

Ontology-based Web Service Composition: Part I

Rolland Brunec
Betreuerin: Sabine Maßmann
Universität Leipzig, Abteilung Datenbanken

Motivation

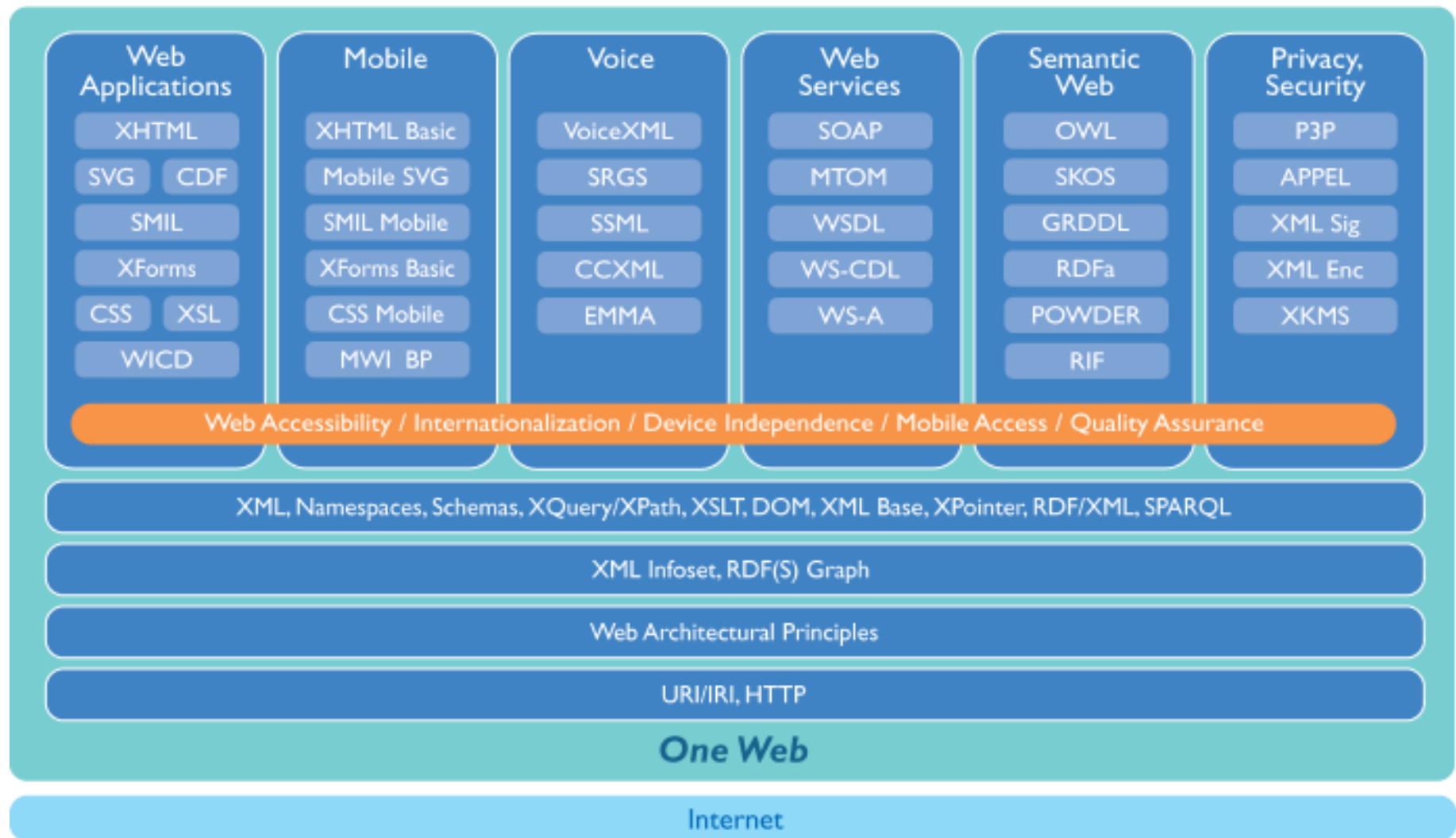
- Semantic Web
- Web Services
- Web Service Composition
- Web Services Composition + Semantic Web

Outline

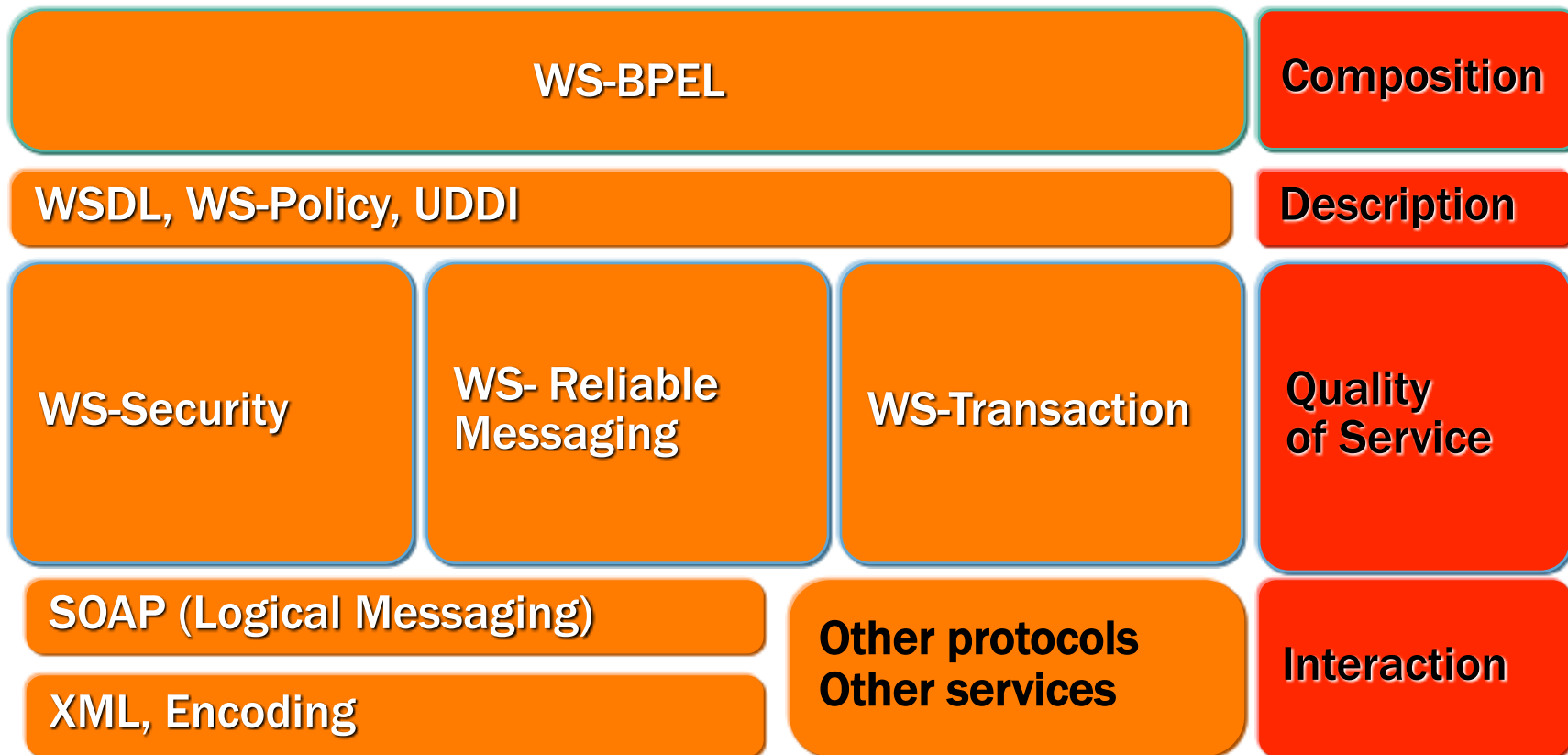
- Semantic Web
- Web Services
- Semantic Web Services
- Tools

Ontologies and the Semantic Web

- Machine-readable format
- “Understanding” of symbols
 - **Human:** Meaning is represented by the term itself (natural language + semantic relationship between terms “*Student is-a Person*”)
 - **Machine:** formal description of relationships
- Automated services



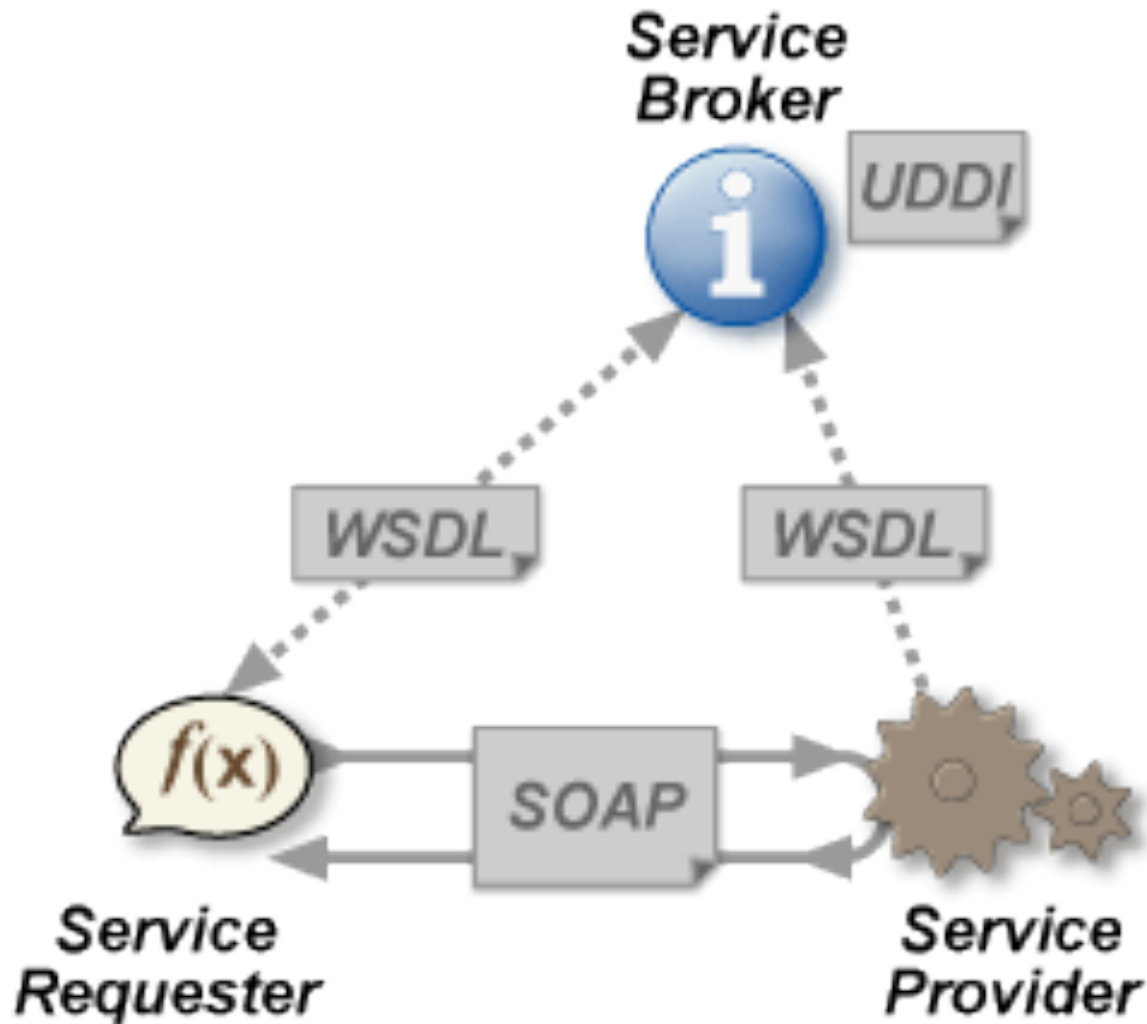
The W3C Technology Stack [w3c.org]



Web Service Standards [K.Verma, A. Sheth]

Web Services

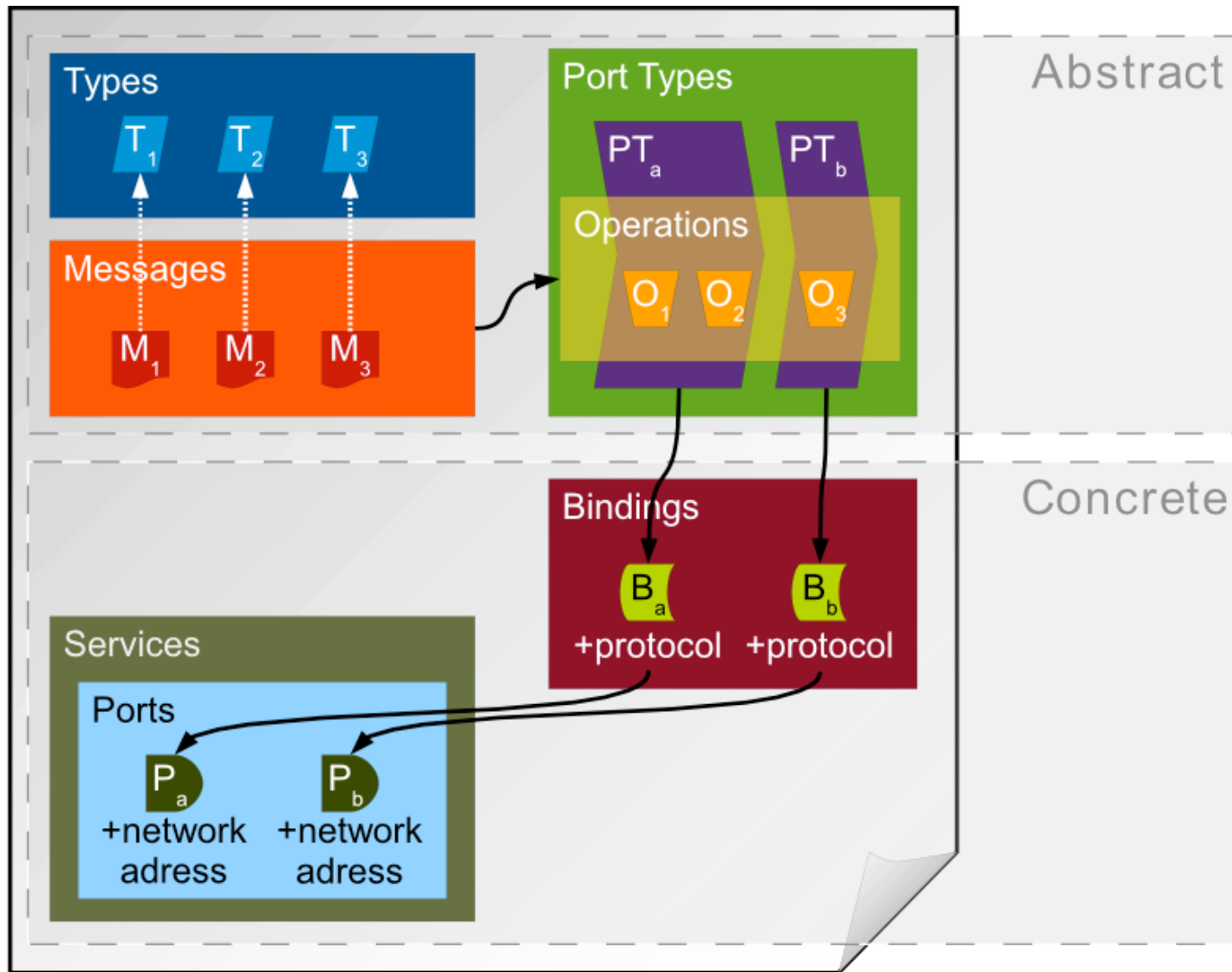
- A **Web Service** is a software system designed to support interoperable machine-to-machine interaction over a network
- Web API
- Application areas of Web Service technology
 - Enterprise application integration
 - E-commerce



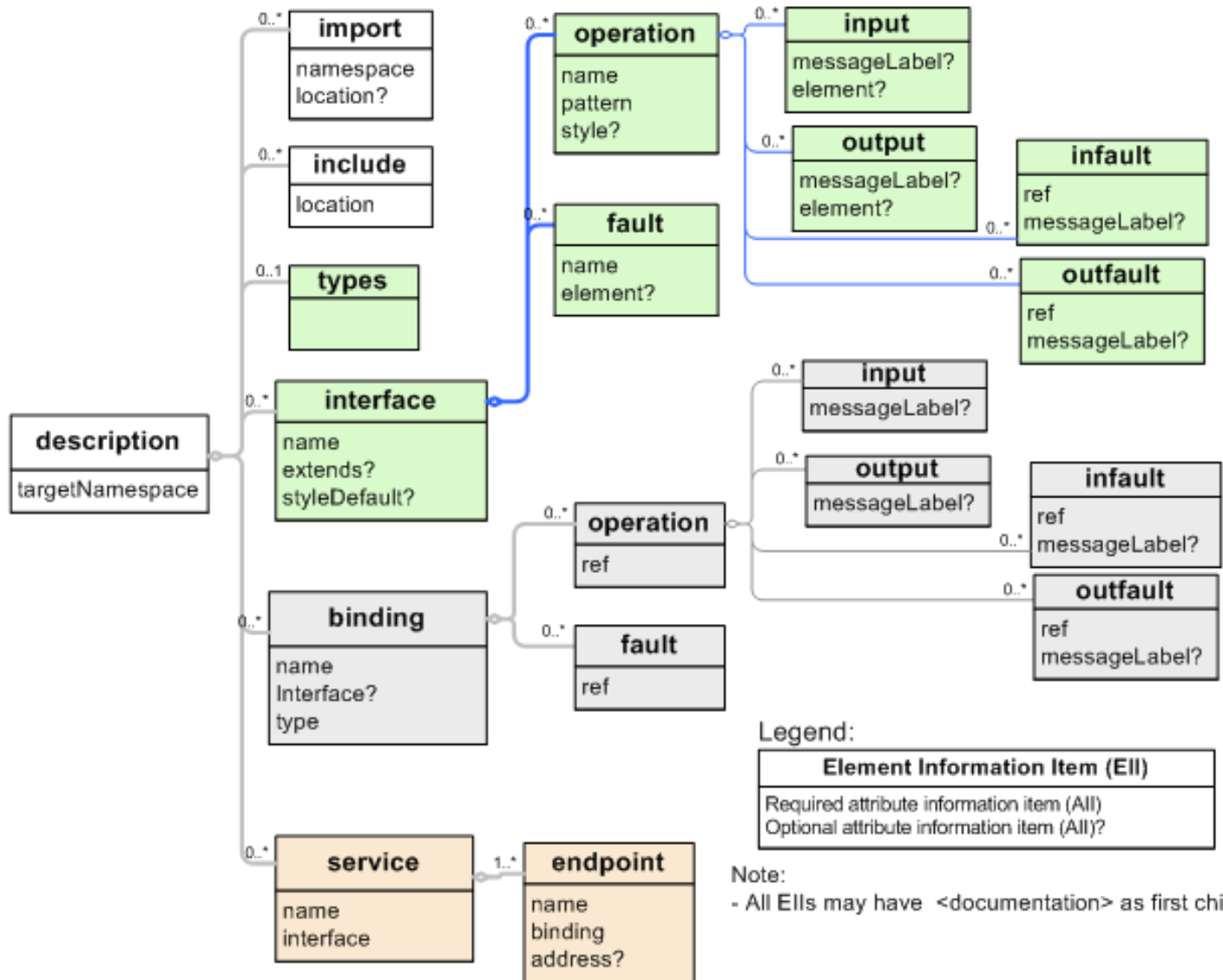
Web Service [Voormann]

WSDL

- An extensible, platform independent XML-based language for describing services
- Public interface to the web service
- Client reads WSDL and calls one of the functions listed using SOAP



Representation of concepts defined by a WSDL 1.1 document [Le Merdy]



Legend:

Element Information Item (EII)
Required attribute information item (All)
Optional attribute information item (All)?

Note:
- All EIIs may have <documentation> as first child

WSDL 2.0 Component Model

[<http://www.w3.org/TR/wsd120-primer>]

SOAP

- Simple Object Access Protocol
 - Messaging framework for for peers communication
 - XML-based
- Exchanging structured information
- Internet application layer protocol
 - Uses HTTP(S) and SMTP for transport
 - Other distributed protocols (DCOM): filtered by firewalls

Example SOAP message

```
<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <GetWeather xmlns="http://www.weather.com">
    <CityName>Leipzig</CityName>
    <CountryName>Germany</CountryName>
  </GetWeather>
</soap:Body>
</soap:Envelope>
```

Publishing and Discovering

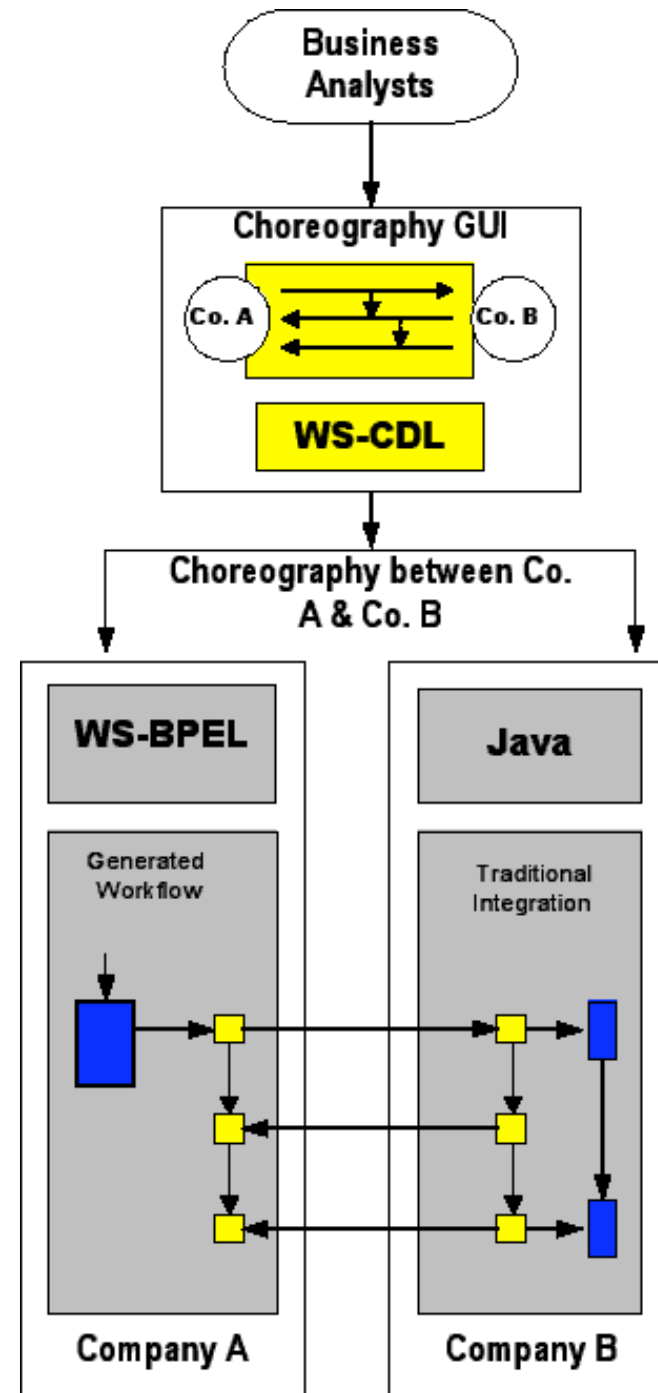
- Universal Description, Discovery and Integration (UDDI)
 - Offers an interface which allows business partners and service providers to dynamically find one another
 - In 2006 Microsoft, IBM, SAP closed their public UDDI nodes (Intranet vs Internet)
- UDDI is integrated into WS-I
- Web Services Interoperability (WS-I)
 - Consortium for promoting interoperability amongst WS specifications
 - Creates guidelines and tests for interoperability

WS Choreography

- Interaction of services with users
- Not executable (allows many realizations)
- Can be realized by writing orchestration
- WS-CDL (Web Services Choreography Description Language): describes how peer-to-peer participants collaborate

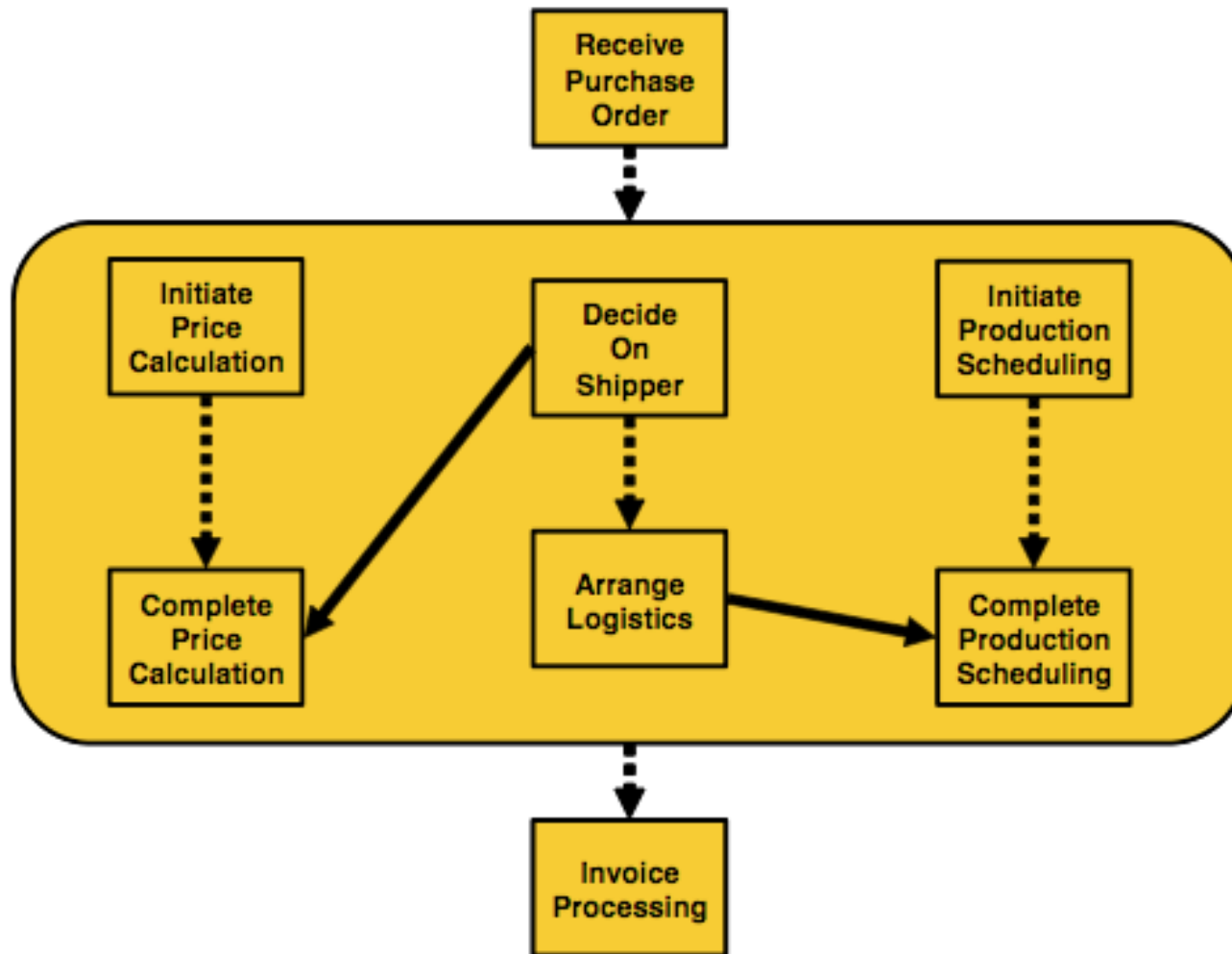
WS-CDL

- Company A and B wish to integrate their Web Services based applications
- Business analysts at both companies agree upon the services involved in the **collaboration**, their **interactions**, and their common ordering and constraint **rules** under which the interactions occur
- Generate a WS-CDL based representation
- Choreography specifies the interactions between services across business entities ensuring interoperability, while leaving actual implementation decisions in the hands of each individual company
- Source: <http://www.w3.org/TR/ws-cdl-10/>



WS Orchestration

- Executable Process
- Sequence and conditions of invocation
- Message exchange sequences controlled by orchestration designer
- WS-BPEL (Web Services Business Process Execution Language): executable language for specifying interactions with Web Services



WS Orchestration Example

[<http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>]

What's missing?

- Computer can *understand* the information
- Automatic processing of Web Services
- **Discovery** (decentralized, semantic annotation of service capabilities)
- **Composition** and behavioral description
- **Invocation** of composed services
- → Semantic Web services

What does Semantics brings to the table?

- Better **Reuse** and **Discovery**
- Better Interoperability
- Configuration/**Composition**: Enable dynamic binding of partners
- Automation across process life cycle
 - Process Configuration (Discovery and Constraint analysis)
 - Process Execution

WSDL-S

- Extension to WSDL 2.0
- Should improve the WS by
 - facilitating the composition of services
 - improving discovery
 - enabling the integration with a Web Services framework

OWL-S

- Semantic Markup for Web Services
- Ontology for describing Semantic WS
- Automatic Web service
 - discovery
 - invocation
 - composition and interoperation
- OWL-S 1.2 Pre-Release (2006)

Semantic Annotations for WSDL (SAWSDL)

- W3C Candidate Recommendation
- Defines a mechanism to add semantic annotations to Web services description in WSDL 2.0
- Specifies a way to describe the abstract functionalities of a service and concretely how and where to invoke it
- Key design principles
 - Enables semantic annotations for Web services
 - Agnostic to semantic representation languages
 - Enables semantic annotations for WVS for discovering and invoking

Web Service Modeling Ontology (WSMO)

- Includes support for *choreography* using Abstract State Machine and Introduces the *orchestration* element
- Web Service Modeling Language (WSML) formalizes the WSMO
 - WSML-Core: Description Logic and Horn Logic
 - WSML-DL: fully captures the Description Logic SHIQ(D)
 - WSML-Flight: based on Frame Logic
 - WSML-Rule: extension of WSML-Flight
 - WSML-Full: unifies WSML-DL and WSML-Rule

Semantic Web Services Framework (SWVSF)

- Semantic Web Services Language (SWSL): formal characterizations of WS concepts and descriptions of services
 - SWSL-FOL: express the formal characterization of WS concepts
 - SWSL-Rules: support the use of the service ontology in reasoning and execution environments
- Semantic Web Services Ontology
 - conceptual model for WS description
 - Axiomatization in SWSL-FOL → First-Order Logic Ontology for Web Services (FLOWS)
 - FLOWS axioms translated into SWSL-Rules → Rules Ontology for Web Services (ROWS)

The Roman Model

- Software module capable of performing *stateful* operations (possible conversations between client and server)
- Based on the state it offers its clients a choice of operations it can perform
- Includes an orchestrator (system component)
 - Is able to activate, stop and resume the available services

WSMO Studio

- SWS and Semantic Business Process modelling environment for WSMO
- Java & Eclipse based
- Components
 - WSMO editor
 - Choreography designer
 - SAWSDL editor
- <http://wsmostudio.org/>

IBM Tools

- WebSphere Integration Developer
- Supported features
 - Web Service Interface Matching
 - Web Service Discovery
 - Web Service Composition
- WSDL-S and SAWSDL
- <http://www.alphaworks.ibm.com/tech/wssem>

Conclusions

- Introduction to Web Services
- Web Service composition
- Semantic Web and Web Services
- Semantic Annotations
- Ontologies
- Tools

References

<http://www.sciam.com/article.cfm?id=the-semantic-web>

<http://www.w3.org/Submission/SVVSF/>

<http://www.w3.org/Submission/WSDL-S/>

<http://www.daml.org/services/swsf/1.0/swso/>

<http://www.w3.org/2002/ws/sawSDL/>

Enabling Semantic Web Services: The Web Service Modeling Ontology (Fensel, Lausen, de Bruijn, Stollberg, Roman, Polleres)

<http://www.w3.org/TR/wsdl20>

<http://www.w3.org/TR/ws-gloss>

<http://www.w3.org/Submission/OWL-S>

Semantic web services and web process composition (SWSWPC San Diego)

Enabling semantic web services (Fensel)

Web-Technologien (Badach)

Automatic Service Composition and Synthesis: the Roman Model (Calvanese)