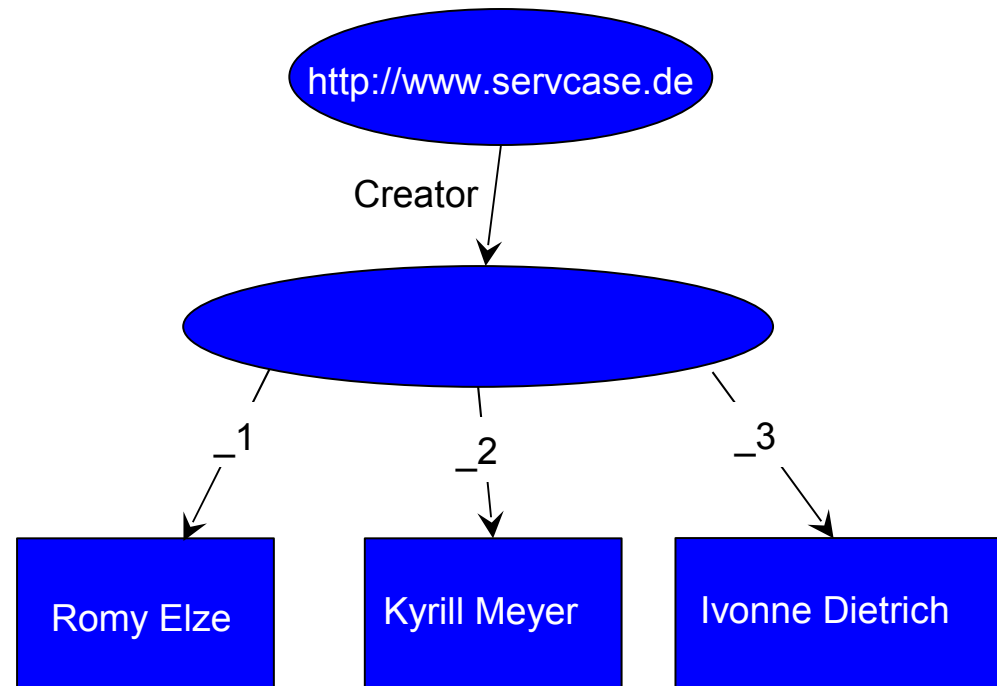
```
<li>Kyrill</li>
```

```
<li>Ivonne</li>
```

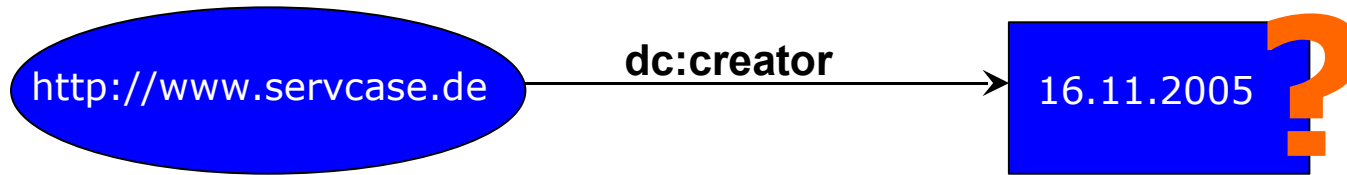
```
</Bag>
```

```
</DC:Creator>
```

...



Technologien: RDF Schema



- Einschränkung der Verknüpfbarkeit von Ressourcen / Literalen
- Strukturierung von Vokabularen
- Instanziierung / Klassifikation
- Bereitstellung spezieller Ressourcen:
 - Klassen (Konzepte, Frames)
<http://www.w3.org/2000/01/rdf-schema#Class>
 - Attribute (Eigenschaften, Properties, Slots, Roles)
<http://www.w3.org/2000/01/rdf-schema#Property>
 - Instanzen (Objekte)
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

Technologien: RDF Schema

Klassen und Eigenschaftenhierarchien

Klassen und Eigenschaften können hierarchisch strukturiert werden

```
<rdfs:Class rdf:ID="Person">
```

```
<rdfs:comment>The class of people.</rdfs:comment>
```

```
<rdfs:label language=„en_US">Person</rdfs:label>
```

```
<rdfs:label language=„de_SN">Dor Mänsch</rdfs:label>
```

```
<rdfs:subClassOf rdf:resource=
```

```
"http://www.w3.org/2000/03/example/classes#Animal"/>
```

```
</rdfs:Class>
```

Technologien: RDF Schema

Eigenschaften

Werden unabhängig von Klassen definiert verwandt

```
<rdf:Property ID="verheiratet">  
  <rdfs:subPropertyOf rdf:resource="verwandt" />  
  <rdfs:domain rdf:resource="#Person" />  
  <rdfs:range rdf:resource="#Person" />  
</rdf:Property>
```

Domain: Zuordnung zu einer oder mehreren Klassen

Range: Werte welche die Eigenschaft annehmen kann

- Instanzen einer bestimmten Klasse
- Literale eines XML-Schema Datentyps

Technologien: RDF Schema

Instanzen

Sind einer (bzw. mehreren) Klasse(n) zugeordnet

```
<rdfs:Class rdf:ID="Familienstand">
```

```
<rdf:Property ID="mindestAlter">
```

```
<rdfs:domain rdf:resource="#Person" />
```

```
<rdfs:range rdf:resource=
```

```
"http://www.w3.org/2001/XMLSchema#nonNegativeInteger" />
```

```
</rdf:Property>
```

```
<Familienstand rdf:ID="Verheiratet">
```

```
<mindestAlter>16</mindestAlter>
```

```
<mindestAlter>18</mindestAlter>
```

```
</Familienstand>
```

Technologien: Web Ontology Language

OWL

reichert RDF-S um weitere Möglichkeiten an:

- Restriktionen (Constraints)
- Kardinalität (min/max) von Eigenschaften
- Identifikation gleicher Ressourcen
- Eigenschaften können transitiv, symmetrisch sein

Äquivalent zu einer entscheidbaren Untermenge der Prädikatenlogik

1. Stufe (Description Logic - SHIQ)

- Automatische Klassifikation
- Konsistenzprüfung
- Einfache Inferenz

Anwendungen: (Vokabulare / Modelle / Ontologien)

Dublin Core

VCard

RSS

DMoz

XUL

SPDL

Knowledge Management – Motivation

Warum Knowledge Management?

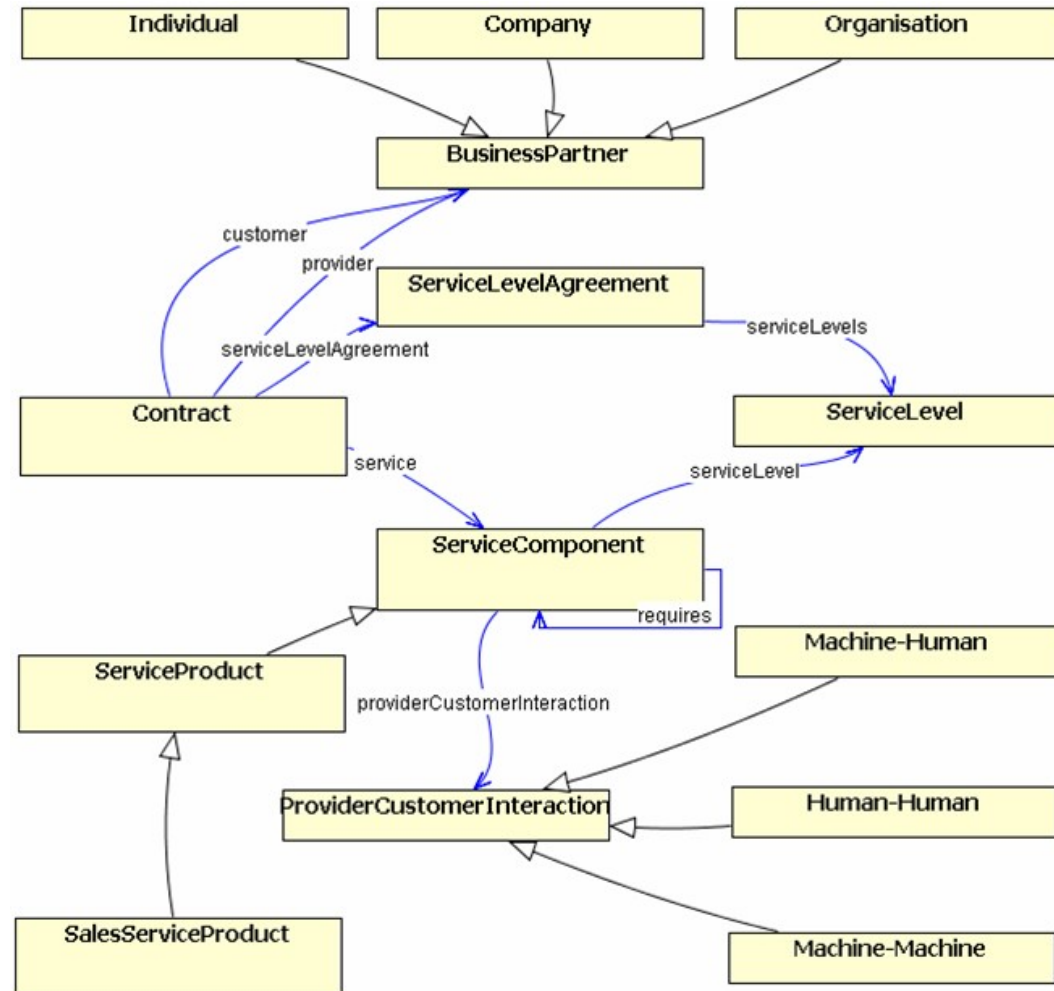
- *Prozesse des Service Engineering sind wissensintensiv, z.B.:*
- Ideenfindung
 - Suche nach bereits früher gefundenen Lösungen
 - Abgrenzung der Domänen, um gefundene Ideen bewerten zu können
- Anforderungsanalyse
 - Strukturierung von vorhandenen Informationen ist notwendig, um den Überblick zu behalten und Entscheidungen über Prioritäten, wichtige Kriterien usw. zu treffen
- Modellierung
 - Semantik der Modelle muss klar definiert sein
- generell:
 - Übertragung von Daten zwischen den Prozessschritten benötigt Meta-Informationen über diese
 - Beziehungen zwischen den Beteiligten des SE-Prozesses und den benötigten Daten

Knowledge Management – Beispielansätze

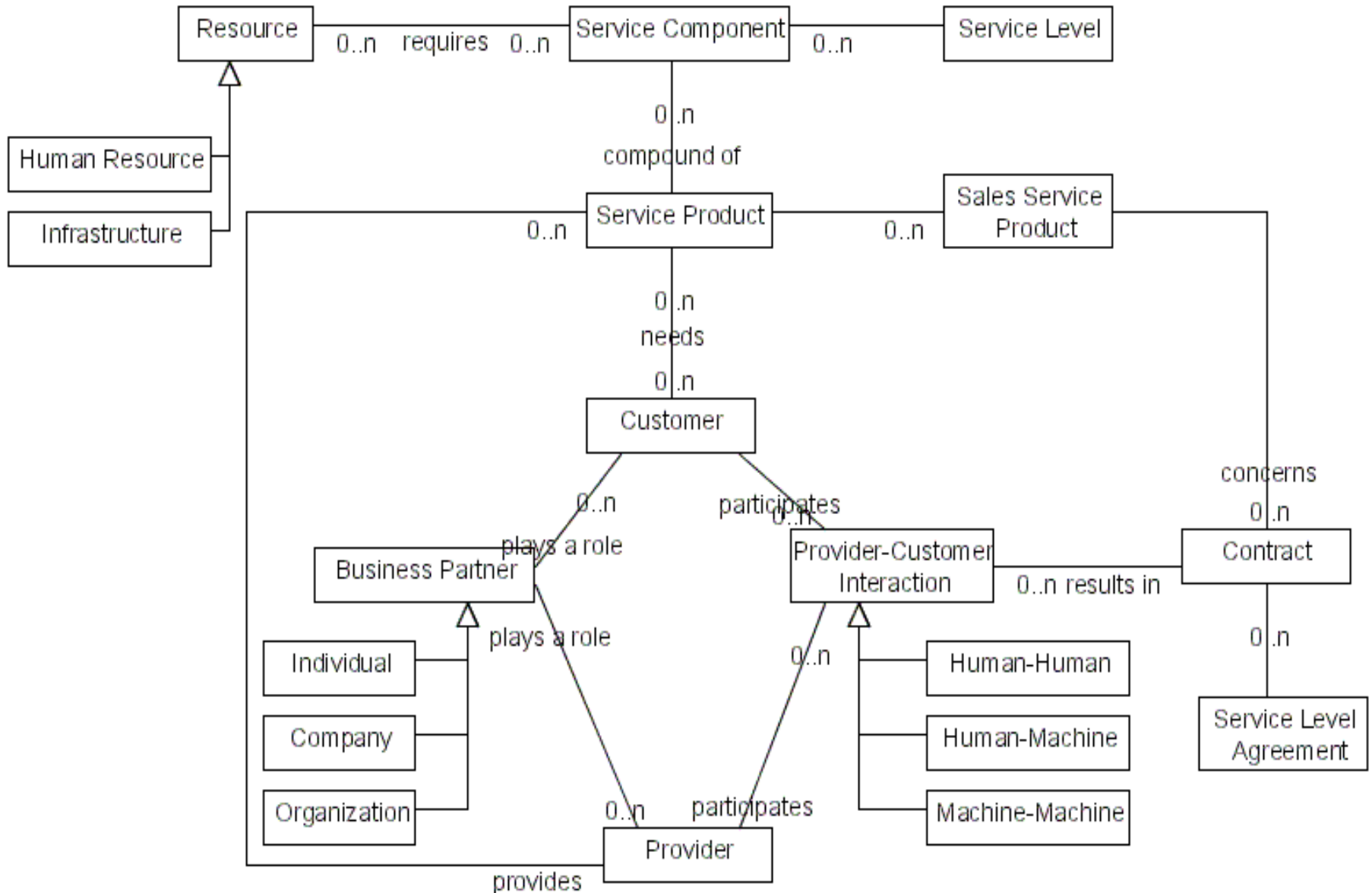
- **SPDL** (<http://www.informatik.uni-leipzig.de/~auer/spdl/>)
 - Service Product Description Language, basiert auf OWL
 - *Ziel*: Sammeln alles Wissens, das im Lebenszyklus von Dienstleistungsprodukten benötigt wird, maximal mögliche Integration mit Webservices
 - *Methoden*: Entwickeln von domänenspezifischen Ontologien, rekursiv-modulare Modellierung von Service-Produkten
 - *Ergebnis*: ein Dokument beschreibt Service-Produkte, Prozesse, Modelle, generell alle zum Service relevanten Informationen

Technologien: SPDL**SPDL – Service Product Description Language**

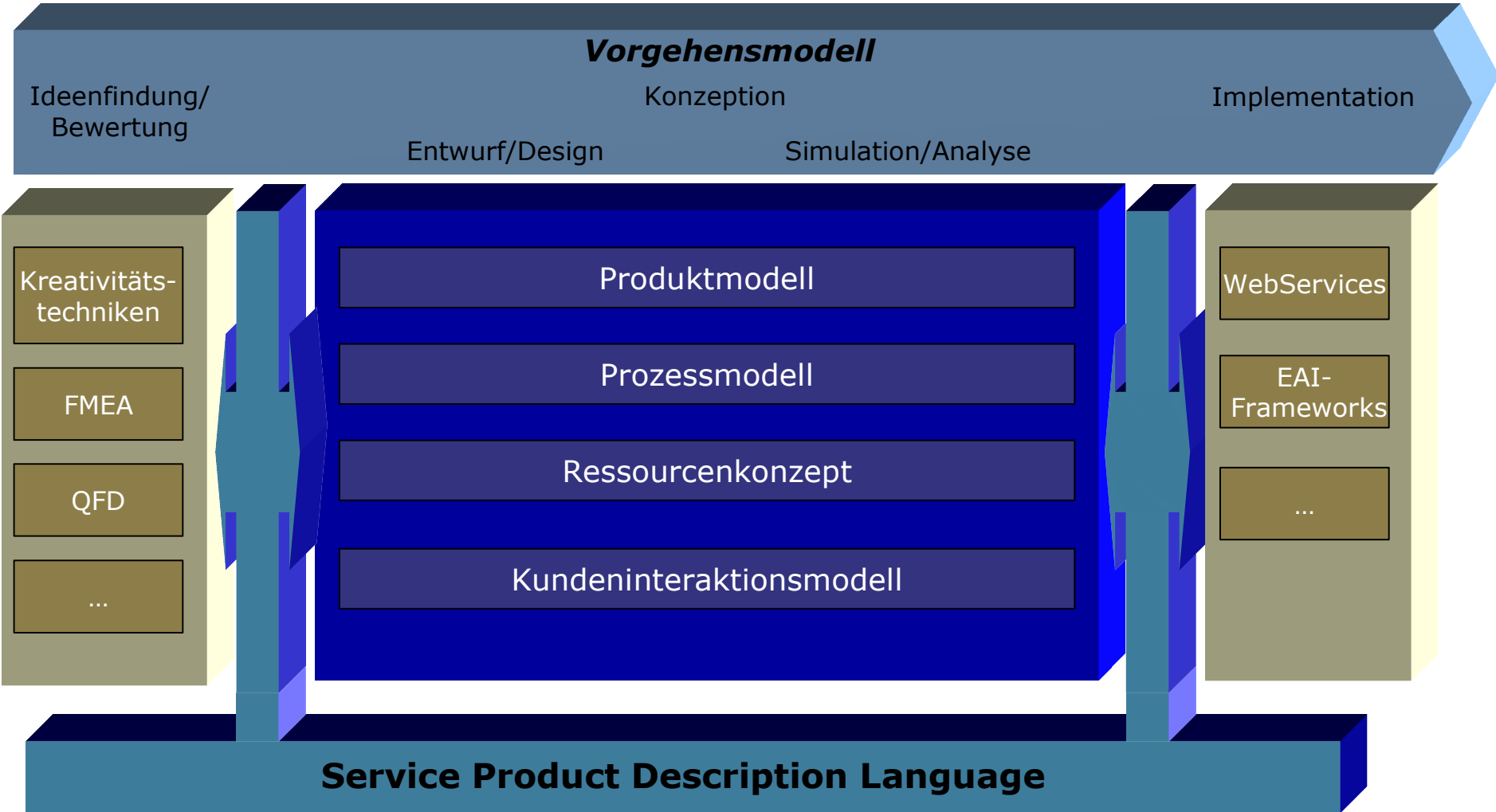
- Eingebettet in die semantische Pyramide:
- ❖ ***XML / Unicode / RDF / RDF Schema / OWL***
- ➔ ***SPDL***
Dienstleistungsbeschreibung mit Produkt-, Ressourcen-, Marketingmodell
- ➔ ***OWL-S / WSDL***
Webservice zentrierte Dienstleistungsbeschreibung

Zentrale SPDL Konzepte:

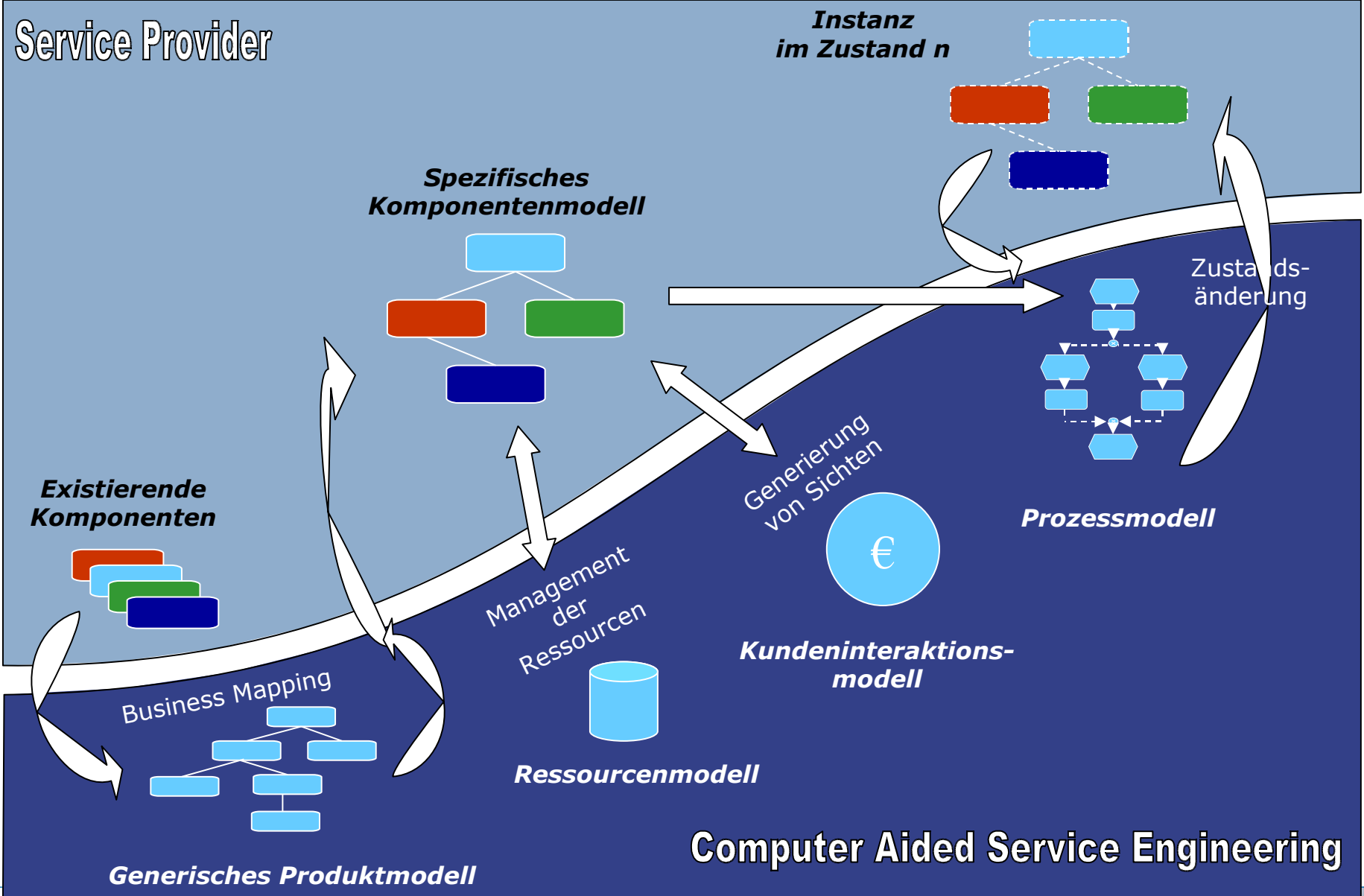
Technologien: SPDL – Ontologie



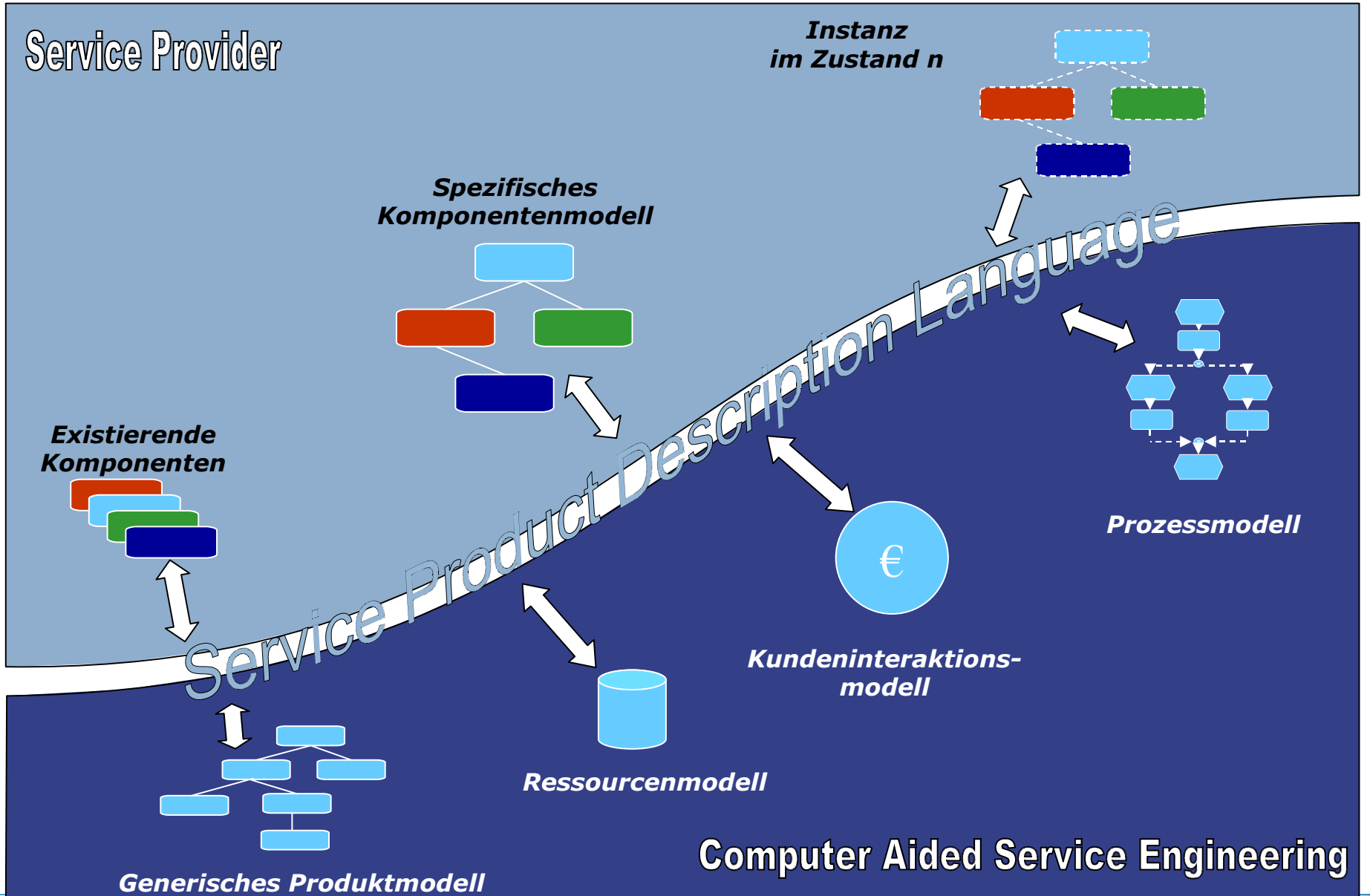
Anwendung von SPDL im Projekt ServCASE



Anwendung von SPDL im Projekt ServCASE

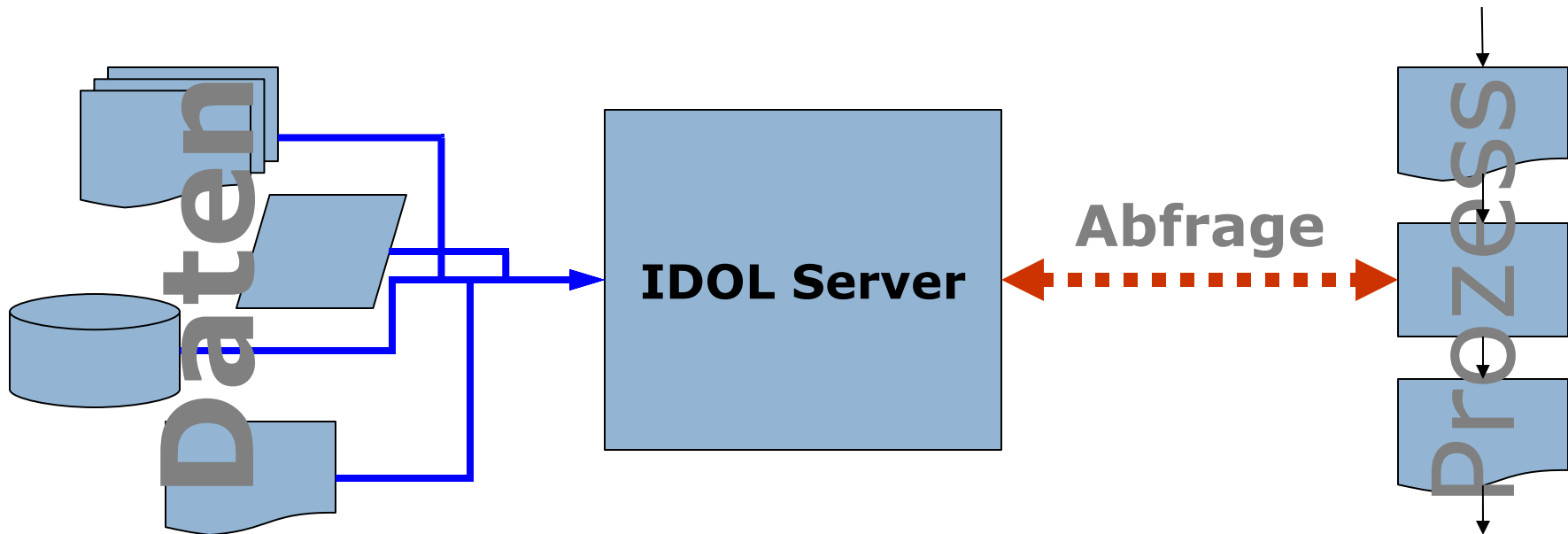


Anwendung von SPDL im Projekt ServCASE



Knowledge Management – Autonomy

- **Autonomy's** Ansatz: (www.autonomy.com)
 - „making sense of an unstructured world“
 - *Ziel*: automatische Strukturierung und Klassifizierung von unstrukturierten Daten
 - *Methoden*: Dokumentenanalyse mittels Shannon's Informationstheorie und Bayesscher Inferenztheorie
 - *Ergebnis*: prozessbegleitende Unterstützung bei der Deckung des Informationsbedarfs



Technologien: Content Management

- Verknüpfung von Dokumenten mit ERP/DB-Daten und sonstiger Information (z.B. via Kunden- oder Auftragsnummer)
- dezentrale Informationsbereitstellung (unterschiedliche Pools)
- personalisierbare Informationsbereitstellung (sowohl nach Adressat als auch nach Inhalt und Layout)
- Prozessmanagement durch Vorlagenmanagement