4. Concepts and Technologies for B2C, B2E, and B2B Transaction

4.4 Exchanging Information within Open Business Communities

4.4.1 “Pre-Internet” B2B standards: EDI, Interactive EDI, Universal EDI, OpenEDI

4.5 Technologies for Internet-based B2C or B2B Commerce

4.5.1 XML, SOAP, UDDI

4.5.2 ebXML, BizTalk, ICE

4.5.3 WebServices

Electronic Data Interchange (EDI) (1)

The **Electronic Data Interchange (EDI)** standard existed before the Internet existed and was used to conduct business. It allows for exchange of electronic business documents. It standardizes electronic messages and the content semantics:

- EDI itself exists as two disparate *de-jure standards*: EDIFACT (Europe) and X12 (USA).
- *De-facto standards* were introduced by „big players“ ⇒ many different formats exist.

EDI messages were primarily specified for usage on Telex machines ⇒ their format is severely restricted, they may not contain binary data

Subsequent standards of EDI:

- Non-internet-based EDI Successors:
  - Open EDI, Universal EDI (OO-EDI), Interactive EDI
- Internet-based EDI Successors:
  - ICE, ebXML, WebServices.

Example

EDI (EDIFACT order)

```
UNH+0002771776+ORDERS:D:99A:UN:FI0084'
BGM+105+4006300027717760274849 7+9'
DTM+4:20000705:102'
DTM+2:20000706:102'
DTM+9:20000705:102'
NAD+BY+003709895955:100++TRADEKA OY'
NAD+SE+003702134547:100++OY HARTWALL AB'
NAD+CN+40063000::92++VALINTATALO
HERVANTA+LINDFORSINKATU 2+TAMPERE++33720'
LIN+1++6413600001584:EN'
IMD+F+8+:::HTW NOVELLE ORANGE LIME 1,5 L'
QTY+21:144'
LIN+2++6413600000280:EN'
IMD+F+8+-:::VICHY 0,33L/HARTWALL'
QTY+21:430'
UNS+S'
CNT+2:2'
UNT+17+0002771776'
```

Source: TIEKE (EDI standards implementation guidelines document)

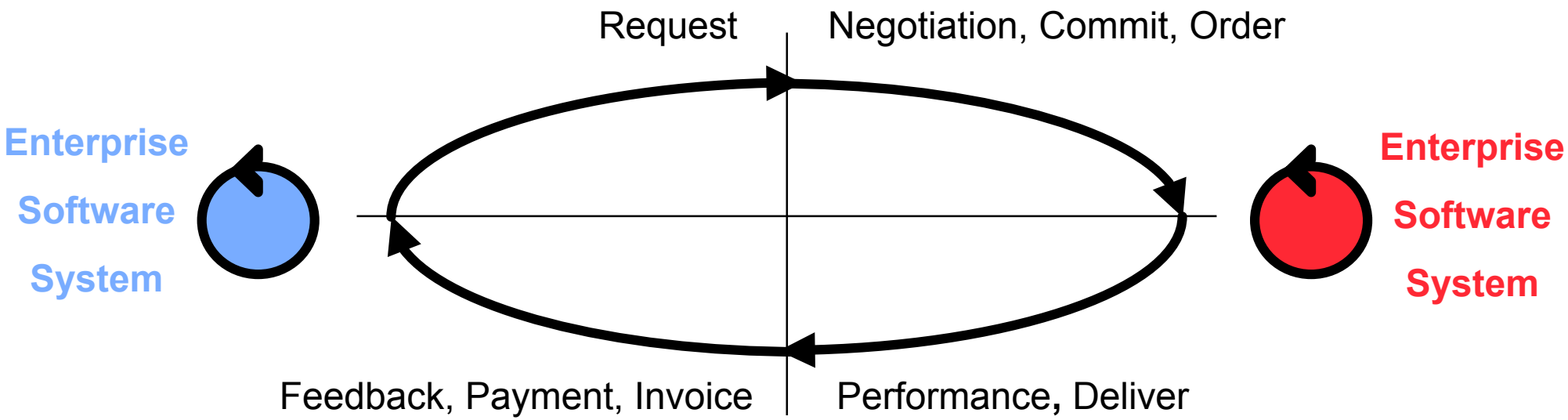
(c) Paavo Kotinurmi 2001-2004

24.11.2004

EDI explained

- EDI is a set of message formats that act as templates for purchase orders, invoices, payments, shipping manifests, and the like.
- They allow data to be moved around between companies while ensuring that it can be automatically read by all.
- EDI has nothing to say about how individual applications are built, only that they should export data in a particular format, and that they can expect to receive data in a particular format.

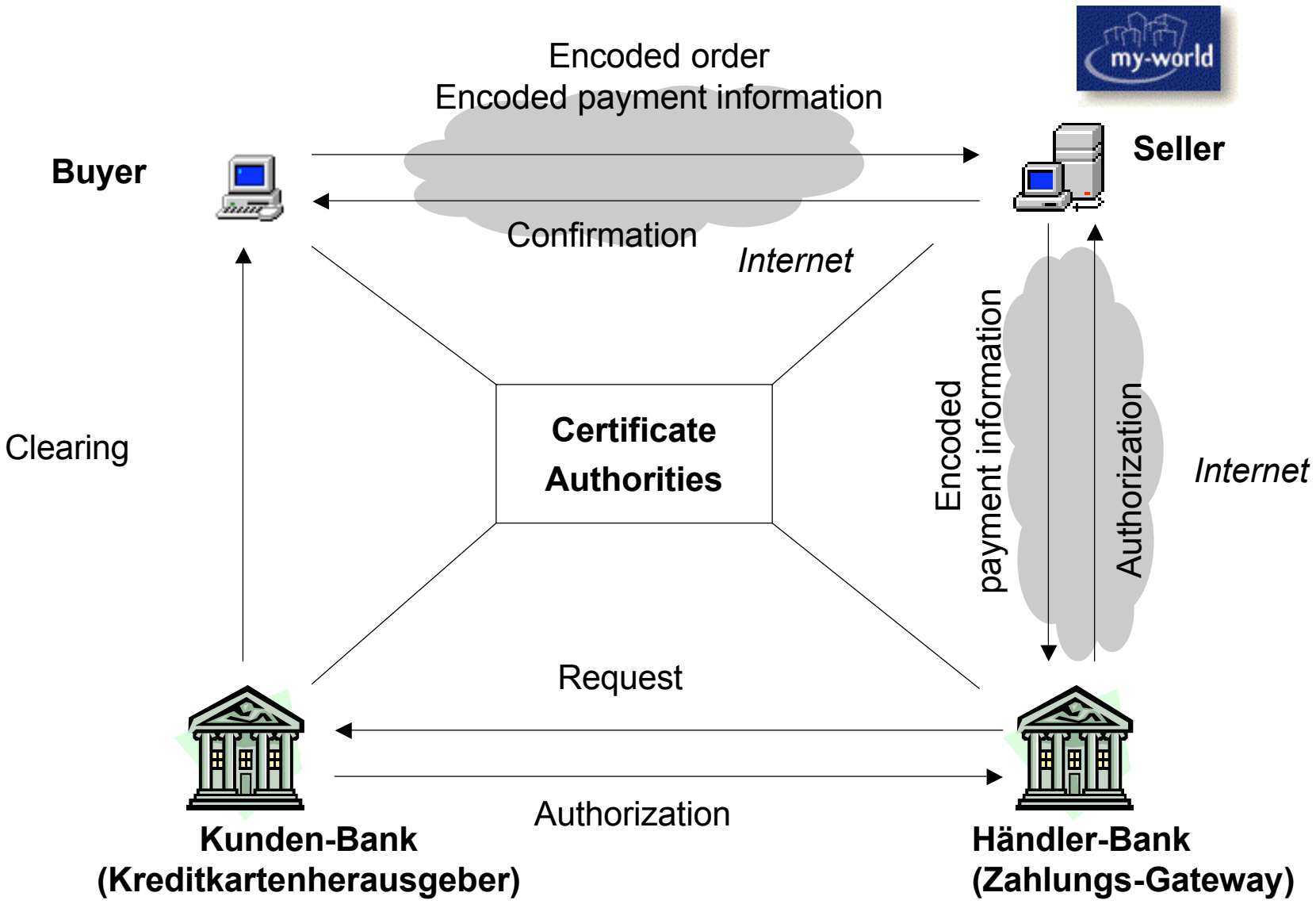
Example Scenario 1



- Order
- Deliver
- Invoice
- Payment

handled by EDI

Example Scenario 2



4. Concepts and Technologies for B2C, B2E, and B2B Transaction

4.4 Exchanging Information within Open Business Communities

4.4.1 “Pre-Internet” B2B standards: EDI, Interactive EDI, Universal EDI, OpenEDI

4.5 Technologies for Internet-based B2C or B2B Commerce

4.5.1 XML, SOAP, UDDI

4.5.2 ebXML, BizTalk, ICE

4.5.3 WebServices

RDM Revisited: Project Database

Projects

Nr	*Title*	*Budget*
100	DB Schedules	300.000
200	ADAC Customers	100.000
300	Telecom Statistics	200.000

Project execution

Nr	*Abbr*
100	MFSW
100	UXSW
100	LTSW
200	UXSW
200	PERS
300	MFSW

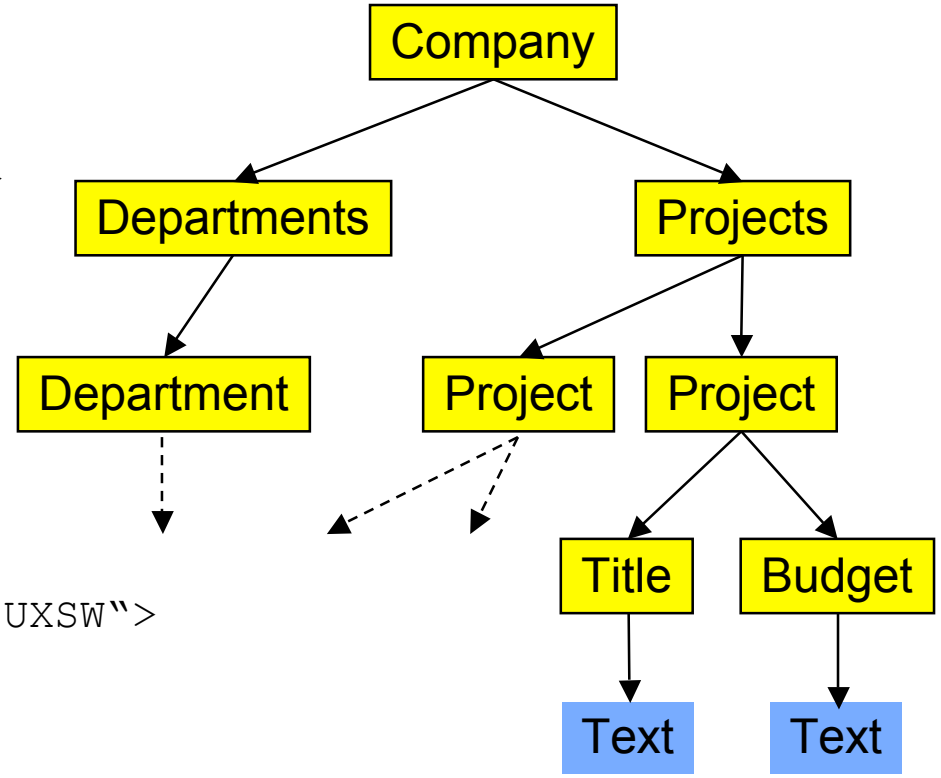
Departments

Abbr	*Name*	*MajorDp*
MFSW	Mainframe SW	LTSW
UXSW	Unix SW	LTSW
PCSW	PC SW	LTSW
LTSW	Leitung SW	**NULL**
PERS	Personal	**NULL**

Company‘s Project Database

XML Data Modeling: XML Document Structure

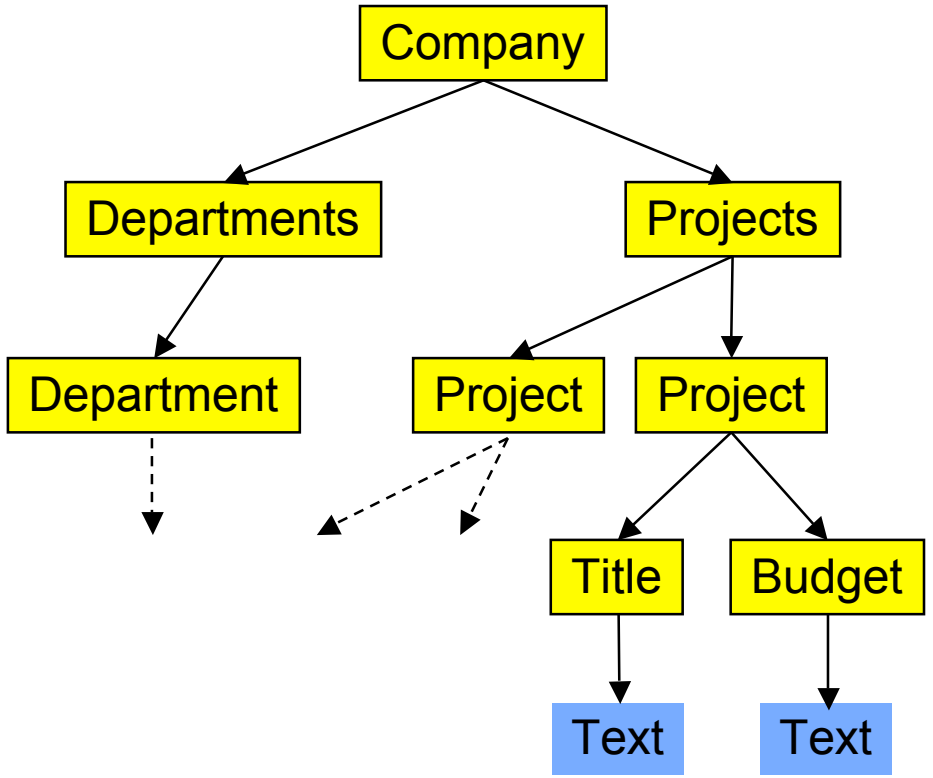
```
<Company>
  <Departments>
    <Department id=„MFSW“ projects=„P100“>
      <Name>Mainframe Software<Name/>
    </Department>
    ...
  </Departments>
  <Projects>
    <Project id=“P100“ departments=„MFSW, UXSW“>
      <Title>DB Schedules</Titel>
      <Budget>300000</Budget>
    </Project>
    ...
  </Projects>
</Company>
```



XML Data Modeling: Type Definition

Type definitions in XML:

- Definition of valid elements and attributes
- Definition of valid structures (sequences, repetitions, optional elements, combinations)



1. DTDs (Document Type Definition)

- have a non-XML structure

2. XML Schemas

- replacement for DTDs, defined in XML themselves

Document Type Definition (DTD)

```
<!ELEMENT Company (Departments, Projects?)>

<!ELEMENT Departments (Department*)>
<!ELEMENT Department (Name, Subdivisions?)>
<!ATTLIST Department
          id ID #REQUIRED
          projects IDREFS #IMPLIED>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Subdivisions (Department+)>

<!ELEMENT Projects (Project*)>
<!ELEMENT Project (Title, Budget)>
<!ATTLIST Project
          id ID #REQUIRED
          departments IDREFS #IMPLIED>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Budget (#PCDATA)>
```

XML Schemata

DTDs have some disadvantages:

- DTDs are not XML themselves. Different syntaxes, tools are needed for authoring XML and XML DTDs.
- DTDs have restricted expressiveness, e.g. it is not possible to define basic data types other than String (*Integer*, *Date*, *Time*, *boolean,* ...)
- DTDs do not support inheritance of elements, e.g. Person _ Student)

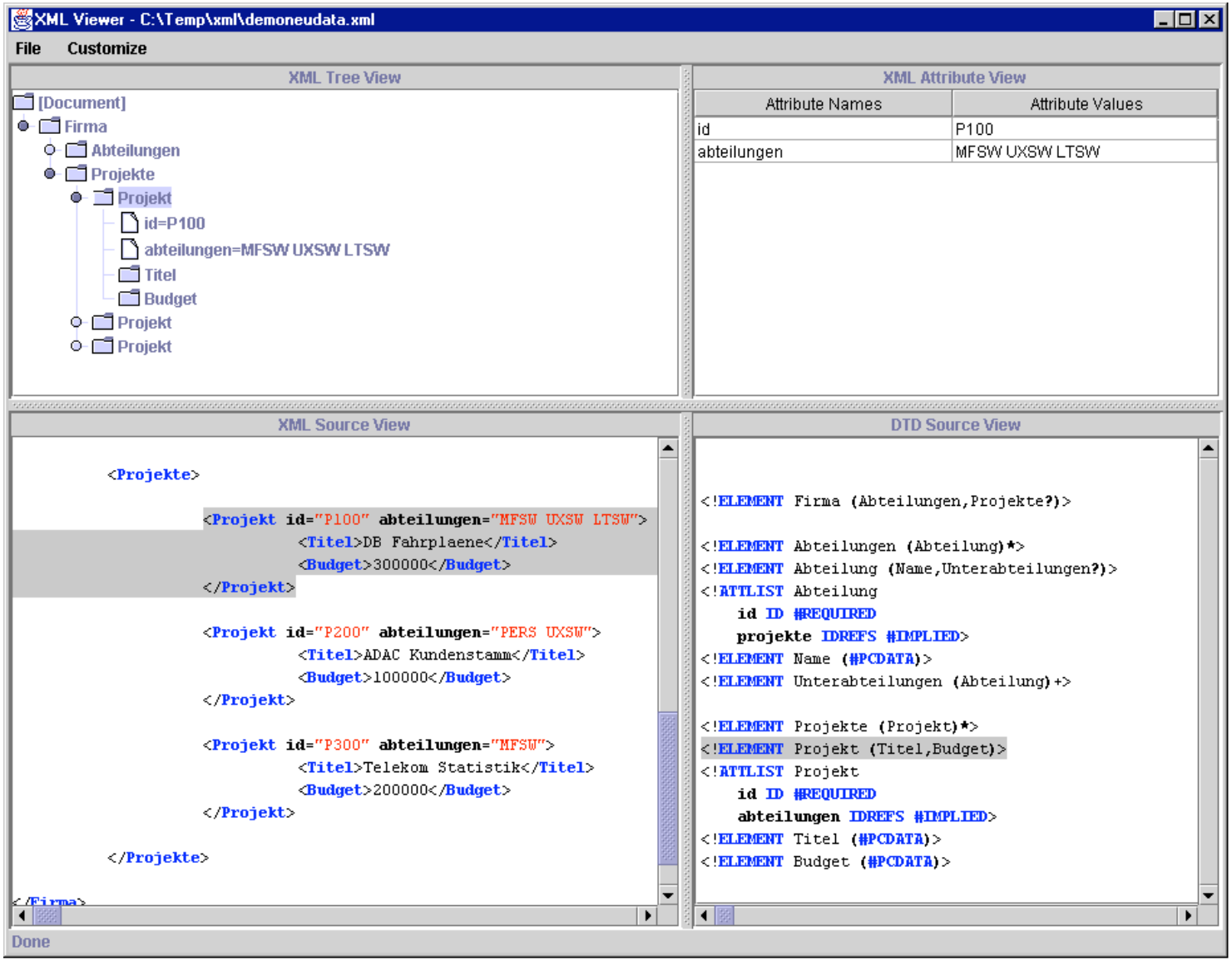
Solution: XML Schemas

- provide a superset of modelling capabilities of DTD.
- are XML documents: Schemas can be authored, managed, transformed with the same tools as XML documents.

  Examples: Product type definition, business process definitions.
- define base datatypes, e.g. *int*, *boolean*, *DateTime*.
- allow domain-specific datatypes to be specialized from existing datatypes, e.g., bank account number, immatriculation number.
- allow import of datatypes from other schemas

XML Schemas are much better suited for modeling business data, process data.

Visualization of XML Data: Editor



Visualization of XML Data: XSL (1)

NOTE: XML does only specify content, not formatting / visualization aspects.
For presentation, XML must be converted into a visualization format. **Style languages** are used (like CSS).

The **XSL (eXtensible Stylesheet Language)** is a style language that is used to transform XML documents into another format (usually, a visualization format, e.g., HTML).

An XSL document defines the formatting of a class of XML documents.

XSL comprises two distinct parts:

- Transformation (XSLT)
- Formatting

Only the transformation part is described more detailed here (see www.w3.org/xsl) for further information).

XSL (2)

Transformation rules (template rules)

Transformations rules comprise two parts: a *pattern* and an *action*.

```
<xsl:template match="Project">
<B>
      <xsl:apply-templates/>
</B>
</xsl:template>
```

Pattern

Action

The pattern specifies the XML elements to which the action applies. More than one rule may be applicable to one element. In that case, the most specific rule is chosen by the XSLT processor.

This way, a standard transformation can be defined that can be overridden for specific elements.

XSL (3)

```
<Project id="P100"
         departments="MFSW UXSW LTSW">
   <Title>DB Schedules</Title>
   <Budget>300000</Budget>
</Project>
...
```

XML

```
<TR>
   <TD><B>DB Schedules</B></TD>
   <TD>300000</TD>
   <TD><I>MFSW UXSW LTSW</I></TD>
</TR>
...
```

HTML

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://...">

<xsl:template match="Project">
 <tr>
  <td><B>
      <xsl:value-of select="Title"/>
  </B></td>
  <td><xsl:value-of select="Budget"/></td>
  <td><I>
      <xsl:value-of select="@departments"/>
  </I></td>
  <xsl:apply-templates/>
 </tr>
</xsl:template>

</xsl:stylesheet>
```

XSL Stylesheet

DB Schedules	300000	*MFSW UXSW LTSW*
ADAC Customers	100000	*PERS UXSW*
Telecom Statistics	200000	*MFSW*

visualized as a table

Simple Object Access Protocol (SOAP)

The **Simple Object Access Protocol** (SOAP) is an XML / HTTP based protocol for **accessing services, objects and servers in a platform-independent manner**.

- developed from the idea to create an XML-based RPC (remote procedure call) mechanism
- mainly driven by Microsoft and IBM ➔ large support base
- platform- and programming language independent
- asynchronous communication is possible
- standard transport protocol is HTTP.

SOAP is NOT a replacement for CORBA / DCOM, it is simply a wrapper technology to make services more accessible over the Internet

Spec: http://www.w3.org/TR/SOAP/

SOAP: Related, Existing Technologies

Existing distribution architectures and technologies include

- OMG CORBA, Microsoft DCOM, Java RMI, Unix-RPC, ...

But no single technology is used for dynamic and interorganizational service integration.

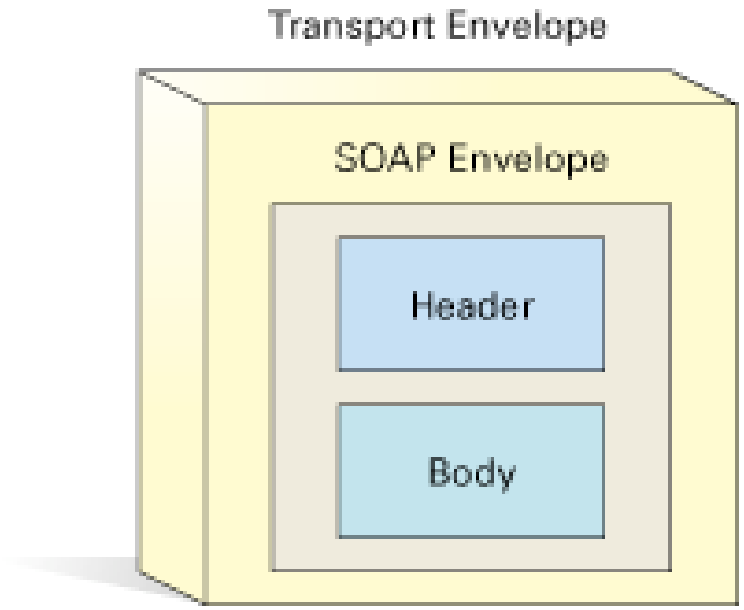
Some reasons:

- Unix-RPC: requires binary-compatible Unix systems on both sides
- CORBA: not widely adopted
- RMI: requires Java on both sides
- DCOM: requires Windows systems on both sides
- Firewall problems (packets are filtered by firewalls if using dynamic ports)

SOAP Message Structure

SOAP extends HTTP request / responses with

- SOAP-Header to identify that it's a SOAP message
- SOAP-Body containing an XML *payload*: the actual data to be transmitted



SOAP: Example Message: Stockquote

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope
     xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
      <m:GetLastTradePrice xmlns:m="Service definition (namespace) URL">
         <symbol>SAP</symbol>
      </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Lookup Services: UDDI

The **Universal Description, Discovery and Integration** (UDDI) project aims at creating a platform-independent, open framework for

- describing services
- discovering businesses and
- integrating business services using the Internet,

as well as an

- operational registry (i.e., naming service) [UDDI02].

UDDI uses SOAP as protocol.

A **UDDI node** is a global directory service for companies and web services that companies provide.

Currently, there exist some public UDDI nodes at IBM and Microsoft, HP, etc.

See: http://www.uddi.org

UDDI (2)

Content types of UDDI nodes:

- **White Pages:** General information about companies:

  name, address, contact, etc.

- **Yellow Pages:** Classification of companies

  Based on existing, non-electronic standards (example: North American Industry Classification System, NAICS)

- **Green Pages:** Technical information about WebServices provided by a company.

  WebServices will be explained in 6.3.4.

4. Concepts and Technologies for B2C, B2E, and B2B Transaction

4.4 Exchanging Information within Open Business Communities

4.4.1 “Pre-Internet” B2B standards: EDI, Interactive EDI, Universal EDI, OpenEDI

4.5 Technologies for Internet-based B2C or B2B Commerce

4.5.1 XML, SOAP, UDDI

4.5.2 ebXML, BizTalk, ICE

4.5.3 WebServices

XML / EDI ⇒ ebXML

Original purpose: Realize EDI using XML documents as EDI messages.

Motivation: XML for data and document modelling:

- Portable exchange format for data of different application domains.
- Storage of semi-structured data in (XML-)databases. Mixes describing data (meta data) and data ítself.

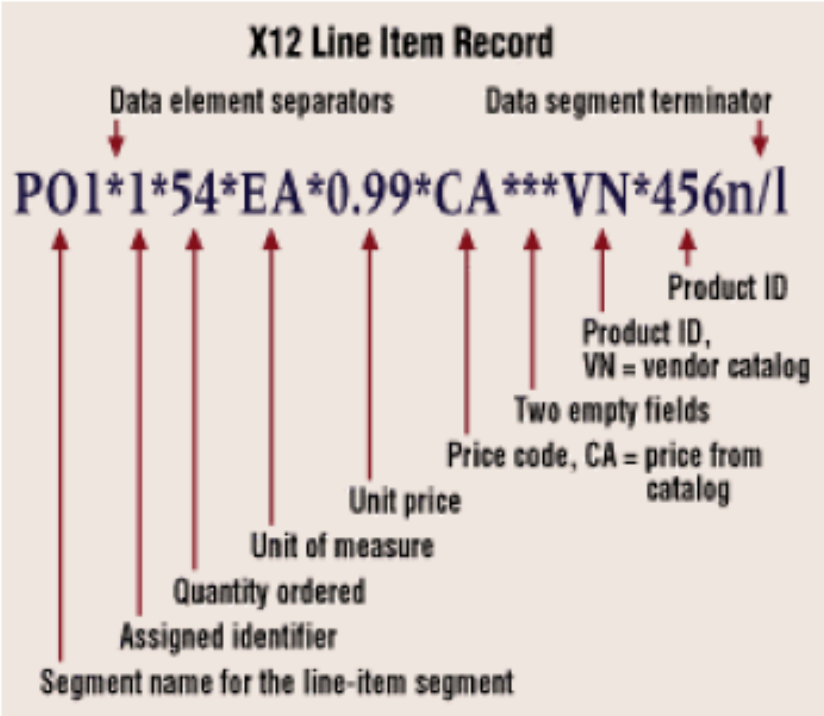
XML / EDI has surfaced as the **ebXML** standard. ebXML has only few connections to the original EDI standard:

- Open standard platform (unlike EDI)
- Business service modeling is done in UML (design) and XML (representation)

Example

XML vs. EDI example

```
<po1>
 <ass.Id>1</ass.id>
 <Q.Ordered>54</Q.Ordered>
 <Unit.M>EA</Unit.M>
 <Unit.P>0.99</Unit.P>
 <Price.C>CA</Price.C>
 <Prod.ID.C>VN</Prod.ID.C>
 <Prod.ID>456</Prod.ID>
</po1>
```



Source: Drummond and Spearman, 1999.

(c) Paavo Kotinurmi 2001-2004

ebXML (1)

Purpose of **ebXML (Electronic Business eXtensible Markup Language)** [ebXML00]:

- The **United Nations body for Trade Facilitation and Electronic Business (UN/CEFACT)** and the **Organization for the Advancement of Structured Information Standards (OASIS)** have initiated a project to standardize XML business specifications.
- UN/CEFACT and OASIS have established the Electronic Business XML Initiative to develop a **technical framework** that will enable XML to be utilized in a consistent manner for the exchange of **all electronic business data**.
  - ebXML is an open architecture, not a standard
- **Lower the barrier of entry** to electronic business in order to **facilitate trade**, particularly with respect to **small- and medium-sized enterprises (SMEs)** and **developing nations**.

Technically:

- Provide an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure, and consistent manner by all parties.

[ebXML00]

ebXML (2)

ebXML values [ebXML00]:

- Provides a globally developed open XML-based standard built on a rich heritage of electronic business experience.
- Creates a Single Global Electronic Market
- Enables all parties irrespective of size to engage in Internet-based electronic business.
- Provides for plug and play shrink-wrapped solutions.
- Enables parties to complement and extend current ECommerce / EDI investment and expand electronic business to new and existing trading partners.
- Facilitates convergence of current and emerging XML efforts.

Note:

- Proof-of-concept has been successfully done.
- Widely regarded as a strong successor-elect of EDI.

ebXML (3)

Applications of ebXML:

- Exchange electronic business documents:
  - Dynamically formulate trading partnerships through a registry and repository (REG/REP)
  - Negotiate and implement collaborative partner agreements (CPAs)
  - Exchange electronic business transactions using a consistent XML-based messaging infrastructure (transport, routing, and packaging)
- Integration of businesses‘ back-end systems:
  - Public and private trading networks are the focus of many supply-chain initiatives. These networks must provide integration between the back-end systems of their participants to truly automate the exchange of the documents underlying routine commercial transactions.

[TW00, Sun00]

BizTalk: Goals

BizTalk is Microsoft's platform for

- business partner integration as well as
- enterprise applications integration (EAI) (not covered here).

Goals for integrating business partners by

- Designing and implementing dynamic business processes (B2B – electronic document exchange processes)
- Integrating legacy systems and enterprise applications into business processes (Enterprise Application Integration, EAI).
- Open Standards: BizTalk relies on open standards and specifications (SOAP / XML, HTTP, SMTP).

Purpose: Relate interorganizational messaging technologies (messaging systems, emailing, XML messaging) with a company's applications and legacy systems.

[MS01]

BizTalk: Technical Aspects

Technical aspects:

- BizTalk server tracks data and documents exchanged in business processes.
- Document type definition (schema) transformation support for mapping business documents of different companies.
- BizTalk provides a programming framework (2.0) for (legacy) system integration.

Document exchange formats:

- EDI (both standard ANSI X12 and EDIFACT)
- HTTP / HTTPS (web-based)
- SMTP (mail-based)
- File-transfer
- Fax – only outbound

Application / Legacy System Integration:

- Integration via *Open Binding Architecture*: Adapters connect BizTalk System to applications & legacy systems.

[MS01]

Information and Content Exchange (ICE) Format

The **Information and Content Exchange (ICE)** standard allows exchange of (business) content (e.g., business data, contracts, product information, copyright information) between business partners (see http://www.w3.org/TR/NOTE-ice.html). ICE uses XML as data format and HTTP as transfer protocol.

ICE is a

- two-partner (subscriber / syndicator)
- request / response-based protocol for
- inter-server (Business-to-business) data exchange format.

It defines common

- Operations (establishing data subscriptions, data supply, etc)
- Data elements (catalogs. etc),

that can be extended for markets. Technically, the elements of external DTDs can be included in ICE documents.

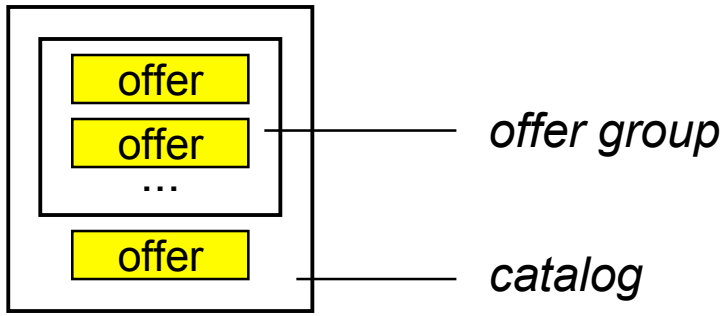
[Merz99]

ICE (2): Operations, Subscription Catalogs

Operations:

- **Establishing data subscriptions**: Subscriber and syndicator negotiate terms of subscription: start and end date, date and frequency of data supply, supply method (push/pull). ICE furthermore allows for negotiation of copyright information and priority of data supply.
- **Data supply**: Content is exchanged packet-wise. ICE supports two modes: full data supply and incremental data supply. Content can be (as usual with XML) included in the data stream or referenced using a URI.

For establishing data subscriptions, the syndicator sends offers as a **ICE catalog** to the subscriber. The catalogs must conform to an ICE catalog DTD. A catalog contains *offers* and *offer-groups*.



[Merz99]

RosettaNet.org

RosettaNet

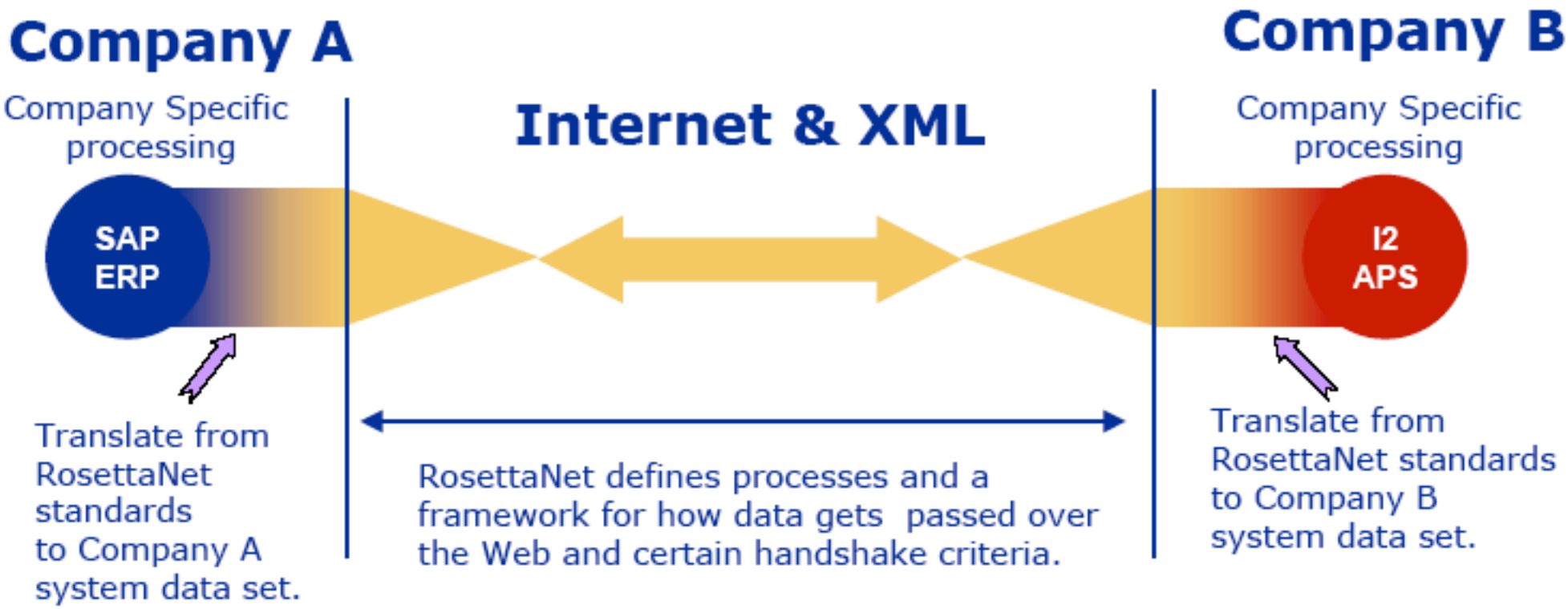
- RosettaNet is a non-profit organization that directs business process and supply chain standardization. It is especially strong in the electronics industry even though it is not limiting itself to any particular industry.
- Over 500 members (HP, Cisco, Intel, Microsoft, Nokia, Sun…)
  - Electronic Components (EC),
  - Information Technology (IT),
  - Semiconductor Manufacturing (SM),
  - Solution Provider (SP)
  - Telecommunication (TC) (Quite recent)
  - Logistics (LG) (Quite recent)

Purpuse

RosettaNet

- Standardizes inter-company "public" processes (PIPs)
- The related messages (DTD + Message guidelines) = Business documents
- Standard messaging framework (RNIF).
- Defines dictionaries (RNTD and RNBD) and codes (GTIN and DUNS)
- Trading partner agreement (TPA)
- RN technical dictionary only really industry specific part

Communication based on Internet and XML

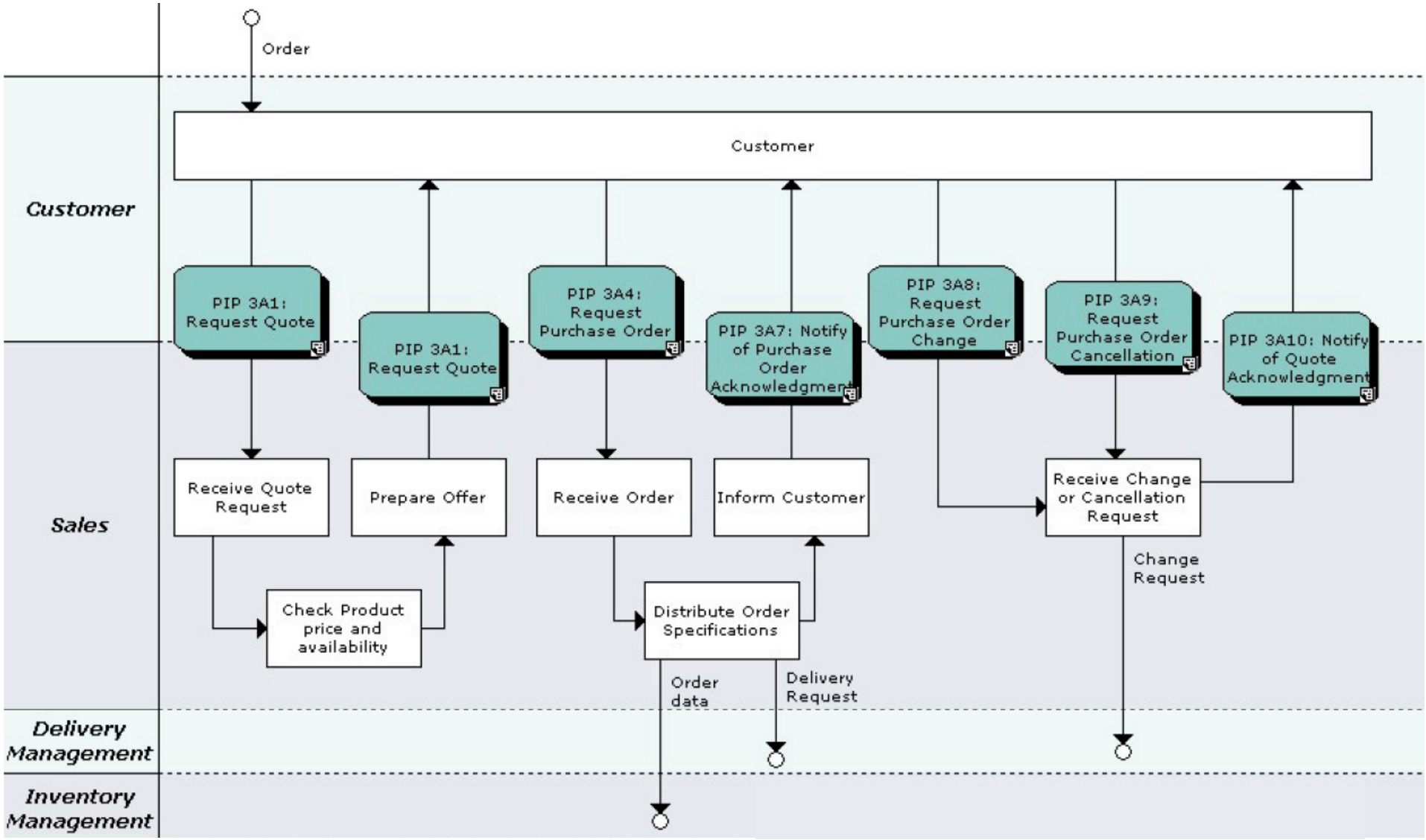


source: RosettaNet

HELSINKI UNIVERSITY OF TECHNOLOGY

(c) Paavo Kotinurmi 2001-2004 24.11.2004

Order Processing in Generic Company



4. Concepts and Technologies for B2C, B2E, and B2B Transaction

4.4 Exchanging Information within Open Business Communities

4.4.1 “Pre-Internet” B2B standards: EDI, Interactive EDI, Universal EDI, OpenEDI

4.5 Technologies for Internet-based B2B Commerce

4.5.1 XML, SOAP, UDDI

4.5.2 ebXML, BizTalk, ICE

4.5.3 WebServices

4.5.3 WebServices

WebServices Overview

WebServices Technologies

Combination of WebService and HTML Frontend

Extending WebServices to Business WebServices

WebService: Definitions

WebServices are ...

... self-contained, modular applications that can be described, published, located, and invoked over a network, generally, the Web. [IBM]

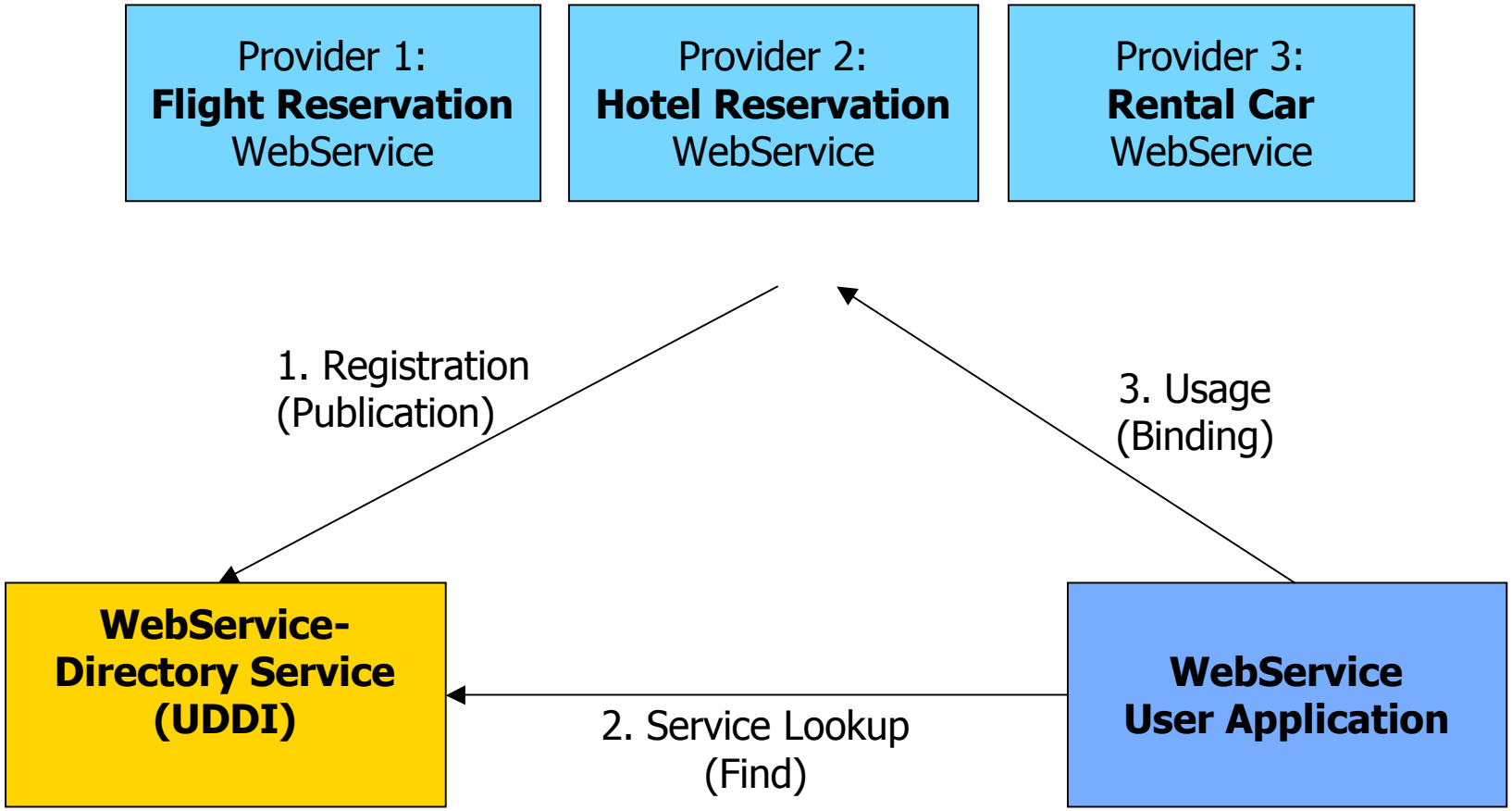
... a type of service that can be shared by and used as components of distributed Web-based applications [BEA].

... programmable application logic accessible using standard Internet protocols [Microsoft].

WebServices are **online services** (e.g. flight reservation service, hotel booking service, etc.) that are accessed over the Internet via **method invocation** and not by requesting HTML pages. WebServices can technically be understood as remote procedure calls (RPC). They use SOAP messages for invocation, parameter passing and returning results. As a result, they provide information but no visualization.

WebServices can be used by other WebServices or by human users. As WebServices do not provide presentation of content, the content must be rendered (visualized) for the human user.

WebService: Example



Benefits and Shortcomings of WebServices

Benefits of WebServices

- Uses open and widely adopted standards
- Platform- and language-independent
- Broad support by major players (IBM, Microsoft, Sun, …) and agreement about new, upcoming standards
- "WebServices are a necessity":

  Dynamic B2B integration requires service-to-service communication.

  Business Application Integration (BAI) will be simplified

Shortcomings:

- General problem: How to describe the *semantics* of services (WebServices do not solve this problem either)
- Presentation of service must be provided separately.

4.5.3 WebServices

WebServices Overview

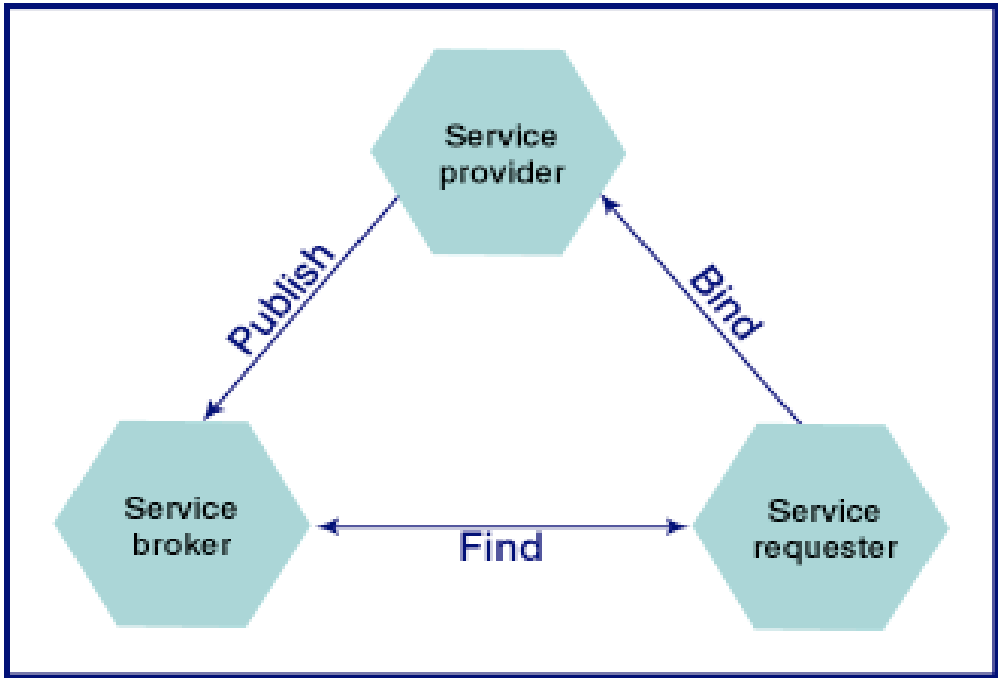
WebServices Technologies

Combination of WebService and HTML Frontend

Extending WebServices to Business WebServices

WebServices Technologies

1. Description ➔ WSDL (XML-based Web Services Description Language)
2. Publishing ➔ UDDI (see prev. sections)
3. Lookup ➔ UDDI (- “” -)
4. Invocation ➔ SOAP (- ”” -)



Quelle: Dan Gisolfi (IBM) – Web Services Architect Part 1

WSDL

WSDL: WebServices Description Language

WSDL evolved from

- ❏ IBM‘s Network Accessible Service Specification Language (NASSL) and
- ❏ Microsoft‘s Service Description Language (SDL).

It is an XML-based description of service operations and message formats for these operations.
It contains

- ❏ Supported operations,
- ❏ Datatype definitions (parameters and result types),
- ❏ Message definitions,
- ❏ Protocol and data format descriptions.

Protocol bindings for SOAP, HTTP GET and POST and MIME types are defined in the standard.

Spec: http://www.w3.org/TR/wsdl

WSDL Example

```
<message name=”OrderMsg">
   <part name=”productName" type="xsd:string" />
   <part name=”quantity" type="xsd:integer" />
</message>

<portType name=”procurementPortType">
        <operation name=”orderGoods">
            <input message=”OrderMsg"/>
        </operation>
</portType>

<binding name=”ProcurementSoapBinding”
         type="procurementPortType">
   <soap:binding transport=“http://schemas.xmlsoap.org/soap/http”/>
   <operation name=“OrderGoods”>
       <soap:operation soapAction=“http://example.com/orderGoods”/>
       <input> <soap:body …/> </input>
   </operation>
</binding>

<service name=”>ProcurementService">
    <port name=”ProcurementPort" binding="ProcurementSoapBinding">
       <soap:address location="http://example.com/procurement" />
    </port>
</service>
```

WSDL Example Explanation

`types:` Specifies all the data types, either predefined or user-provided, that are used in the messages sent between the server and client.

`message:` Describes a one-way message, may it be a request or response to a request message.

`portType:` Combines multiple message elements to one one-way or round-trip operation (for instance for a request/response scheme commonly used in SOAP services).

`binding:` Specifies how the service will be implemented ‘on the wire’, meaning it gives details on how messages should be transported over the Internet. They can be sent via HTTP, using either the GET or POST method, or SOAP (which itself again can be configured to work over HTTP, or e.g., SMTP).

`service:` Defines the address for invoking the service, usually a URL where a SOAP service is deployed at.

4.5.3 WebServices

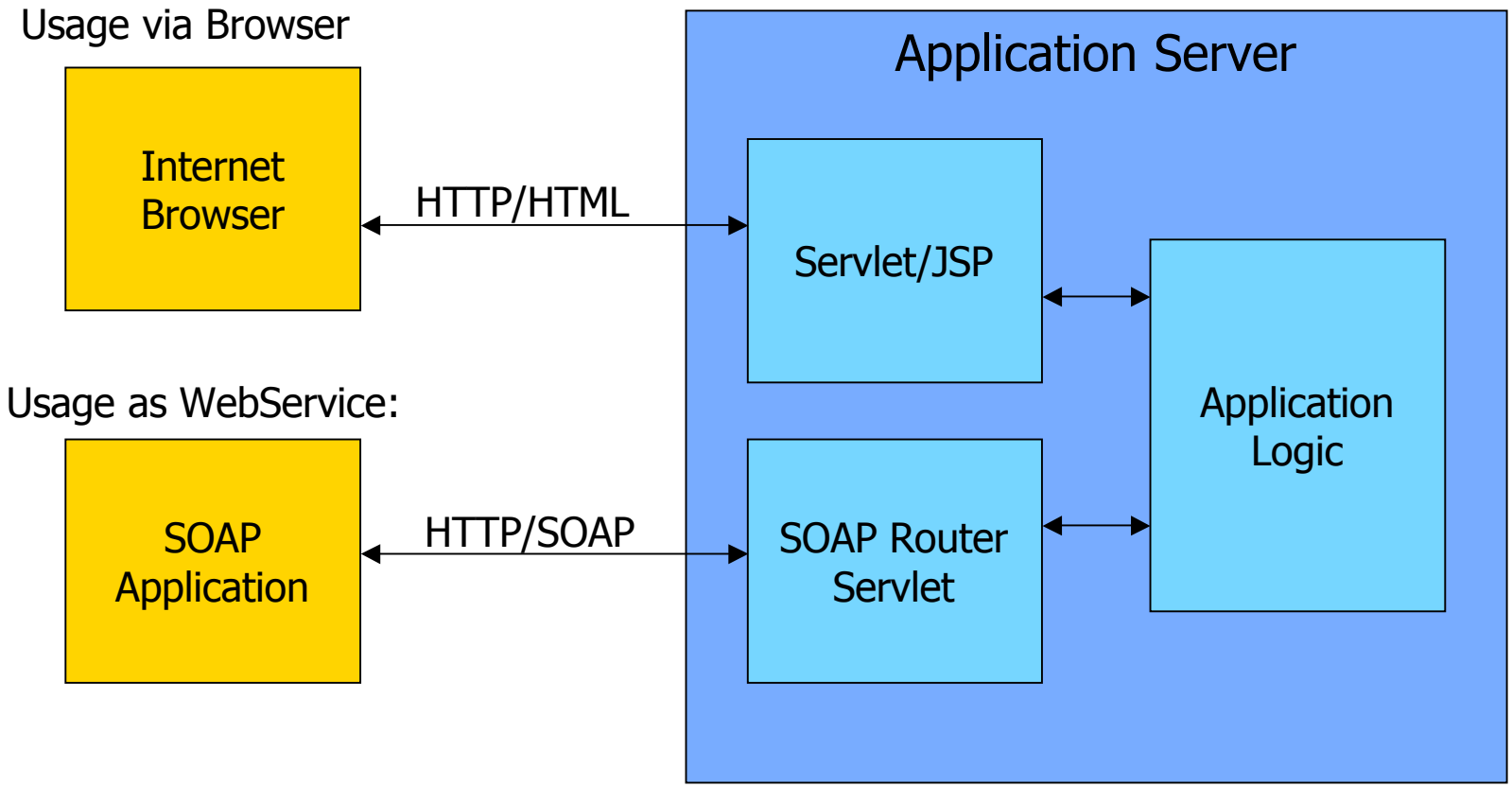
WebServices Overview

WebServices Technologies

Combination of WebService and HTML Frontend

Extending WebServices to Business WebServices

Combination of WebService and HTML Frontend



4.5.3 WebServices

WebServices Overview

WebServices Technologies

Combination of WebService and HTML Frontend

Extending WebServices to Business WebServices

Extending WebServices to Business WebServices

Simple WebServices:

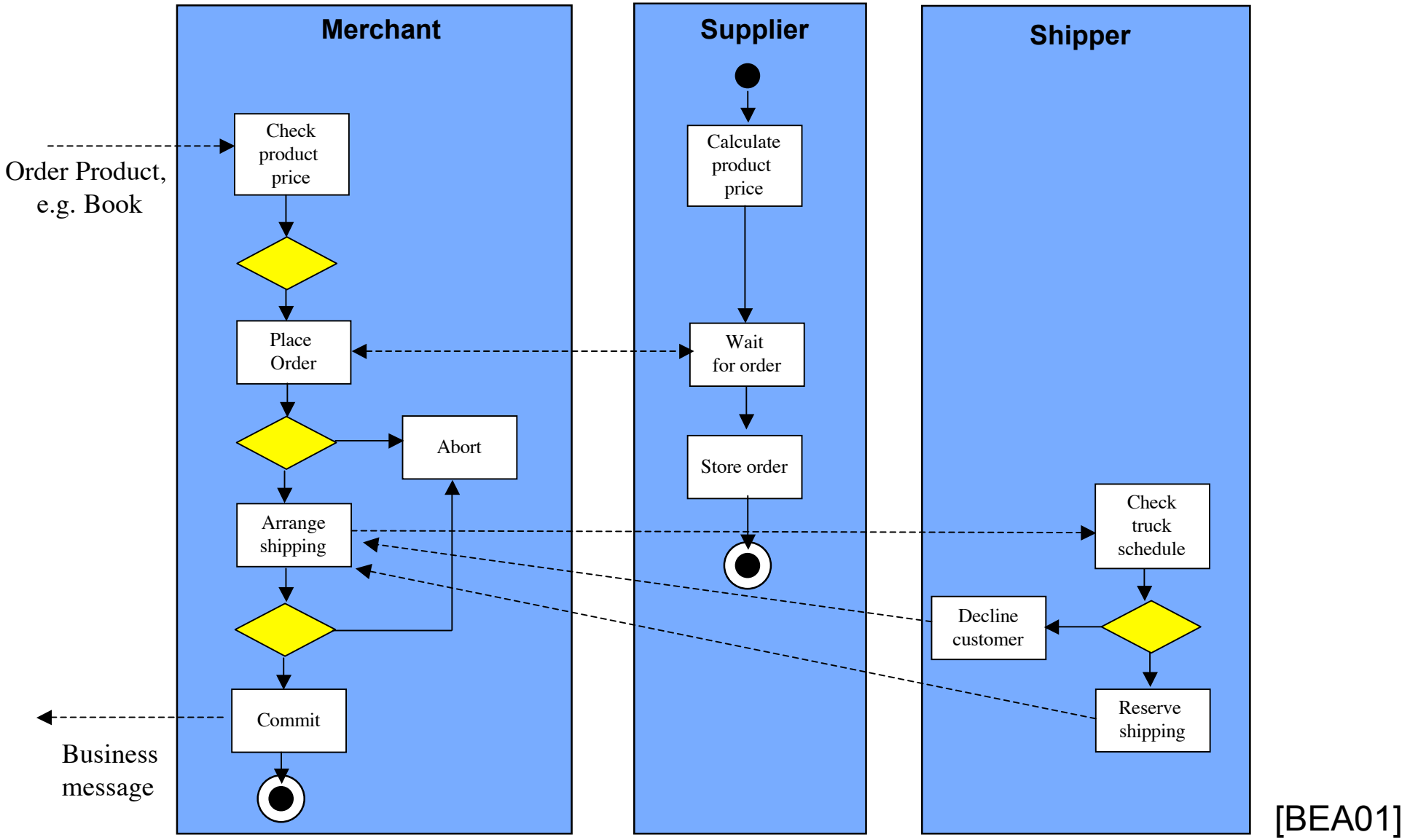
- ❑ Point-to-point connections (binary conversations)
- ❑ No transactions needed
- ❑ No / simple security measures
- ❑ Technologies: SOAP, WSDL, UDDI

Business WebServices:

- ❑ Can involve several business partners (multi-party business processes)
- ❑ Complex workflows (control-flow and information-flow, events, triggers, exceptions)
- ❑ Long-term transactions
- ❑ Security is required

➔ WebServices must be extended

Business WebServices - Example



[BEA01]

WS: Business Processes and Workflows

Proposed standards for WebServices process definitions and management:

- ebXML
  - process models,
  - information models
- IBM Conversation Support for Web Services
  - grouping of Web Services to processes
  - definition of sub-processes
- Microsoft XLANG (Web Services for Business Process Design)

WebServices: Transactions

Complex, long-term, intra-organizational and inter-organizational business processes require transaction support

Proposed standards:

- ebXML: Business Transaction Protocol (BTP)
- BEA: eXtended Open Collaboration Protocol (XOCP) (proprietary BEA protocol used for BEA WebLogic Integration)
- Transaction Authority Markup Language (XAML)
- Microsoft XLANG supports transactions

WebServices: Security

Simple WebServices:

- SOAP via HTTPS

Business WebServices require furthermore:

- Single Sign-On Facility

  Proposed Standard: Security Assertions Markup Language (SAML)

- Non-Repudiation, support for PKI und digital signatures:

  XML Key Management Specification, SOAP Security Extensions