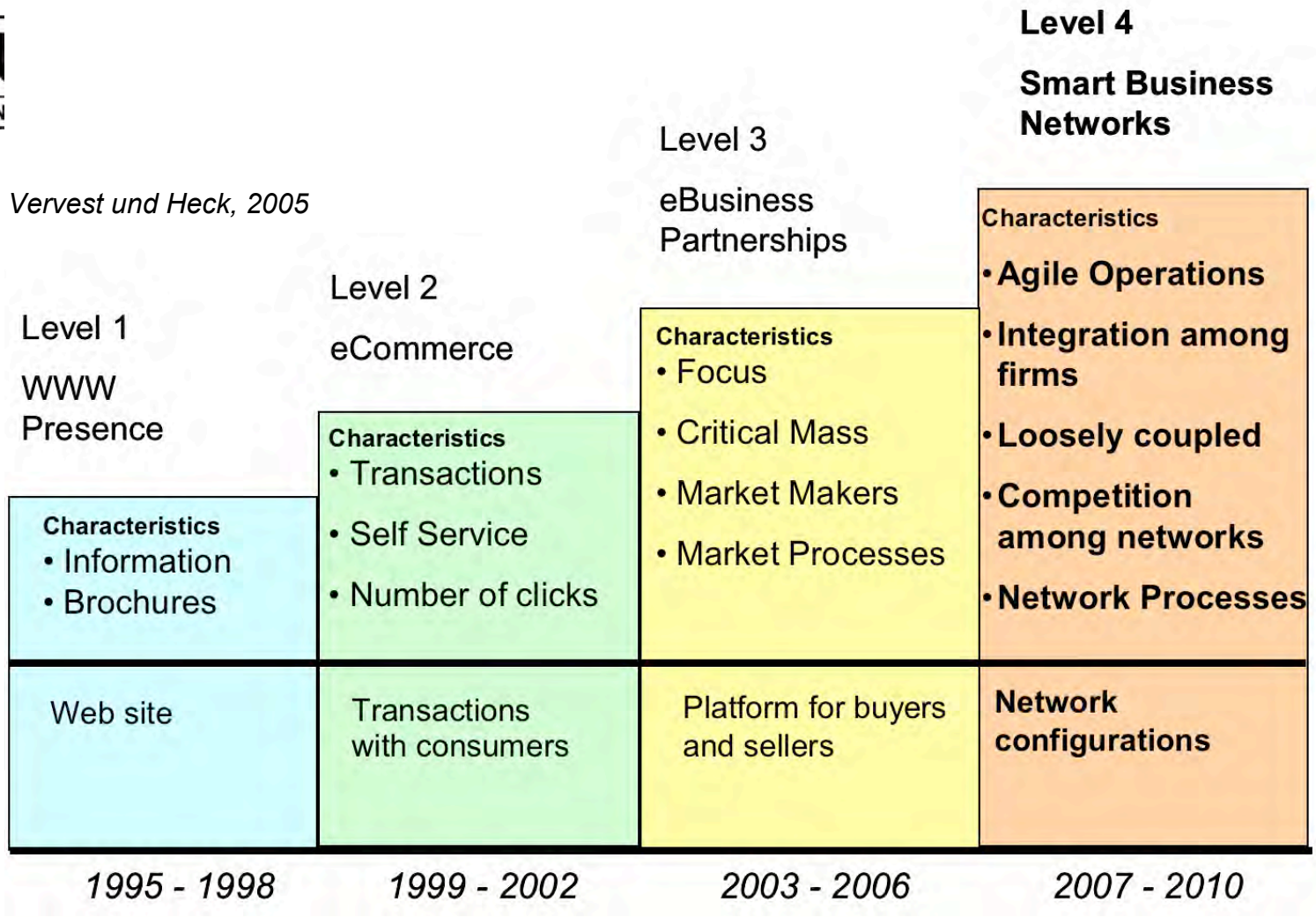
Service Oriented Architectures
Visions – Concepts – Reality

CSC March 2006

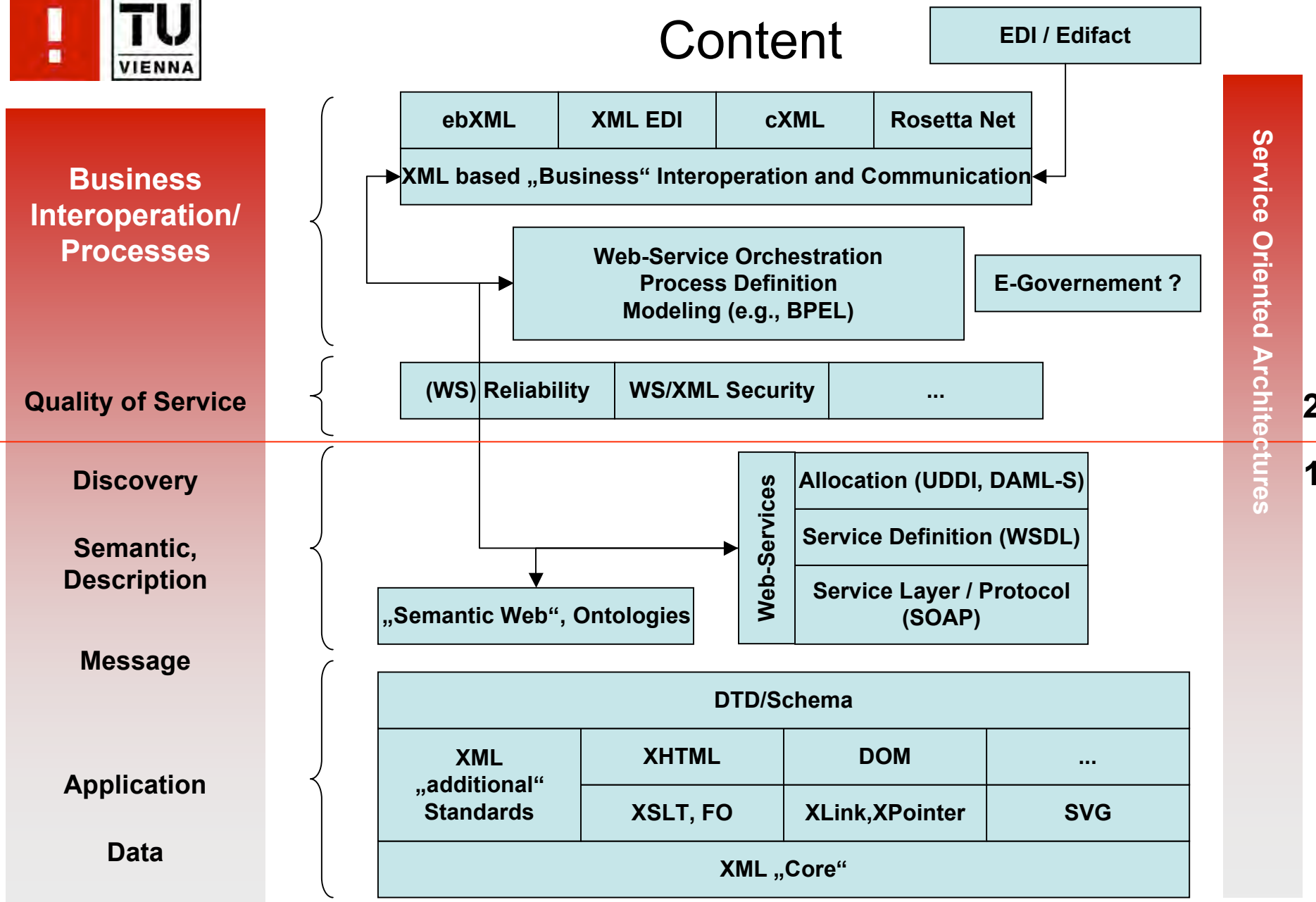
Alexander Schatten
Vienna University of Technology



„A Service Oriented Architecture enhanced by semantics, would permit a degree of dynamic discovery, choreography and adaption. Hence a SOA would [...] emerge dynamically depending on the policies and models that govern the relevant service“

Michael Brodie et.al.

Content


EDI / Edifact
ebXML
XML EDI
cXML
Rosetta Net
XML based „Business“ Interoperation and Communication
Business Interoperation/ Processes
Web-Service Orchestration
Process Definition
Modeling (e.g., BPEL)
E-Governement ?
Quality of Service
(WS) Reliability
WS/XML Security
...
Service Oriented Architectures
2
1
Discovery
Semantic, Description
Message
Application
Data
Web-Services
Allocation (UDDI, DAML-S)
Service Definition (WSDL)
Service Layer / Protocol (SOAP)
„Semantic Web“, Ontologies
DTD/Schema
XML „additional“ Standards
XHTML
DOM
...
XSLT, FO
XLink,XPointer
SVG
XML „Core“

Looking back ...

- Traditional distributed computing
  - Client / Server Model
  - Distributed Object Model
    COM, CORBA, Java RMI and EJB
- What's wrong with that?
  - Distributed object models do not scale to the Internet
    - Tightly coupling service and customer
    - Need for homogeneous infrastructure
    - Versioning problems
  - Limited COM support on non-Windows platforms
  - Generally spoken: conventional technologies are usually bound to specific platforms/languages/operating systems

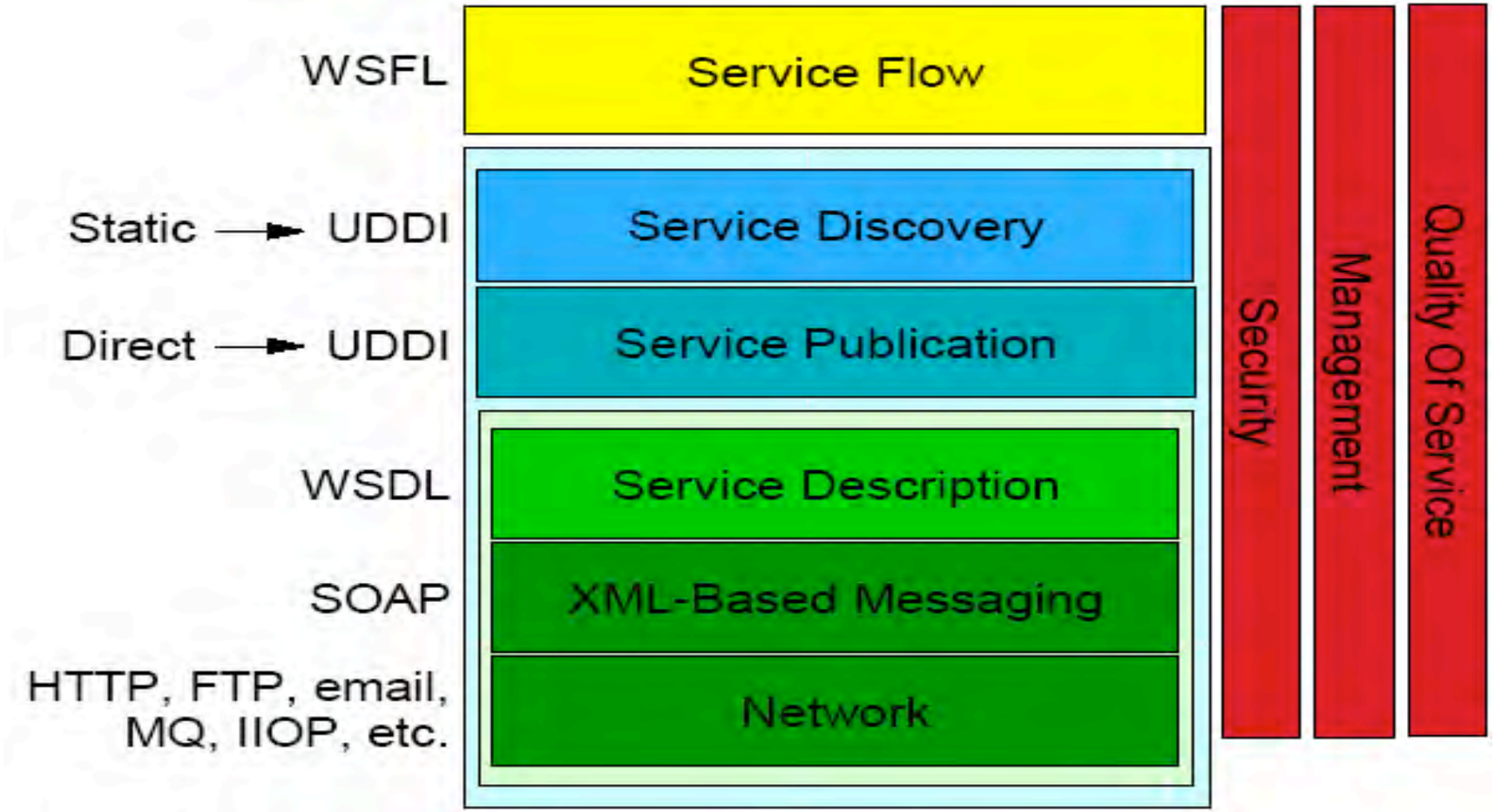
Vision

From local (tight coupled) heavy weight applications towards **Lightweight Applications** composed using **Distributed Services** on a **Global Scale**.

What is a Web Service?

- A **Web Service** is an **interface** that **describes** a collection of **operations** that are network **accessible** through **standardized XML messaging**.
- It is described using a standard, formal XML notion, called **service description.**
- The interface **hides the implementation details**, allowing it to be used **independently of platform and language**.
- This encourages **loosely coupled**, **component-oriented**, **location transparent**, **protocol independent**, **cross-technology** Web Service-based application implementations.

Conceptual Web Services Stack



SOAP (Simple Object Access Protocol)

- SOAP is **the** communications protocol for Web services.
- SOAP is XML based
- SOAP is a specification that defines the XML format for messages.
- The current SOAP standard assumes security is a transport issue and is silent on security issues.
- Beside document-centric messages SOAP also standardizes the enveloping of **Remote Procedure Calls (RPCs)** using XML.
- For RPCs the SOAP message contains a callable function with all input parameters.
- The server returns a SOAP message with the results of the executed function.
- No new transport protocol! Re-use of standard transport protocols: e.g., HTTP (POST), with an XML envelope as payload.
- HTTP binding is optional, almost all SOAP implementations support it. Other options: FTP, Email, ...

WSDL (Web Service Description Language)

- WSDL is an XML-based language that allows formal descriptions of the **interface of Web Services**:
  - which **interactions** does the service provide?
  - which **arguments and results** are involved in the interactions?
  - which network addresses are used to **locate** the service?
  - which communication **protocol** should be used?
  - which **data formats** are the messages represented in?

UDDI

Universal Discovery Description and Integration

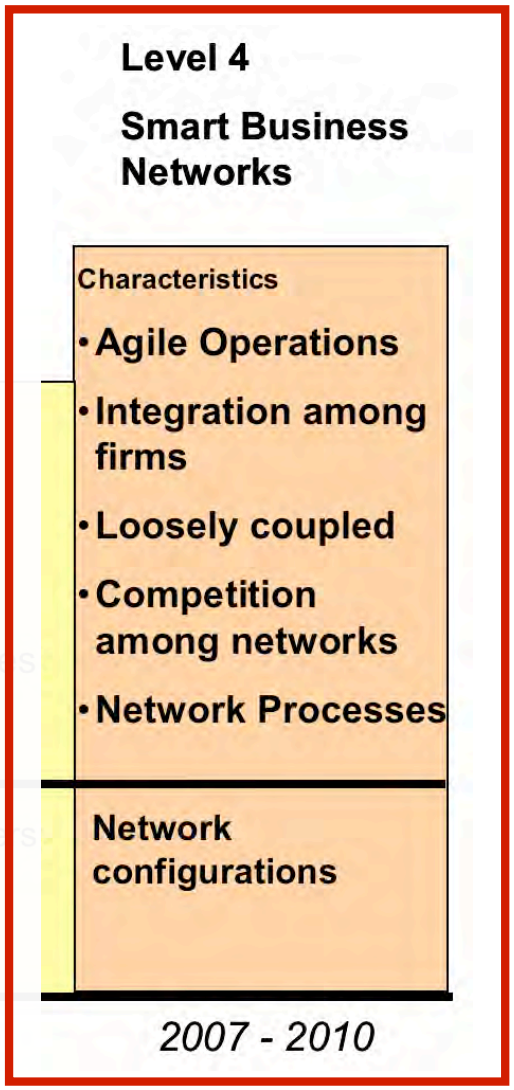
- UDDI is a **directory for Web Services** and allows to search for a company that offers the required service, read about the service offered and contact someone for more information
- A UDDI directory entry is an XML file that describes a business and the services it offers. There are three parts to an entry in the UDDI directory.
  - The **"white pages"** describe the company offering the service: name, address, phone, contacts, etc.
  - The **"yellow pages"** include industrial categories based on standard taxonomies such as the Standard Industrial Classification.
  - The **"green pages"** describe the interface to the service in enough detail for someone to write an application to use the Web service, normally by a link to the corresponding WSDL file
- Unfortunately, (almost) nobody uses UDDI (correctly)
  - Major service providers register their services without classification (e.g. Google, Amazon)
  - It is hard to automatise the selection of a service, e.g. to write a client program which uses UDDI to automatically select a “good” book store to buy a special book.

Open Standards

- Open vs. Proprietary Standards
- W3C: World Wide Web Consortium
- XML / Web Standards

Mission accomplished?

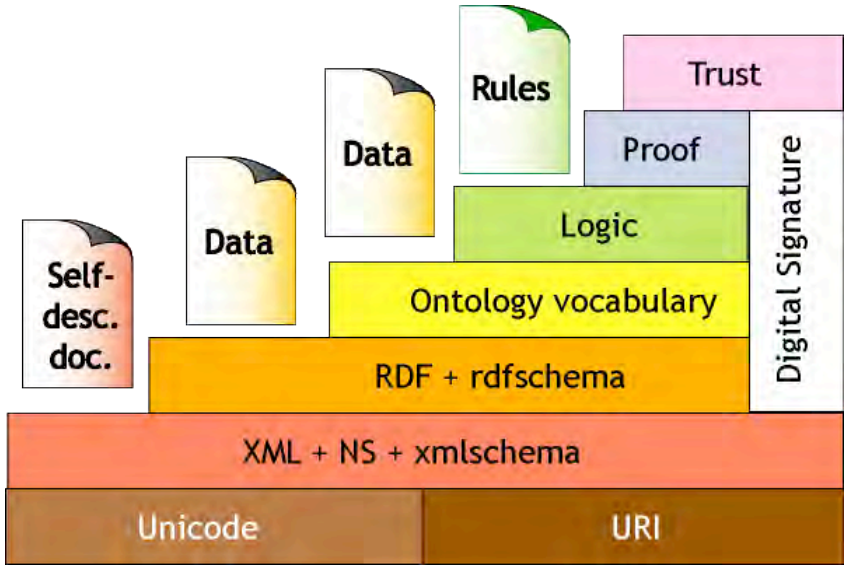
Semantically Enriched Web-Services ?



Semantic Web Layer-Model

- defined by W3C
- each layer uses the functionality of lower layer
- each layer provides extended functionality

- **Vision: Enrich Web-Service Technology with Semantic "Descriptions"**



http://www.w3.org/2001/09/06-ecdl/swlevels.gif

Example: „Intelligent“ Software Agents

- „Smart“ programs that perform delegated tasks from either human beings or other programs

- Work autonomously and proactively
- Receive tasks and preferences from humans or programs
- Seek information from different (web) resources
- Collect information according to specified preferences
- Communicate and collaborate with other agents
- Make individual choices
- Return information to humans or programs

- Sometimes seen as *„just another software development approach/paradigma“*

SOA?

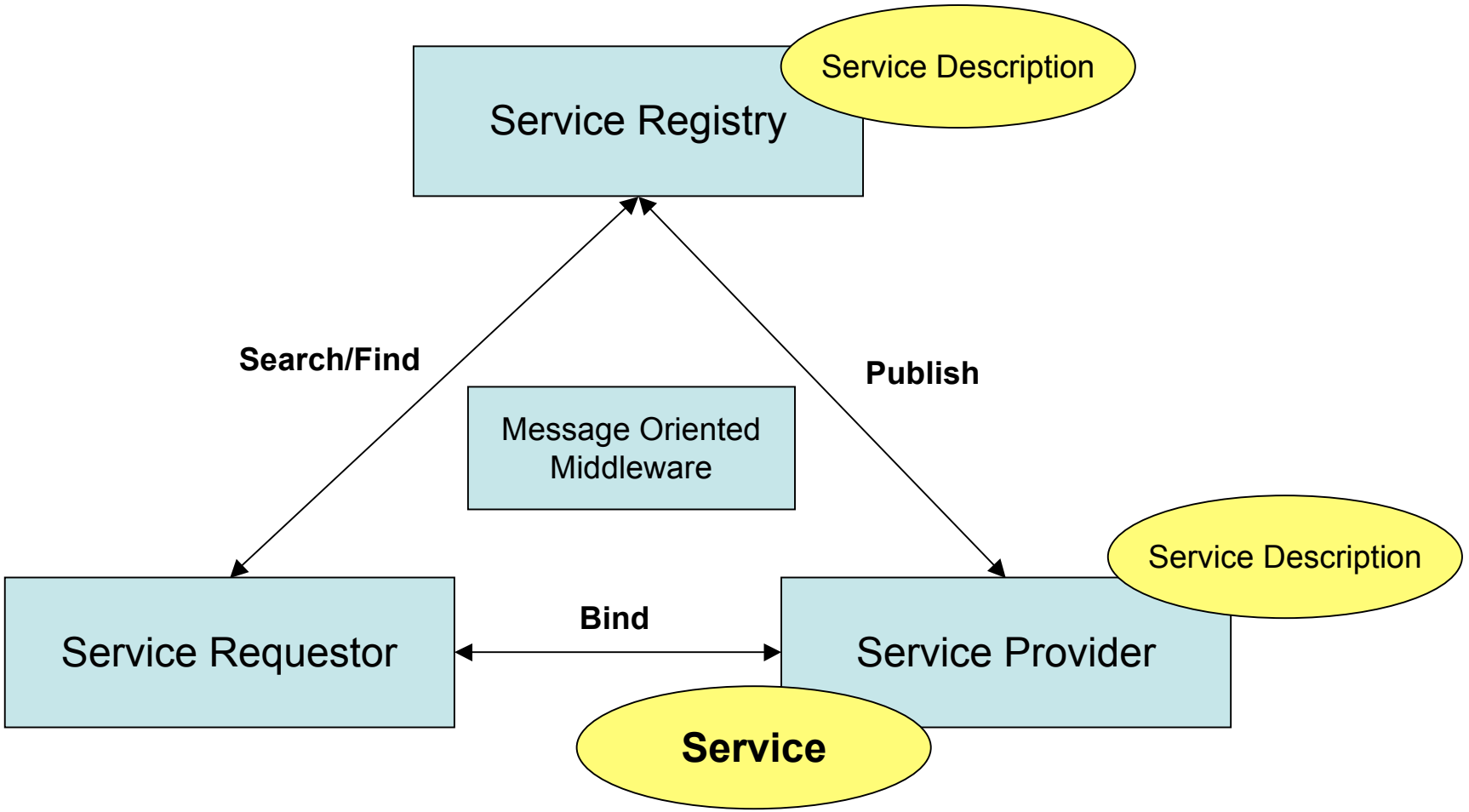
- Modularity
- Loose Coupling
- Separation of Concernce
- Composition
- Independence
- Interoperability

„Computer Science is entering a new generation. The previous generation was based on **abstraction from hardware**. The emerging generation comes from **abstracting from software** and sees all resources as **services** in a SOA“, Michael Brodie et.al.

A new Concept?

- First implementations in the 90th e.g. Sun Java/Jini
- New Momentum with Web-Service Architecture

Find-Bind-Invoke Paradigms



Loose Coupling

- Reuse of Functionality
- Reduce dependencies between systems
- Enhance Robustness
- Enhance Flexibility & Increase Interoperability through standard (XML) interfaces

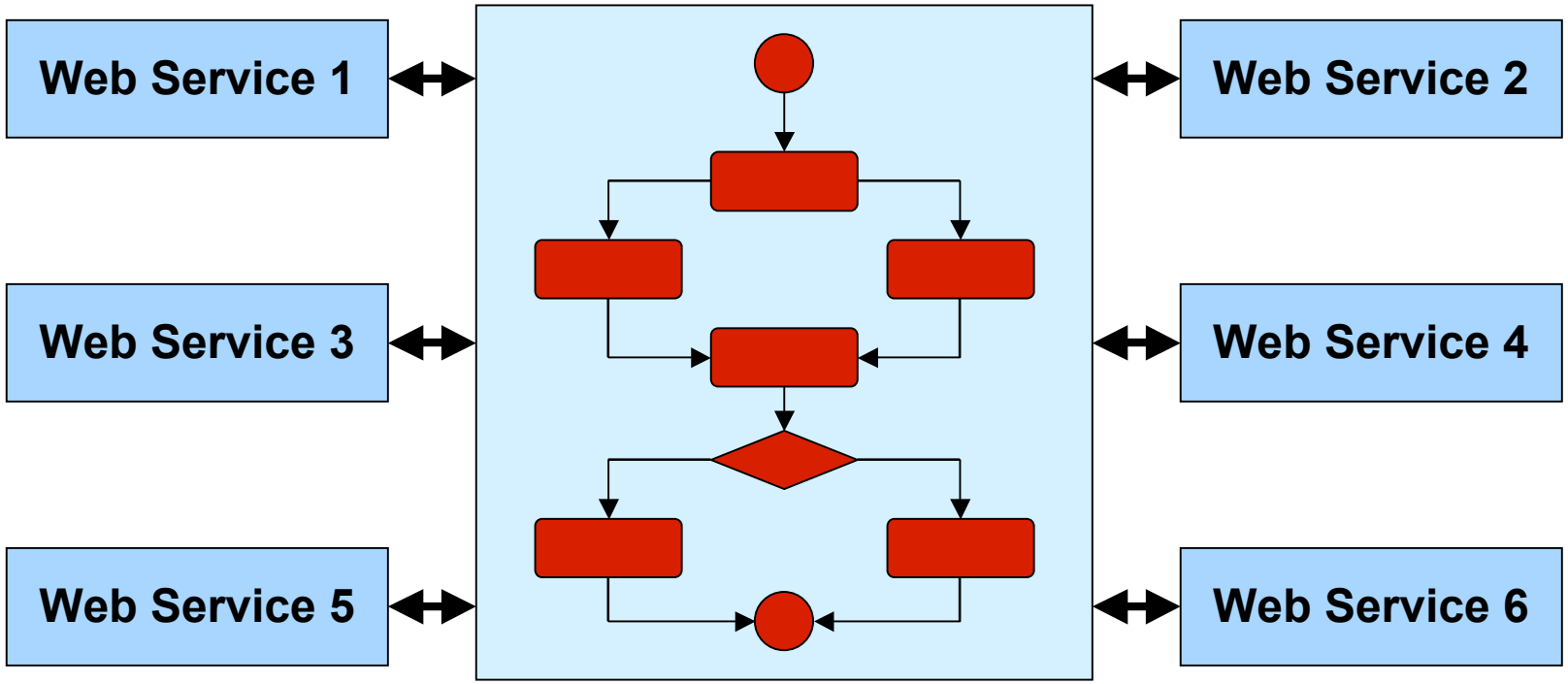
- Can be mediated by *Message Oriented Middleware*

Closing the Gap to... Business Frameworks

- Vision:
  - Self-Organising-
  - Self-Locating-
  - Self-Assembling-Services
- “Down to Earth”: Business Frameworks
  - Can business processes profit from SOA?
  - Workflow Systems Support?
  - Electronic Data Interchange? More then exchanging documents?

Motivation

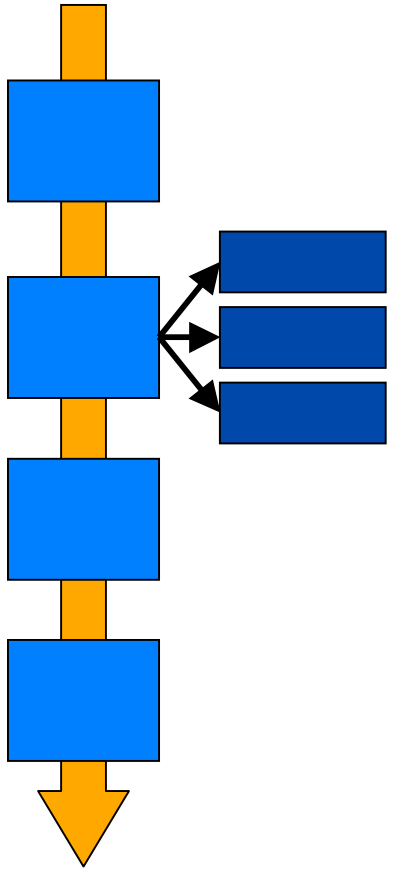
- Enterprises run many (Web-)Services
- Link (Web-)Services into Business-Workflows



Business Workflow (1)

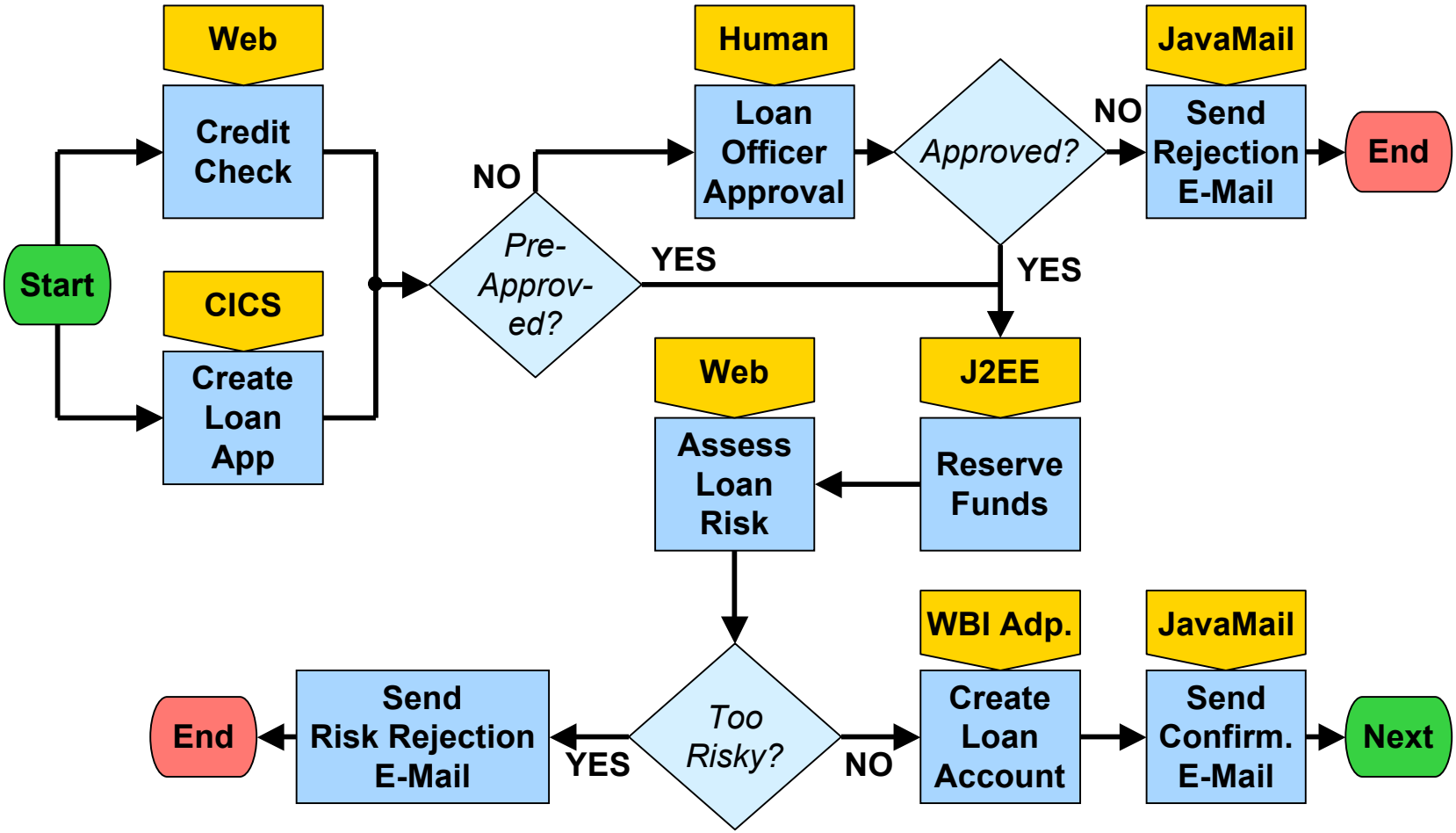
"A process means the treatment of objects. Thus, it consists of activities and objects"
[Nordsieck, 1934]

Workflow = *"Automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedure rules"*
[Interoperability of workflow products, http://wfmc.org]



Business Workflow (3)

A NEW and quick Loan Application System



<Type> ... <Type> of Service

Objectives / Business Process Challenges (1)

Design a Standard (Meta-Language / Framework) which is capable of:

- handle calls to (web-)services
- (re-)use existing (web-)services
- provide language constructs for workflow based software applications:
  looping, conditionals, variables, interfaces to the local system resources, ...
- do as less coding as possible

Objectives / Business Process Challenges (2)

- Coordinate asynchronous communication
- Correlate message exchange
- Implement parallel processing
- Manipulate/transform data (between interactions)
- Support for long running transactions and activities
- Provide consistent exception handling

Business Process Execution Language (BPEL)

Contributors (2002): IBM, BEA, Microsoft, SAP, Siebel

EDI

- History
  - 1975: Publication of Transactions
    - Transportation Data Coordination Committee (TDCC)
  - 1983: First Version of ANSI EDI
  - 1985: Development of UN/EDIFACT (EDI For Administration, Commerce and Transport)
  - Recently: from Data Exchange to Process interaction
- Traditional EDI Standards
  - UN/EDIFACT (worldwide, no specific branch)
  - ANSI X12 (USA, no specific branch)
  - SWIFT (worldwide, Bank)
  - ODETTE (Europe, Automotive)
  - AIAG (USA, Automotive)
  - TDCC (USA, Transport)
  - VDA (Germany, Automotive)
  - etc.

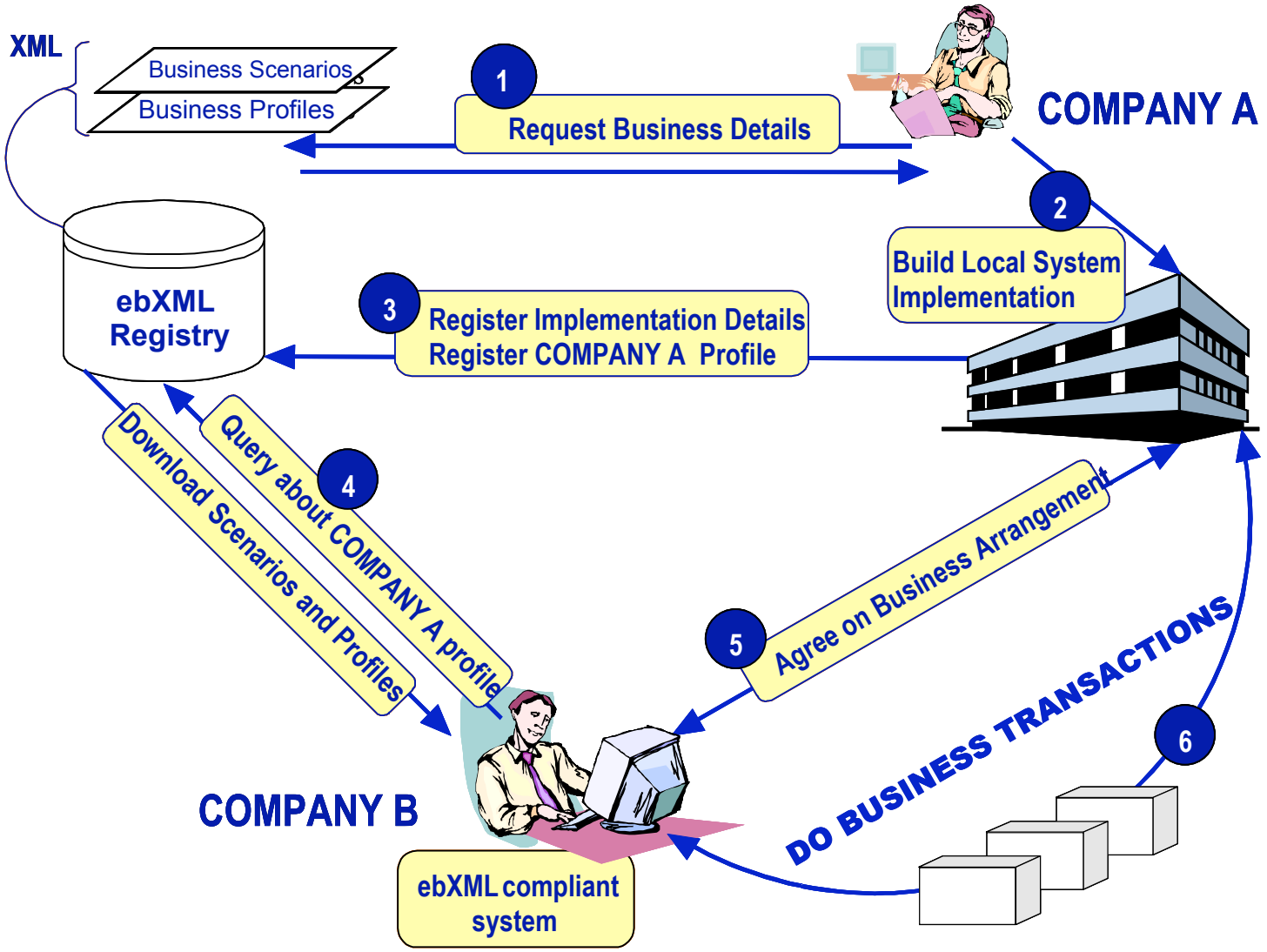
Problems of traditional EDI Standards

- Many different standards, depending on business context
- Complicated Syntax; difficult to understand and implement
- Installation, Setup and Maintenance is expensive
  - Usage base mostly „big“ enterprises
  - SMEs can hardly effort current EDI systems
- Traditionally EDI Systems use„Value Added Networks“ (VAN)
- Standard is highly optional; difficult to implement specific application

XML/Services-based vs. traditional EDI

- XML Based EDI (Open EDI, XML/EDI, xCBL, cXML, BizTalk, RosettaNet...)
- XML usage has advantages
  - Syntax easier to understand and implement
  - Better integration with current technologies
  - Cheaper
  - Easier to implement in existing applications
- But many problems persist:
  - Semantic description not complete
  - Similar „overloaded“ documents are possible
  - Considering the usage scenarios, different dialects of the standards will evolve
- Service Based EDI
  - → **Goal is the Integration of different Standards**
  - → **Business-Process-oriented instead of Data-centric approach** is favourable

Example: ebXML Scenario



Contact

Dr. Alexander Schatten

Vienna University of Technology
Institute for Software Technology
and Interactive Systems

Vienna, Austria, Europe :-)

http://www.schatten.info