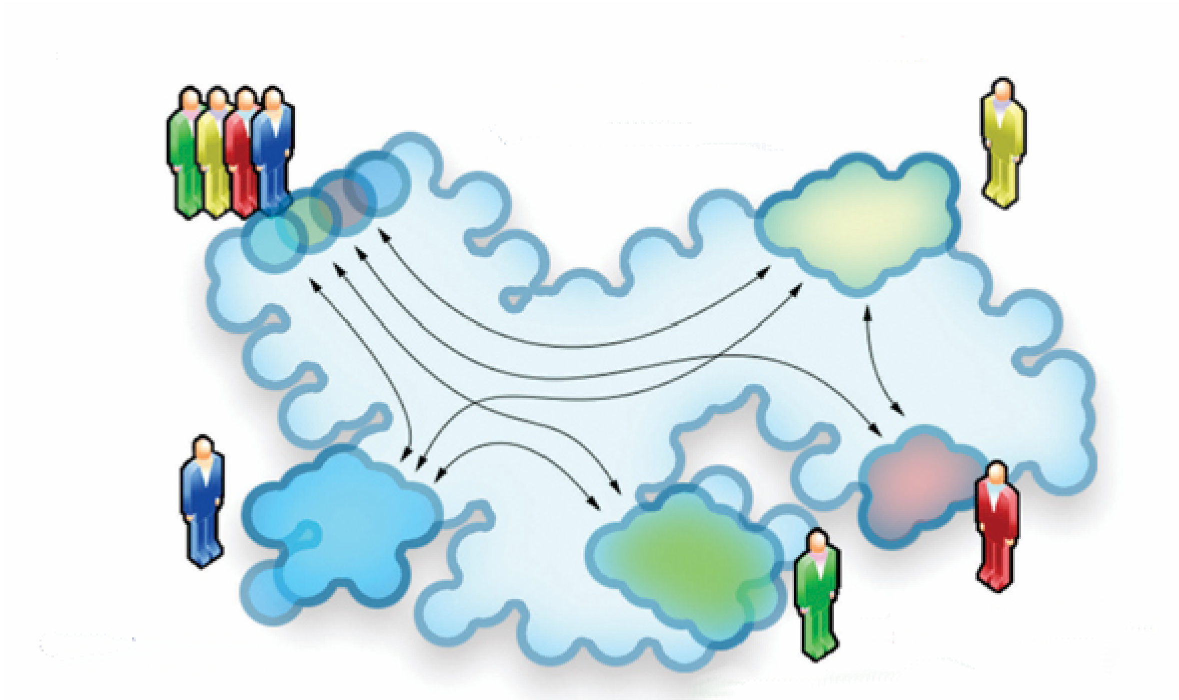
Building Standard-Based Business Processes – Part 2

with Web Services



Josef Schiefer

Vienna, Dezember 2004

Agenda

- Introduction of Oracle BPEL Process Manager
- Details + Demo of BPEL4WS Constructs
- Conclusion + Future Trends
- Business Process Monitoring with Senactive InTime
- Topics for research work (“Bakk.-Arbeiten”, “Diplomarbeiten” etc.) in the area of business process management & monitoring
- Q&A

Oracle BPEL Process Manager

What we use for the Demo

Oracle BPEL Process Manager & Designer

- Formally Collaxa BPEL Designer & Web Service Orchestration Server
- Acquired by Oracle in Jul. 2004
- One of two available native BPEL execution engines
- Available on Oracle Technology Network

http://www.oracle.com/technology/products/ias/bpel/index.html

ORACLE
BPEL PROCESS MANAGER

Why?

- Because it is available and has long and good reputation
- One of “best looking” tools under Eclipse

The good, the bad and the ugly

What we like …

- The Eclipse Design Tool
- The simple Setup of Designer, Application Server and the BPEL Engine
- The simple development/deployment cycle
- The good tutorials, whitepapers and samples
- The Web Console of the BPEL Process Manager
- Available for JBoss and BEA Weblogic

What could be better …

- Complete support of BPEL (e.g. correlations are missing)
- More support for BPEL in the Source Editor (e.g. AutoComplete of BPEL tags)
- Help support in Eclipse

What we miss …

- Oracle discontinues the superb Collaxa BPEL blog
- More information and controlling of the deployment cycle
- A “real” Debugger

BPEL Constructs + Demo

Simple Activities

Receive

- » Wait for a partner inbound message
- » Can be the instantiator of the business process

Reply

- » Synchronous response to a <receive> activity
- » Response to the inbound <receive> from a partner

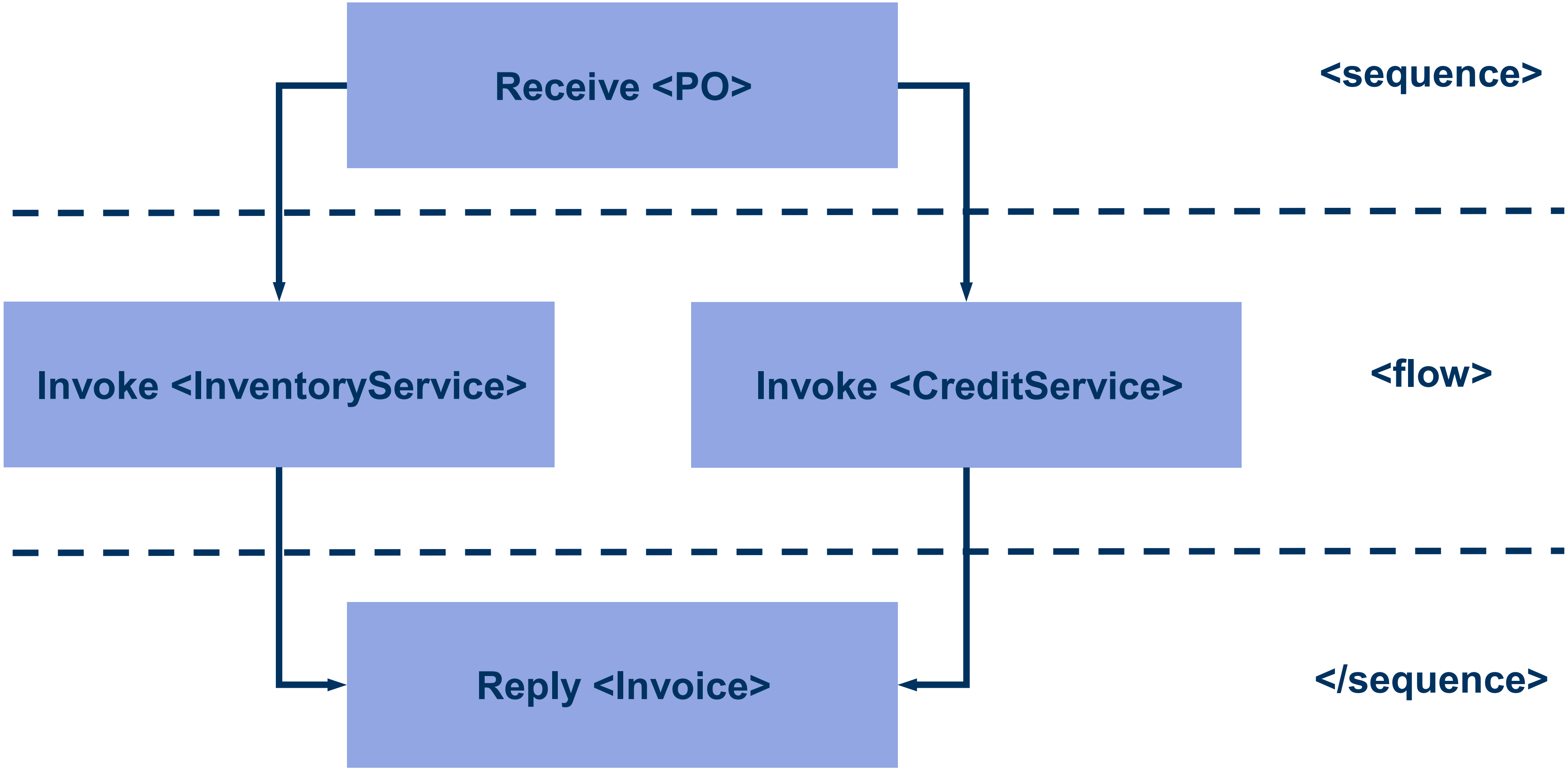
Invoke

- » Issue a request synchronously (two-way) *or* asynchronously (one-way)

Pick

- » Specify an inbound set of messages
- » Can be the instantiator of the business process
- » Activity completes when one of the messages arrives

Simple Activities Combined with Structured Activities



Receive <PO>

<sequence>

Invoke <InventoryService>

Invoke <CreditService>

<flow>

Reply <Invoice>

</sequence>

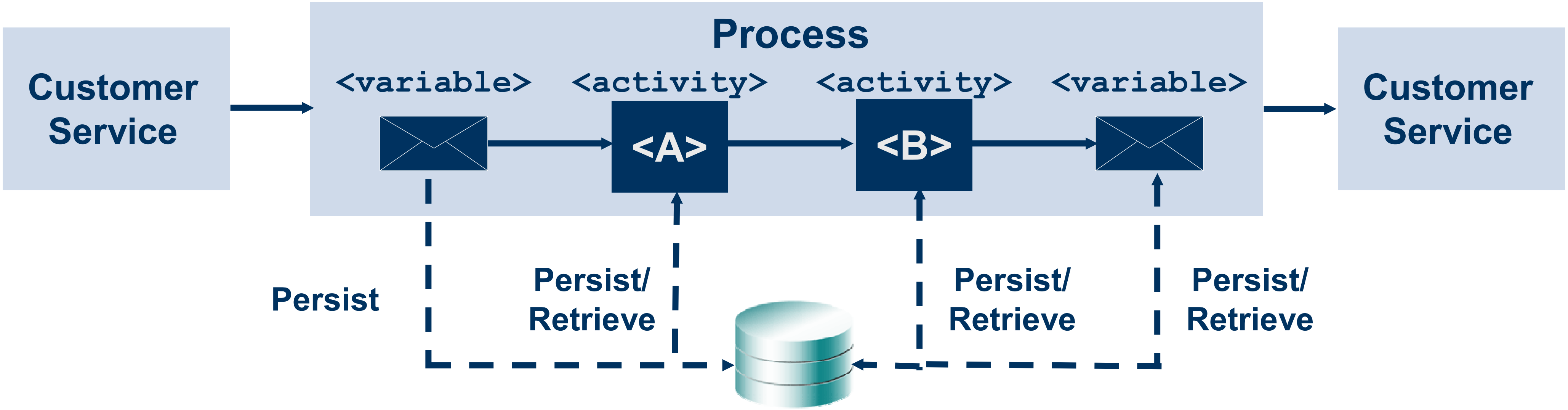
Sample Activities in BPEL

```
<sequence>
    <receive partnerLink=“customer” portType=“lns:purchaseOrderPT"
              operation=“sendPurchaseOrder” variable=“PO”
              createInstance="yes" />
    <flow>
     <invoke partnerLink=“inventoryChecker” portType=“lns:inventoryPT”
             operation="checkINV" inputVariable="inventoryRequest"
             outputVariable="inventoryResponse" />

     <invoke partnerLink="creditChecker" portType=“lns:creditPT"
             operation="checkCRED" inputVariable="creditRequest"
             outputVariable="creditResponse" />
    </flow>
    ...
    <reply partnerLink=“customer” portType=“lns:purchaseOrderPT”
             operation=“sendPurchaseOrder” variable=“invoice"/>
</sequence>
```

Variables

Messages sent and received from partners

- » Persisted for long running interactions
- » Defined in WSDL types and messages
- » Used in most BPEL activities for input and output data



Access to Variables (with Properties)

» Properties allow public access to variables

```
<property name="userID" type="xsd:string"/>

 <propertyAlias propertyName="userID‟
   messageType="orderDetails‟ part="identification‟
   query="/credentials"/>
```

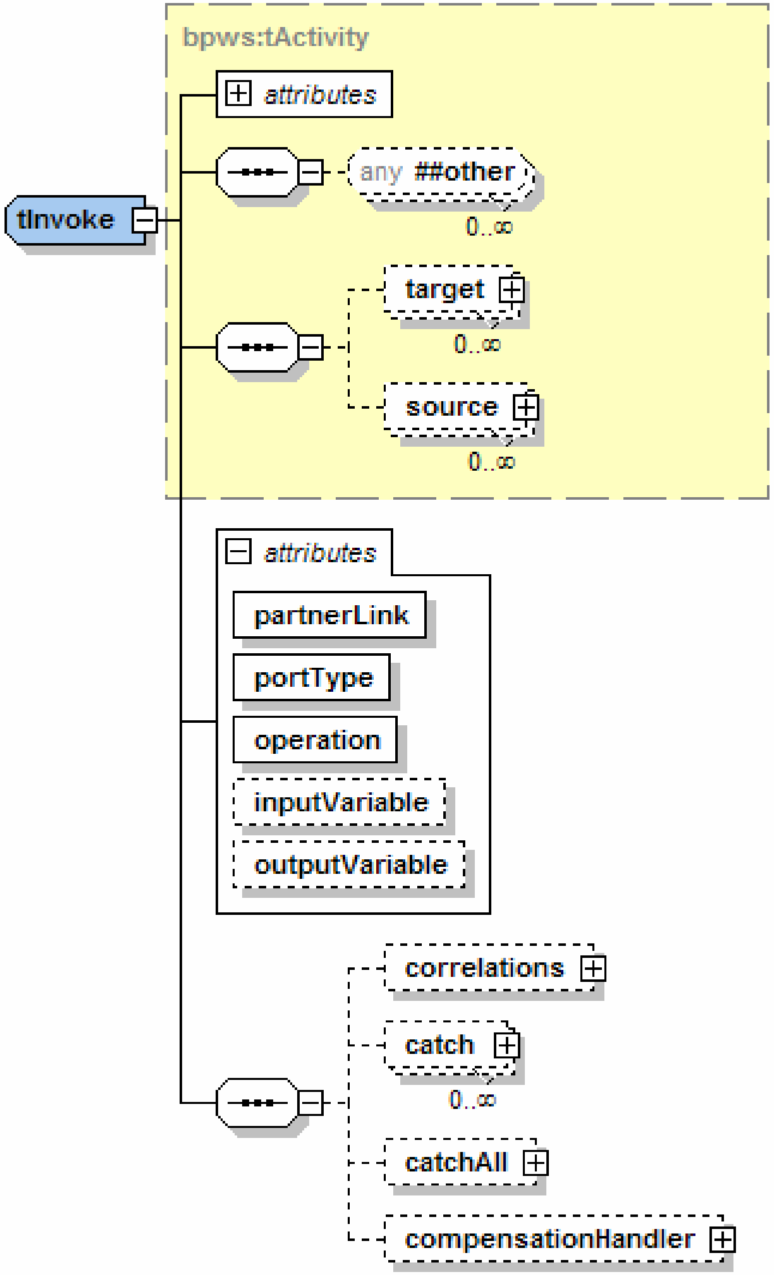
» Correlation sets are defined by properties

```
<correlationSets>
  <correlationSet name=“User‟
    properties="userID"/>
</correlationSets>
```

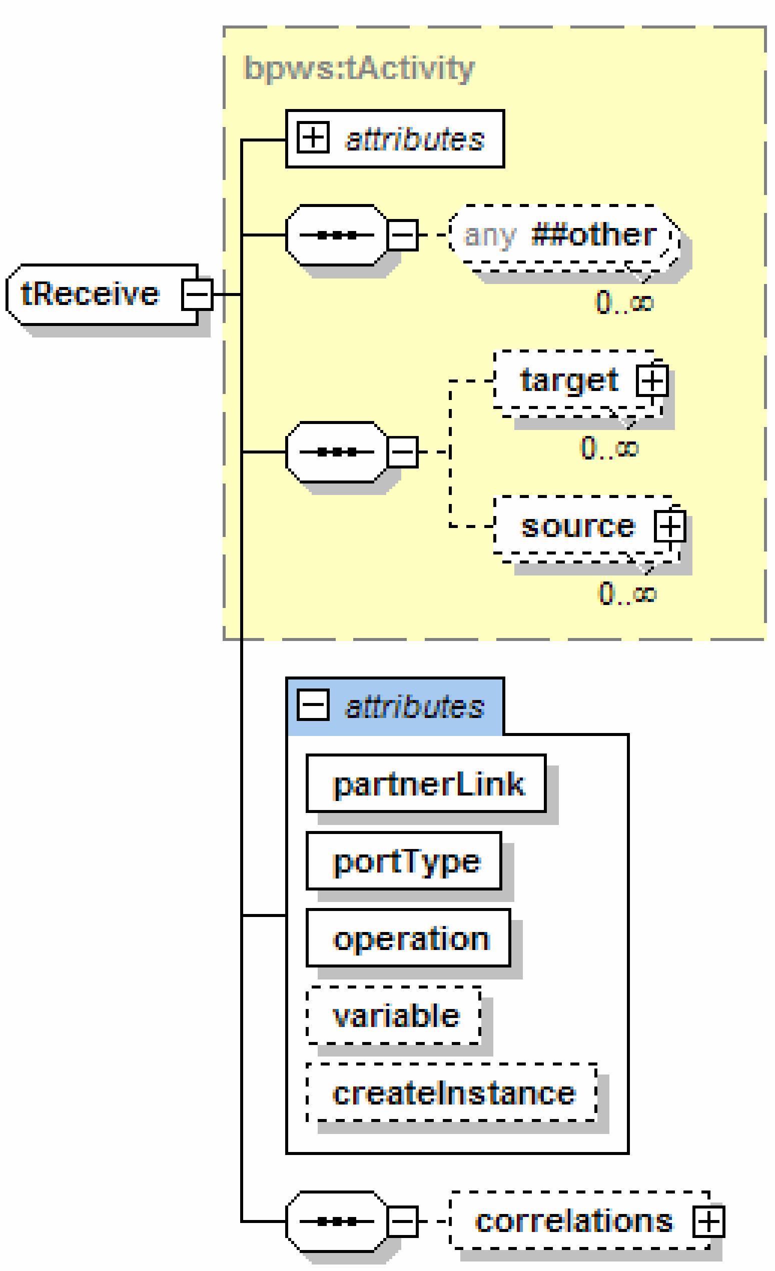
» Access to variables in expressions with **bpws:getVariableProperty ('variableName', 'propertyName')** or **bpws:getVariableData ('variableName', 'partName'?, 'locationPath'?)**

```
  <switch>
      <case condition="bpws:getVariableProperty(Order,‘props:userId‘)=‘Maier‘">
         ...
      </case>
      <case condition="bpws:getVariableData(Order,‘identification‘,‘/credentials‘)=‘Huber‘">
         ...
      </case>
         ...
</switch>
```

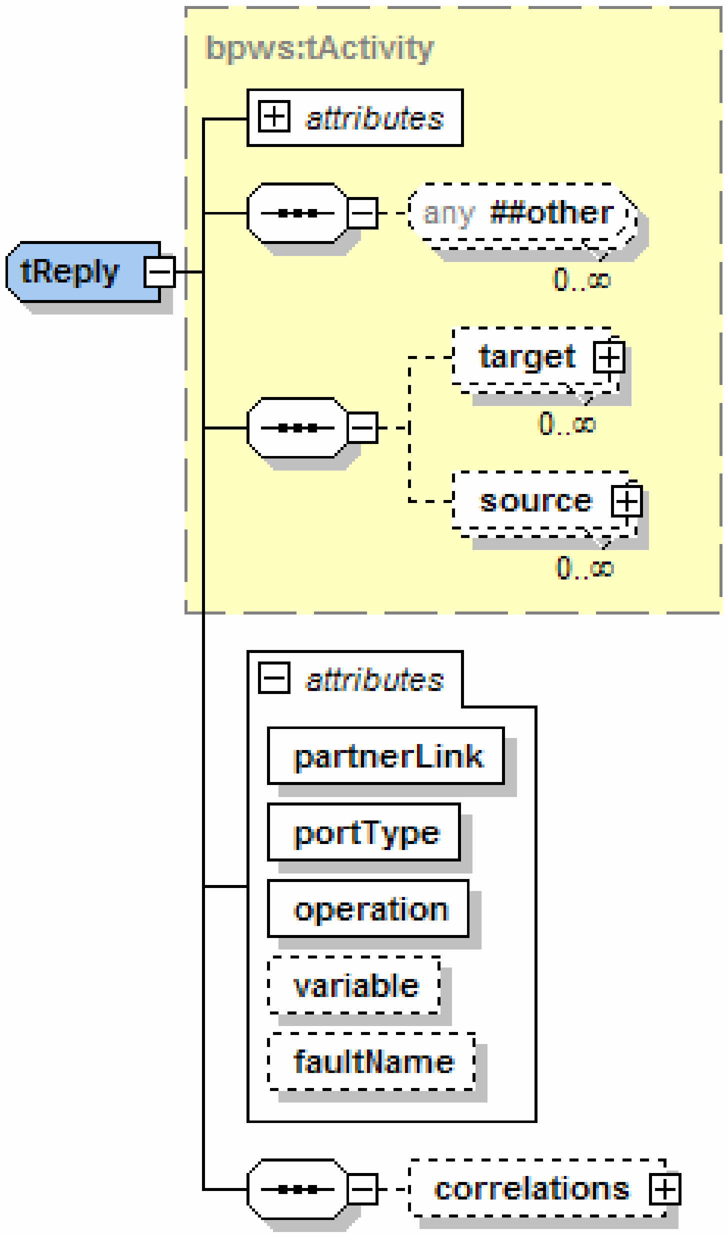
<invoke> Activity


bpws:tActivity
attributes
any ##other
0..∞
tInvoke
target
0..∞
source
0..∞
attributes
partnerLink
portType
operation
inputVariable
outputVariable
correlations
catch
0..∞
catchAll
compensationHandler

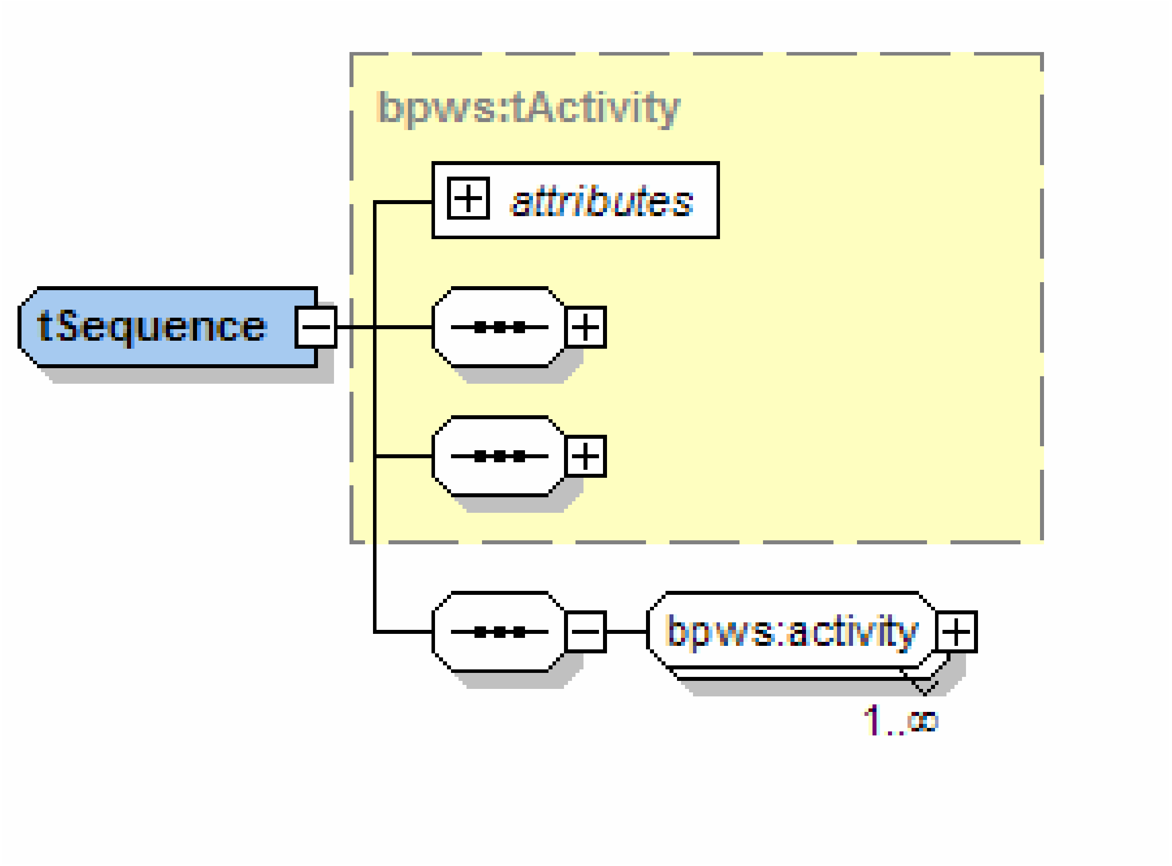
<receive> Activity


tReceive
bpws:tActivity
attributes
any ##other
0..∞
target
0..∞
source
0..∞
attributes
partnerLink
portType
operation
variable
createInstance
correlations

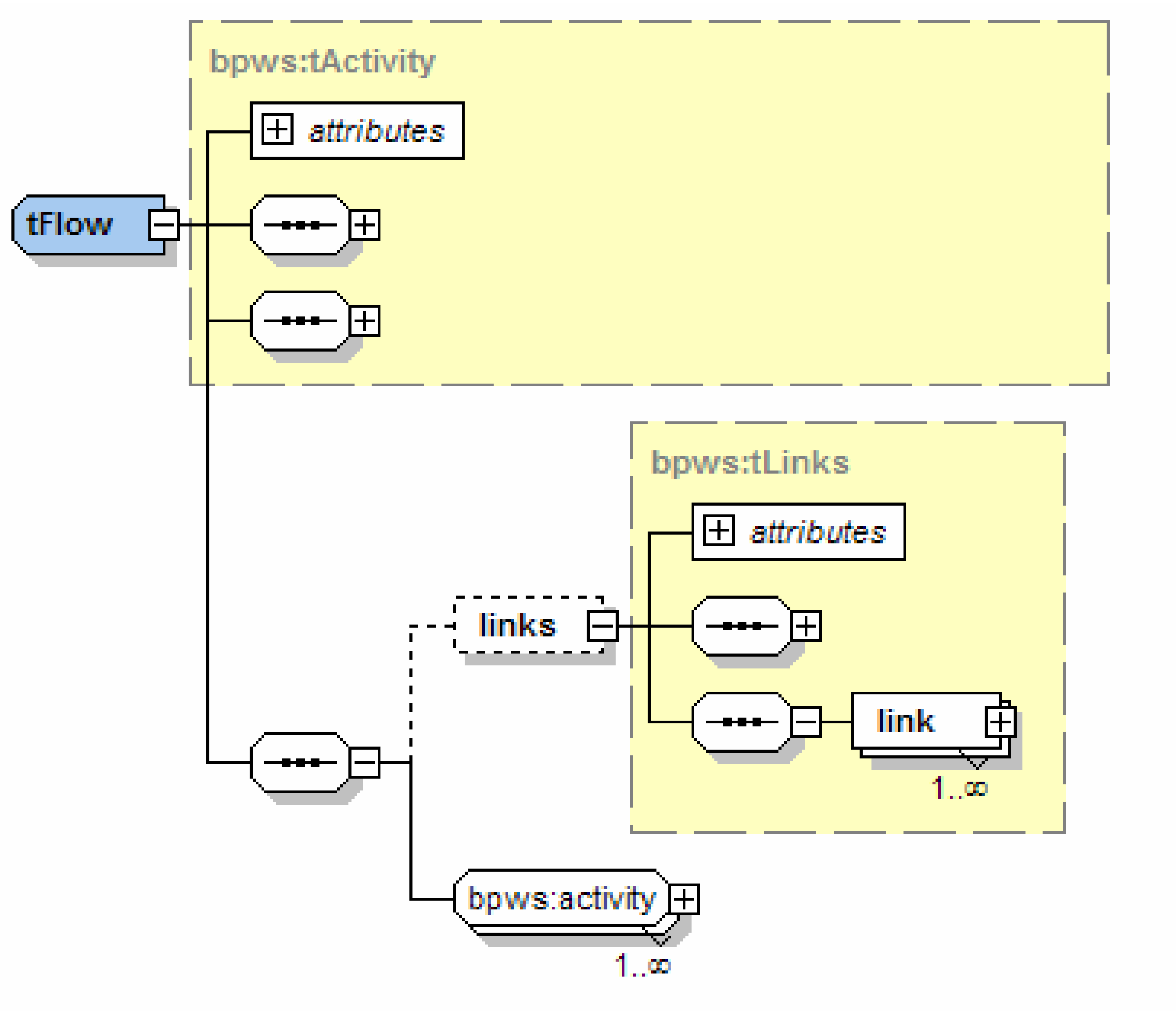
<reply> Activity


bpws:tActivity
attributes
any ##other
0..∞
target
0..∞
source
0..∞
tReply
attributes
partnerLink
portType
operation
variable
faultName
correlations

<sequence> Activity

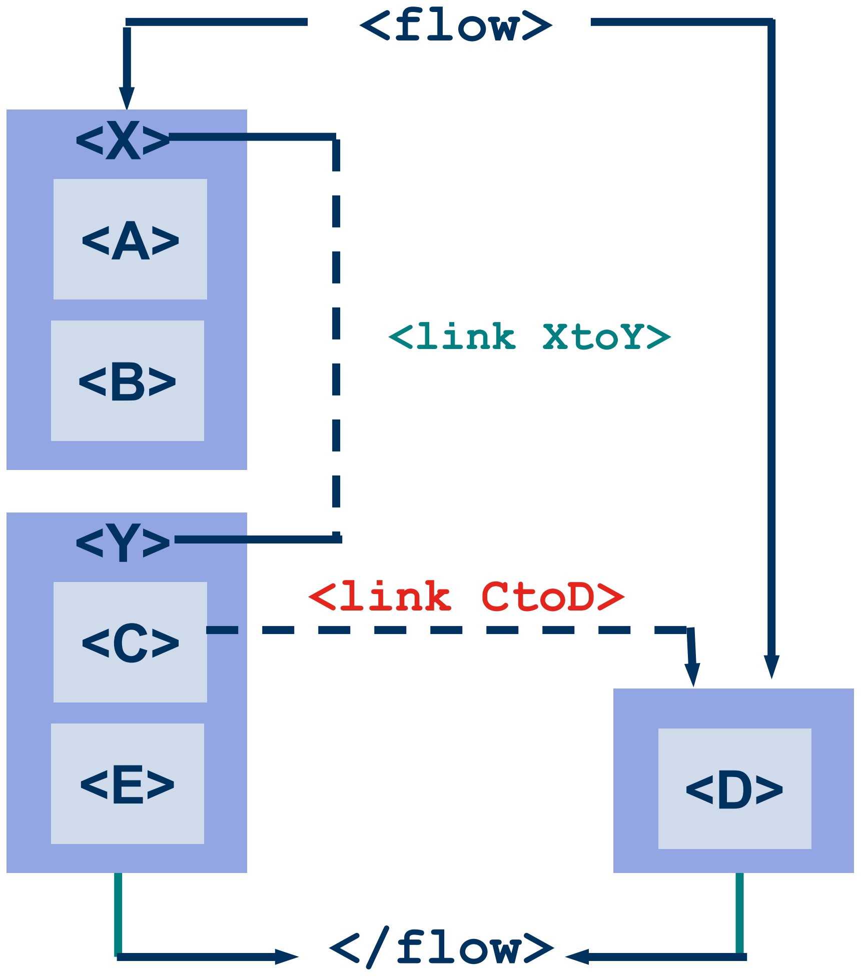


<flow> Activity

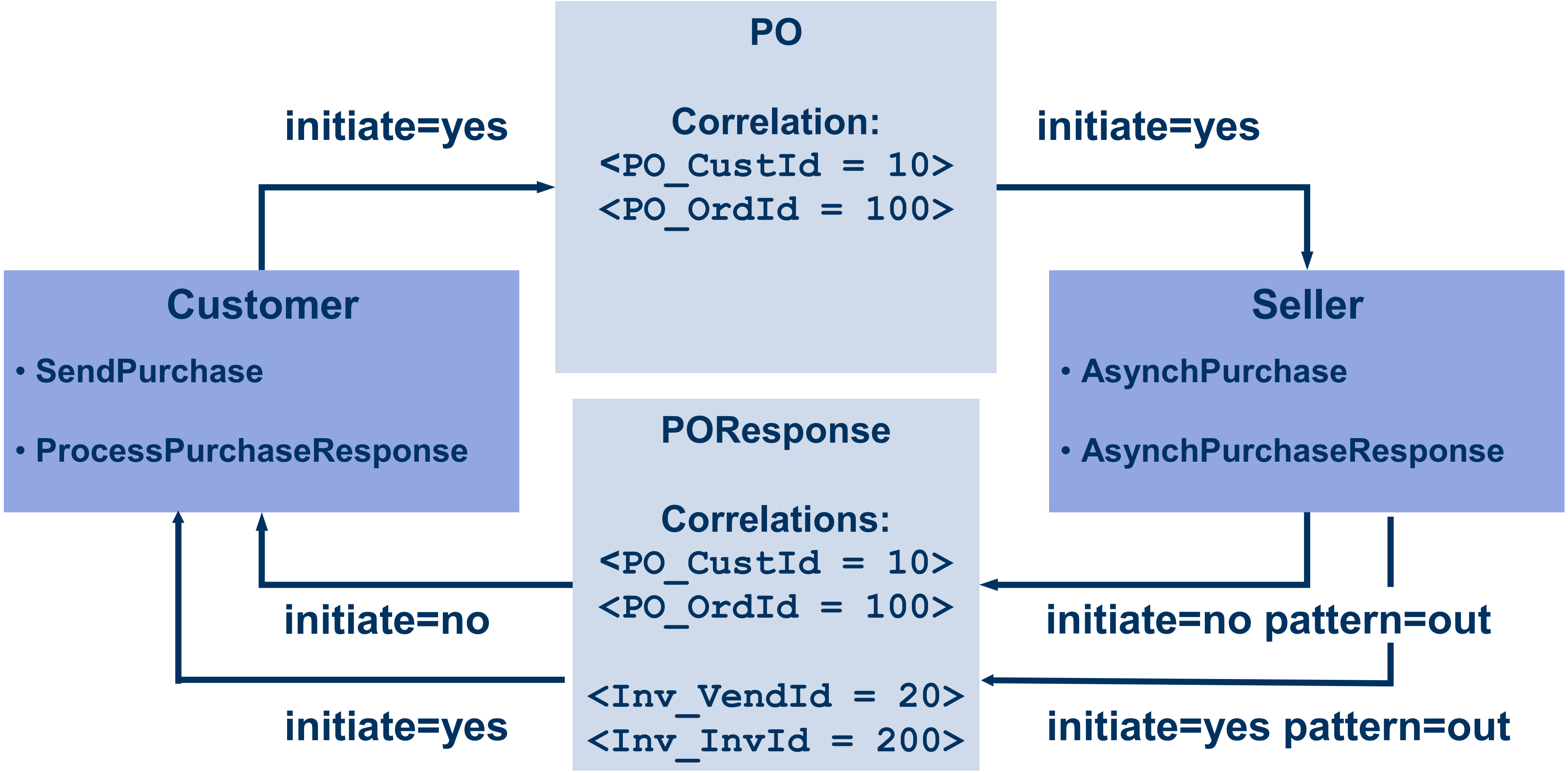

bpws:tActivity
attributes
tFlow
bpws:tLinks
attributes
links
link
1..∞
bpws:activity
1..∞

Links – Control Flow

```
<flow>
 <links>
  <link name="XtoY"/>
  <link name="CtoD"/>
 </links>
 <sequence name="X">
   <source linkName="XtoY"/>
   <invoke name="A" .../>
   <invoke name="B" .../>
 </sequence>
 <sequence name"Y">
   <target linkName="XtoY"/>
   <receive name="C"/>
     <source linkName="CtoD"/>
   </receive>
   <invoke name="E" .../>
 </sequence>
 <invoke partnerLink="D">
   <target linkName="CtoD"/>
 </invoke>
</flow>
```



Correlations in BPEL



PO

Correlation:
<PO_CustId = 10>
<PO_OrdId = 100>

initiate=yes

initiate=yes

Customer
- SendPurchase
- ProcessPurchaseResponse

Seller
- AsynchPurchase
- AsynchPurchaseResponse

POResponse

Correlations:
<PO_CustId = 10>
<PO_OrdId = 100>

<Inv_VendId = 20>
<Inv_InvId = 200>

initiate=no

initiate=yes

initiate=no pattern=out

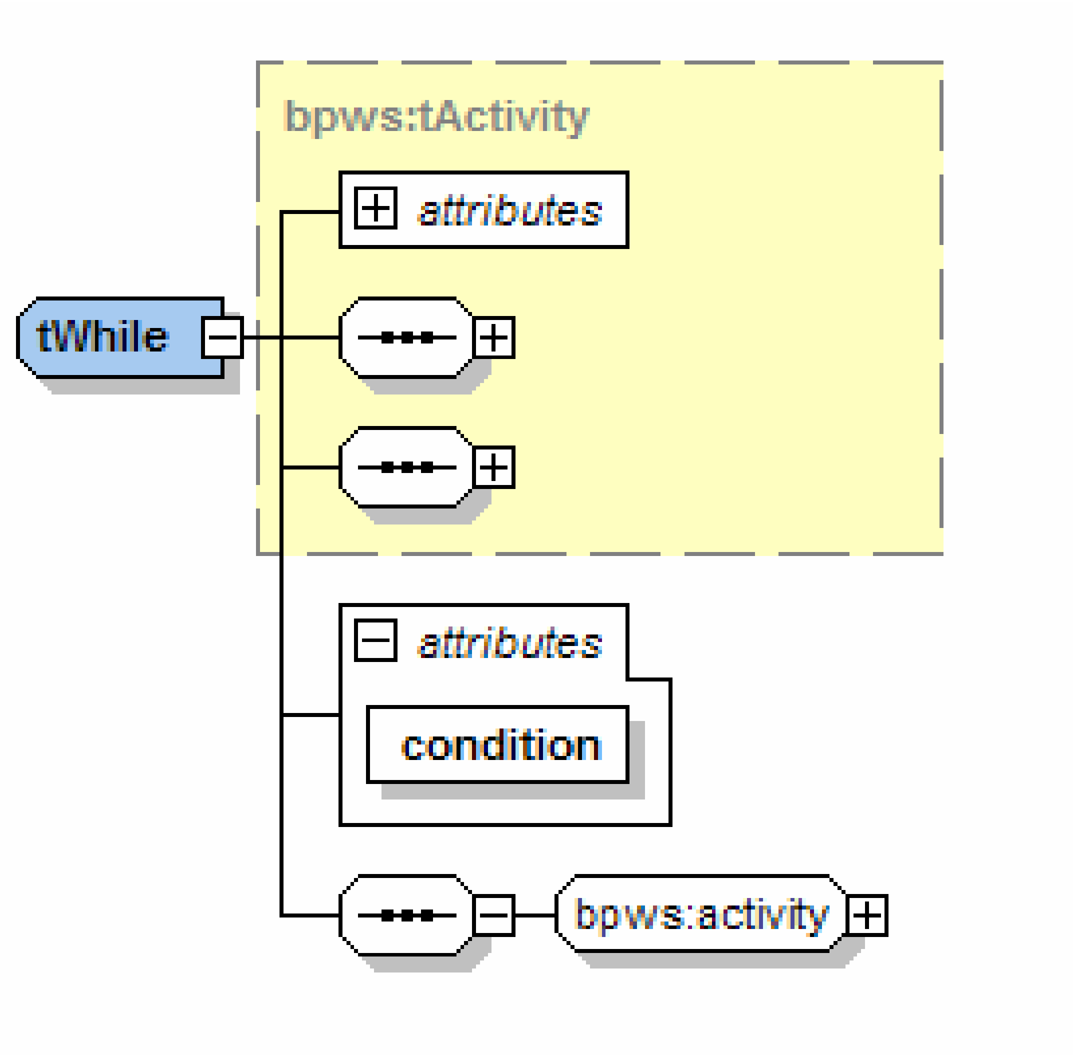
initiate=yes pattern=out

Correlations in BPEL

Seller perspective:

```
<correlationSets>
 <correlationSet name="POCorr" properties="cor:custId cor:ordId"/>
 <correlationSet name="InvoiceCorr" properties="cor:vendId cor:invId"/>
</correlationSets> ...

<receive partnerLink=“Customer” portType="SP:PurchaseOrderPT"
    operation="AsynchPurchase" variable="PO">
  <correlations>
    <correlation set="POCorr" initiate="yes">
  </correlations>
</receive> ...

<invoke partnerLink=“Customer” portType="SP:CustomerPT"
  operation=“ProcessPurchaseResponse" inputVariable="POResponse">
 <correlations>
  <correlation set="POCorr" initiate="no" pattern="out">
  <correlation set="InvoiceCorr" initiate="yes" pattern="out">
 </correlations>
</invoke> ...
```

<while> Activity


bpws:tActivity
attributes
tWhile
attributes
condition
bpws:activity

BPEL Demo

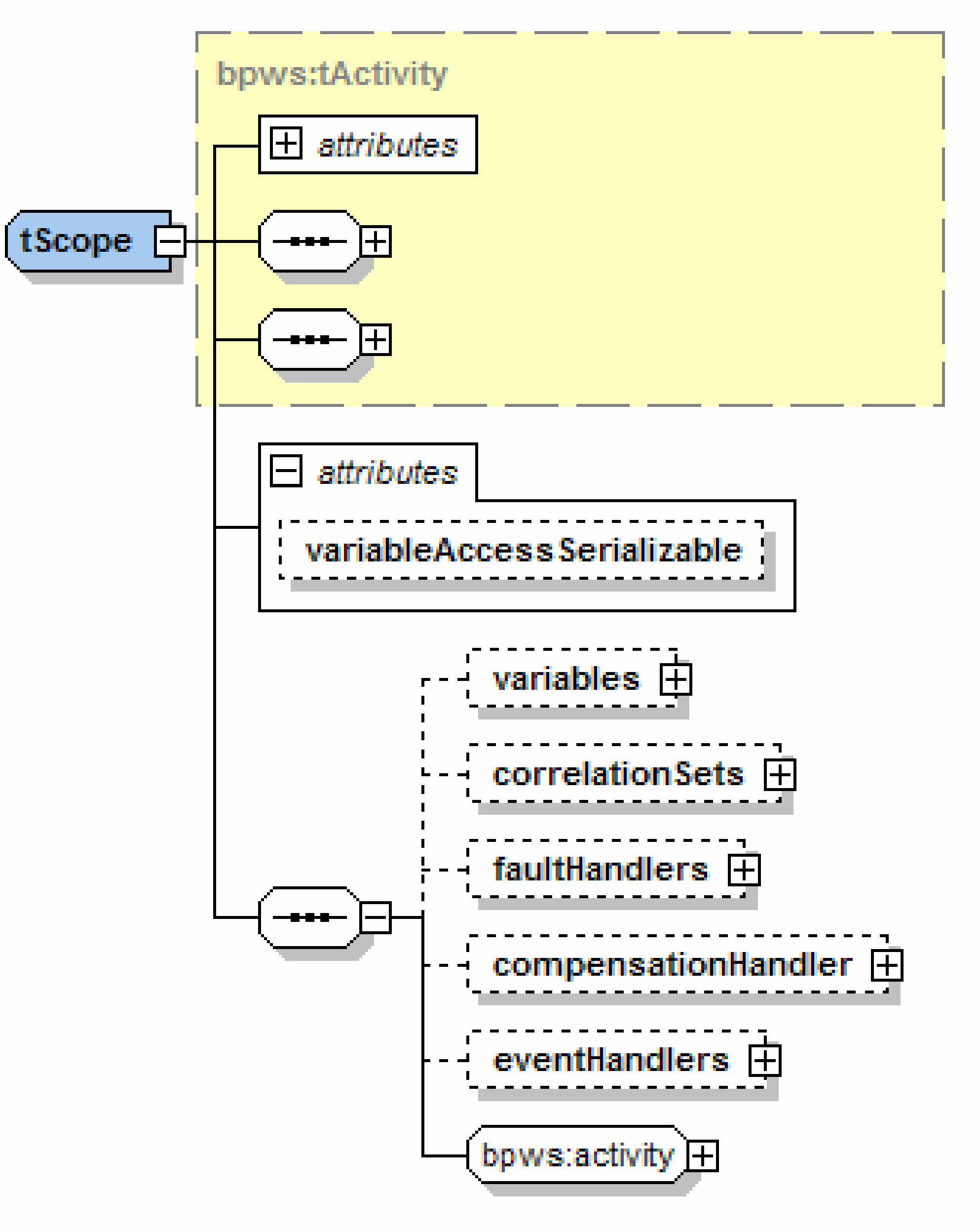
Cont.

Scopes in BPEL

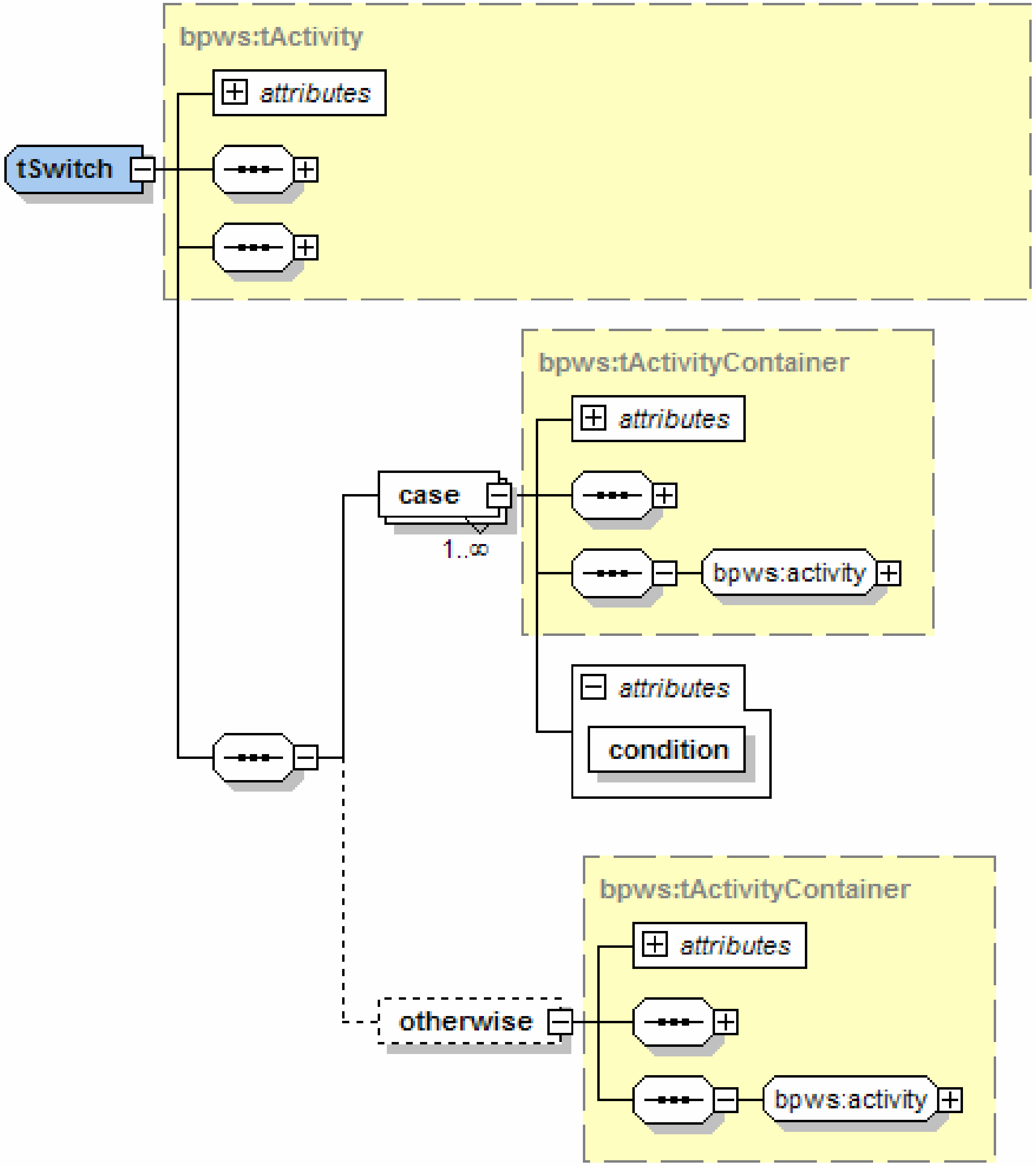
- Provide a shared context for subset of activities
- Can contain
  - fault handlers
  - event handlers,
  - compensation handler
  - variables
  - correlation sets
- Can serialize concurrent access to variables

```
<scope
variableAccessSerializable="yes|no‟
 ...>

    <variables>
    </variables>

    <correlationSets>? ...
    </correlationSets>

    <faultHandlers>
    </faultHandlers>

    <compensationHandler>? ...
    </compensationHandler>

    <eventHandlers>
    </eventHandlers>
    (activities)*

</scope>
```

<scope> Activity


bpws:tActivity
attributes
tScope
attributes
variableAccessSerializable
variables
correlationSets
faultHandlers
compensationHandler
eventHandlers
bpws:activity

<switch> Activity


bpws:tActivity
attributes
tSwitch
case
1..∞
bpws:tActivityContainer
attributes
bpws:activity
attributes
condition
otherwise
bpws:tActivityContainer
attributes
bpws:activity

<switch> Activity

Defines conditional behavior

```
<switch>
    <case condition="bpws:getVariableProperty(stockResult,level)> 100">
        <flow>
        ...
        </flow>
    </case>
    <case condition="bpws:getVariableProperty(stockResult,level) >= 0">
        <throw faultName="FLT:OutOfStock" variable="RestockEstimate"/>
    </case>
    <otherwise>
        <throw faultName="FLT:ItemDiscontinued"/>
    </otherwise>
</switch>
```

BPEL Demo

Cont.

Conclusion + Future of BPEL

BPEL4WS Process Engines

Oracle BPEL Process Manager …

IBM WBISF (Websphere Business Integration Server Foundation)

- » Websphere Process Choreographer (BPEL4WS Process Engine)
- » IBM Websphere Integration Modeler V5.1: Based on Eclipse, includes BPEL Process Designer
- » Proprietary extensions in order to support well IBM software infrastructure (MQSeries, Websphere Application Server, etc.)
- » http://www.ibm.com/software/integration/wbisf/, http://www-306.ibm.com/software/integration/wbimodeler/

BPEL4WS Process Engines

Active BPEL Engine / Active Workflow Designer

- Open Source
- Active Workflow Designer: based on Eclipse, allows simulation of BPEL processes
- http://www.activebpel.org, http://www.activeendpoints.com/products/activewebflow/awfpro/index.html

ObjectWeb MidOffice BPEL Engine (future)

- Will be the first real Open Source engine
- Available in 2005 (product of the company eMaxx, is on transition to ObjectWeb)

Current Situation with BPEL4WS…

- Tool support is very crucial for managing BPEL4WS processes
- Large corporation (IBM, Microsoft, Oracle, SAP) replace their proprietary process languages with BPEL4WS
- Good start BUT many issues are still open to become more “business” oriented:
  - No data translation (EDI, binary formats …)
  - No organizational model (business units, departments, roles)
  - No security
  - No human workflow
  - No trading partner agreements
  - No auditing interfaces
  - Silent on existing business protocols (ebXML, RosettaNet …)
  - Silent on non Web service interactions (e.g. java to java)
  - . . .

Compare BPEL to ebXML BPSS

Common to both

- OASIS Group responsible for both specs (ebXML and BPEL4WS)
- Both use SOAP as message format
- Both are below Business Process Management and Modeling Tools (such as ARIS/Adonis)

Unique to BPEL

- Focus Choreography and Orchestration
- Industry driven
- Executable process definitions
- Implementation friendly
- Compatible to all WS-* specs

Unique to BPSS

- Focus Choreography
- Modeling language (at least we know no direct execution engine)
- Mature peer-to-peer business communication (including complex multi party transactions)
- Organization driven (OASIS and UN/CEFACT)
- Sophisticated messaging, QoS and business description possibilities

Will BPEL replace ebXML?

No, because …

- They are complementary and not competitive specifications
- They are targeting different problems
  - BPSS (as part of ebXML) should replace all EDI-based communication
  - BPEL is mainly an executable language, but services can also have business value
- The OASIS Group develops ebXML to support of the WS-* specs

Yes (maybe), because …

- ebXML and BPSS are mature technologies but not well known (only commerce in Asia makes heavy usage of them)
- BPEL4WS is “hype” and directly supported by easy to use products
- BPEL4WS can be used for most common problems of choreography and orchestration → BPEL4WS will become first choice for many companies

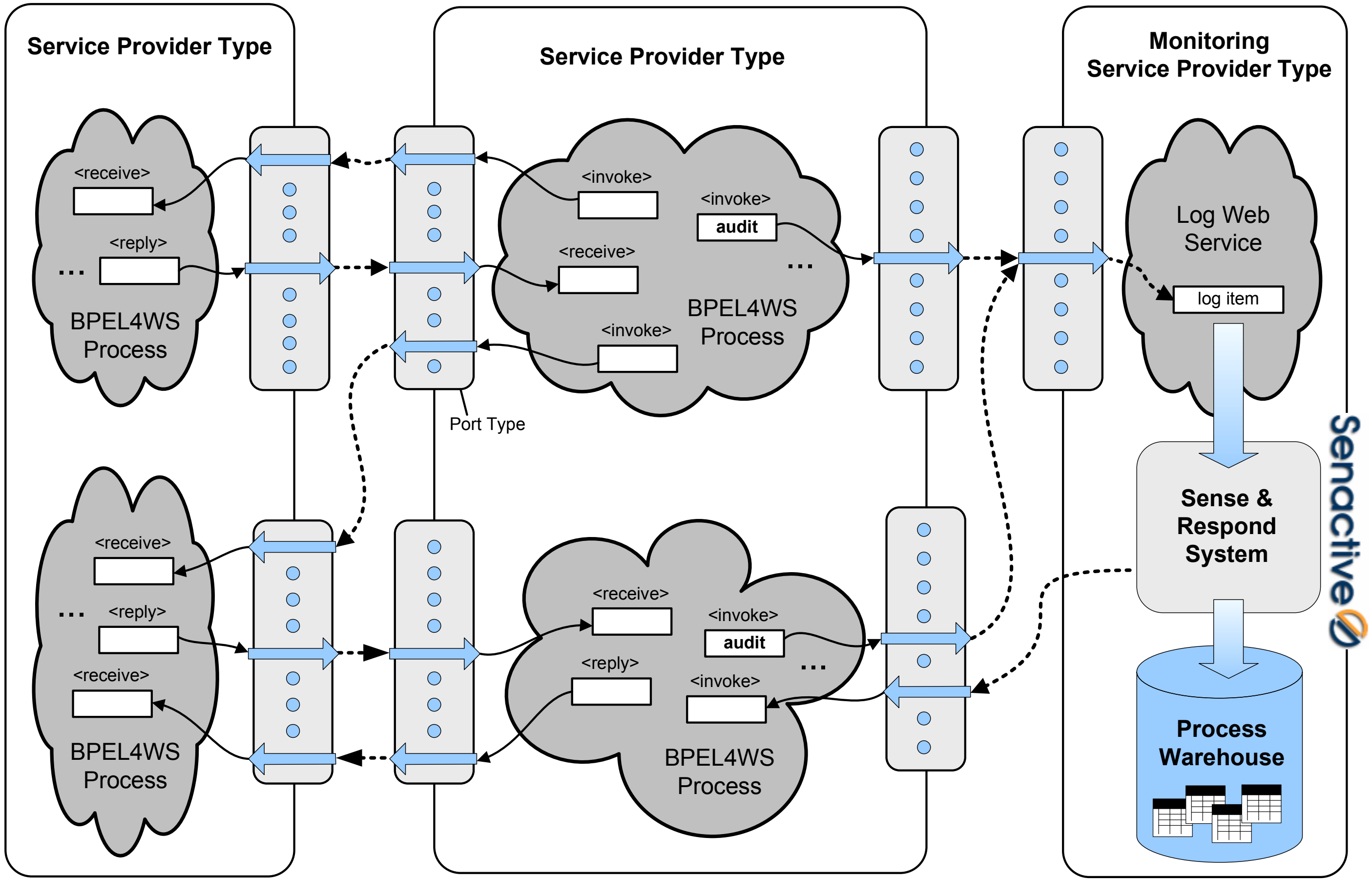
So, what to do?

- Have in mind that each technology serves other masters
- If you need bullet proof and reliable business communication, use ebXML
- If you already have web services, use BPEL4WS because the integration is easier
- You always can combine parts of WS-* and ebXML (e.g. a service which bridges ebMS and WS-Messaging)

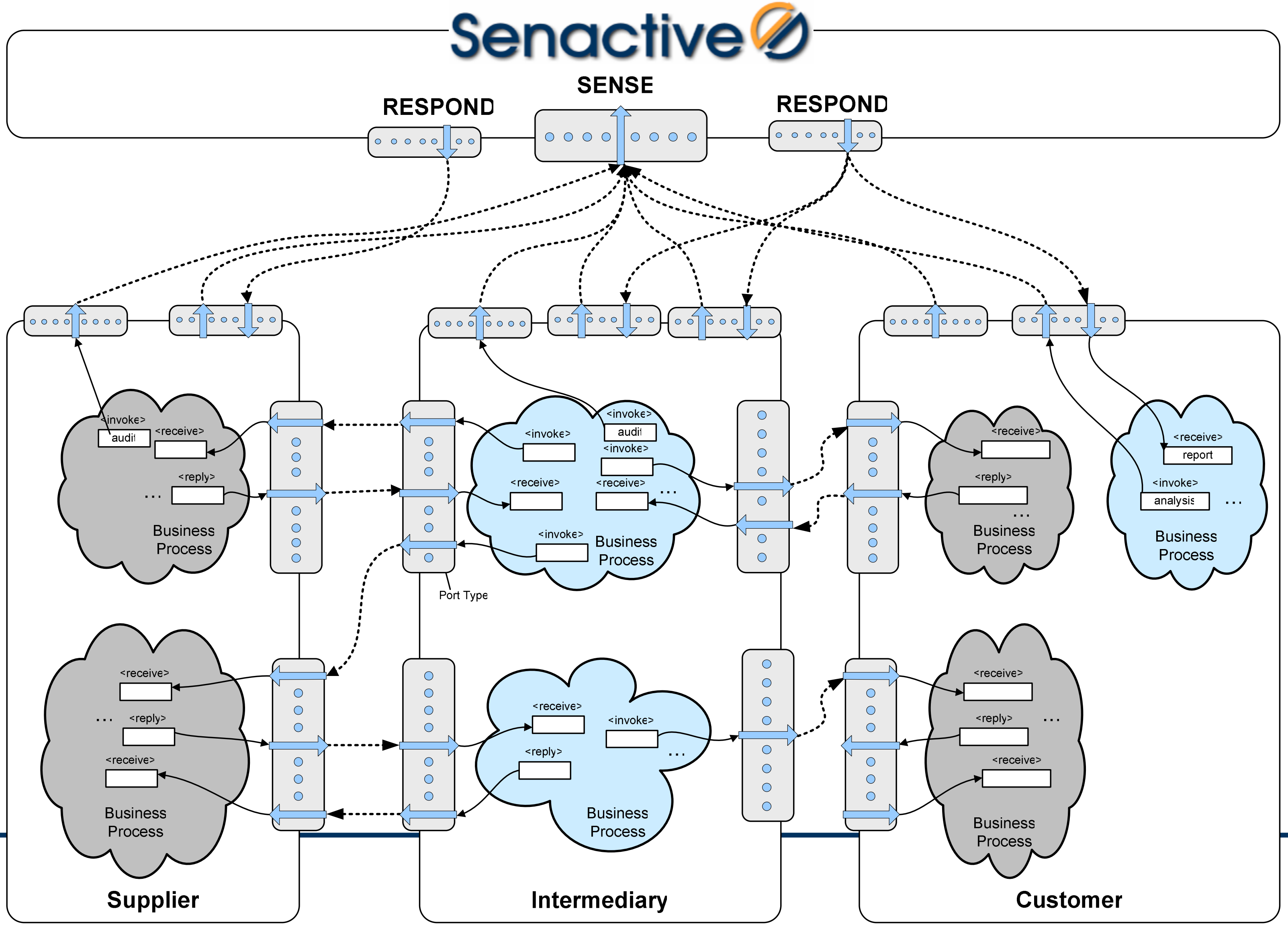
Business Process Monitoring

Senactive

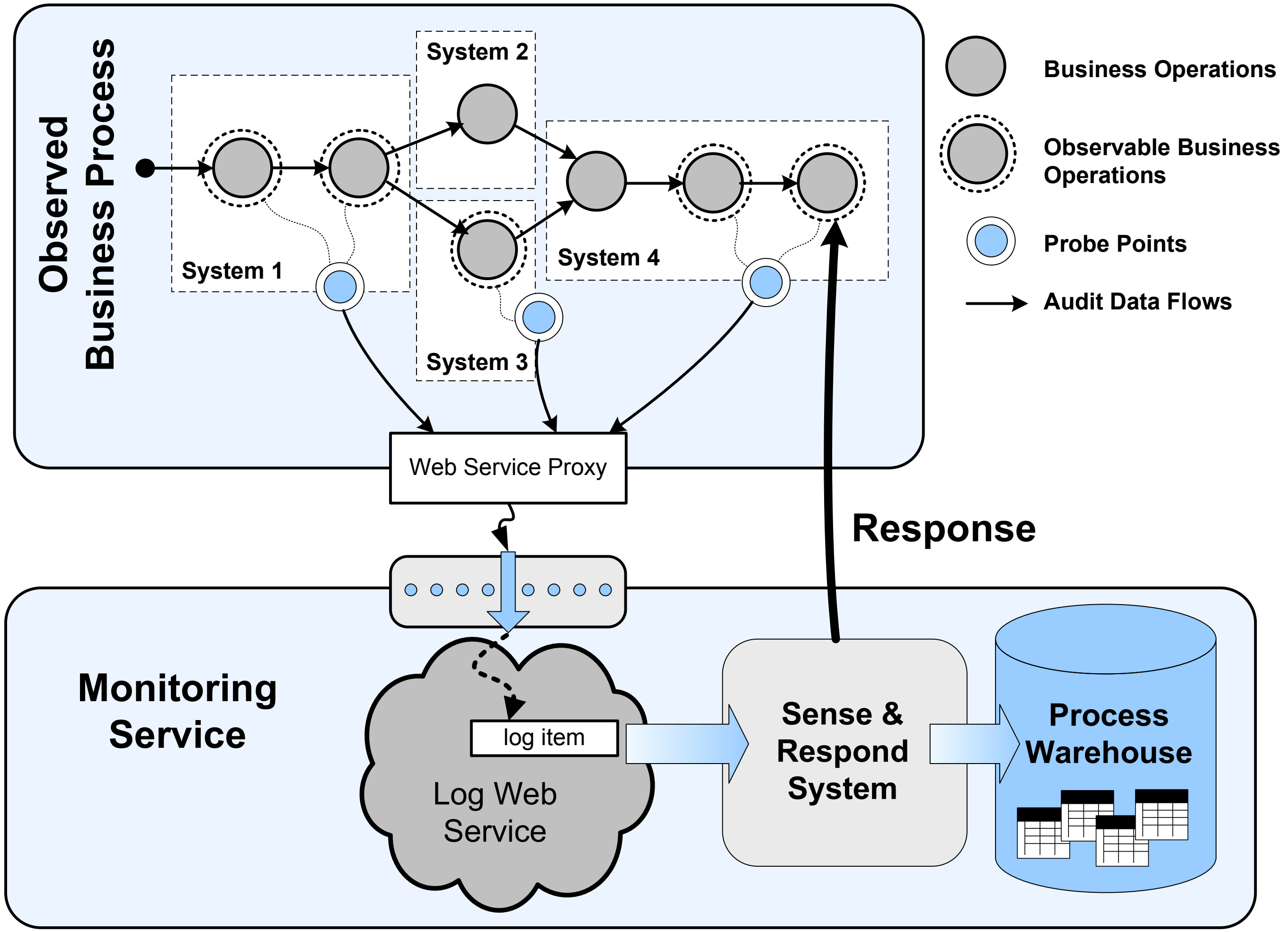
Monitoring of BPEL4WS Processes with Senactive



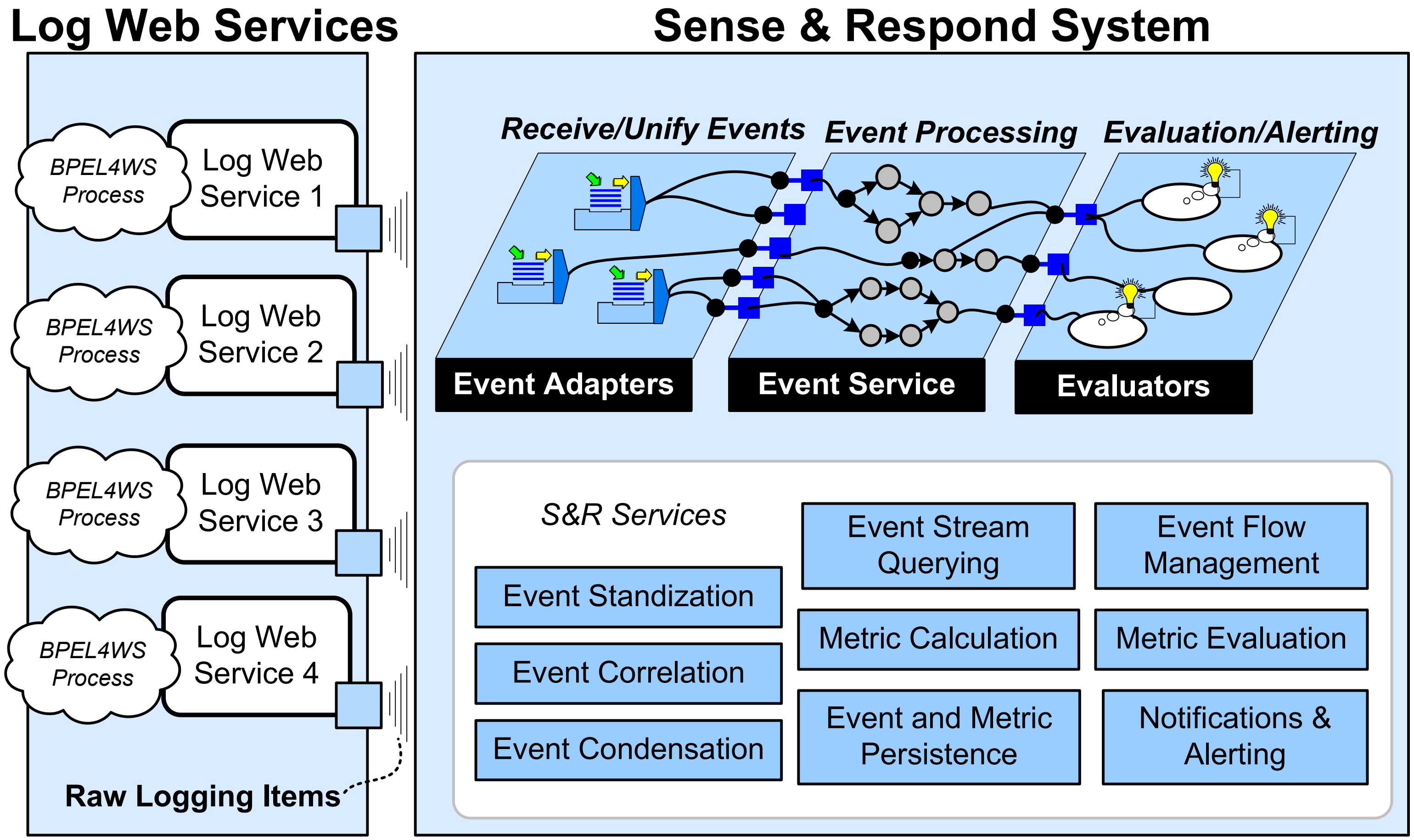
Supply Chain Monitoring of BPEL4WS Processes



Monitoring of Non-BPEL4WS Processes with Senactive



Sense & Respond System



Log Web Services

BPEL4WS Process

Log Web Service 1

BPEL4WS Process

Log Web Service 2

BPEL4WS Process

Log Web Service 3

BPEL4WS Process

Log Web Service 4

Raw Logging Items

Sense & Respond System

Receive/Unify Events

Event Processing

Evaluation/Alerting

Event Adapters

Event Service

Evaluators

S&R Services

Event Standization

Event Correlation

Event Condensation

Event Stream Querying

Metric Calculation

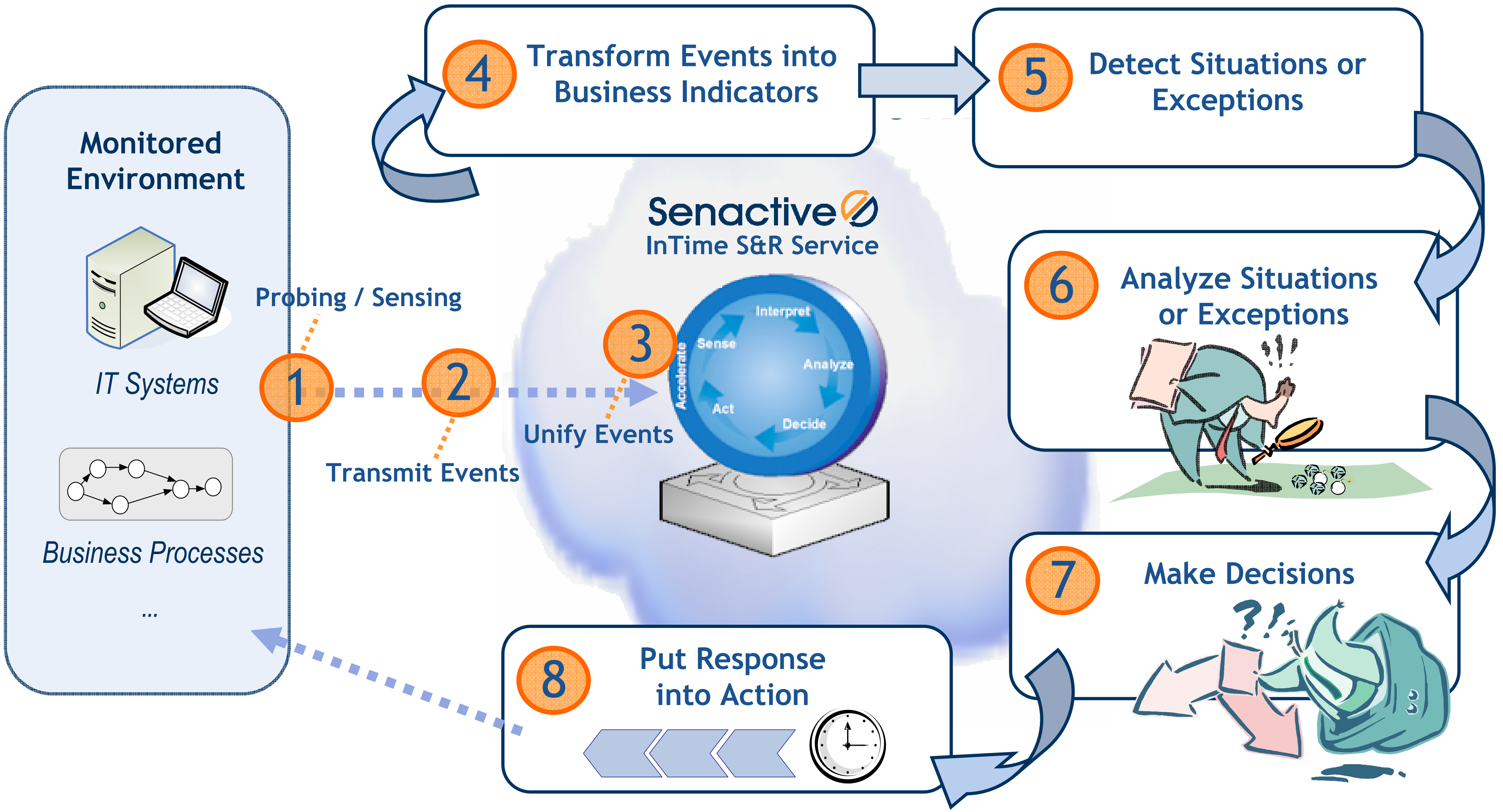
Event and Metric Persistence

Event Flow Management

Metric Evaluation

Notifications & Alerting

Senactive InTime Sense & Respond Loops



Research Topics for “Bakk.-Arbeiten”, “Diplomarbeiten” etc.

Research Topics

Business Process Management

- Web Service Based Auditing of BPEL4WS Processes (→ Web Services, BPEL4WS, Messaging)
- Scalable Messaging Infrastructure for Monitoring Business Processes (→ Messaging, MSMQ, MQSeries, JMS,…)
- Analysis Services with XMLA for Business Process Monitoring (→ OLAP, Data Mining, SQL Server 2005)

Event Management

- Event Models for Monitoring Business Processes (→ XML Schema for events)
- Event Simulation Model for Business Processes (→ Generating consistent XML events)
- Discovering Patterns in Event Streams, Event-Driven ECA Rules (→ Rule Engines, XML Rules)
- Correlation and Synchronization of Event Streams (→ XPath, Concurrency, Distributed Computing)

Data Management

- Data Management for Event Streams (→ Star Schemas, Data Warehousing, SQL Server 2005)
- Real-time Analytics & Data Stream Analysis with OLAP, Neuronal Network, Decision Trees, Clustering Techniques, Association Rules (→ Data Warehousing, Business Intelligence, SQL Server 2005)

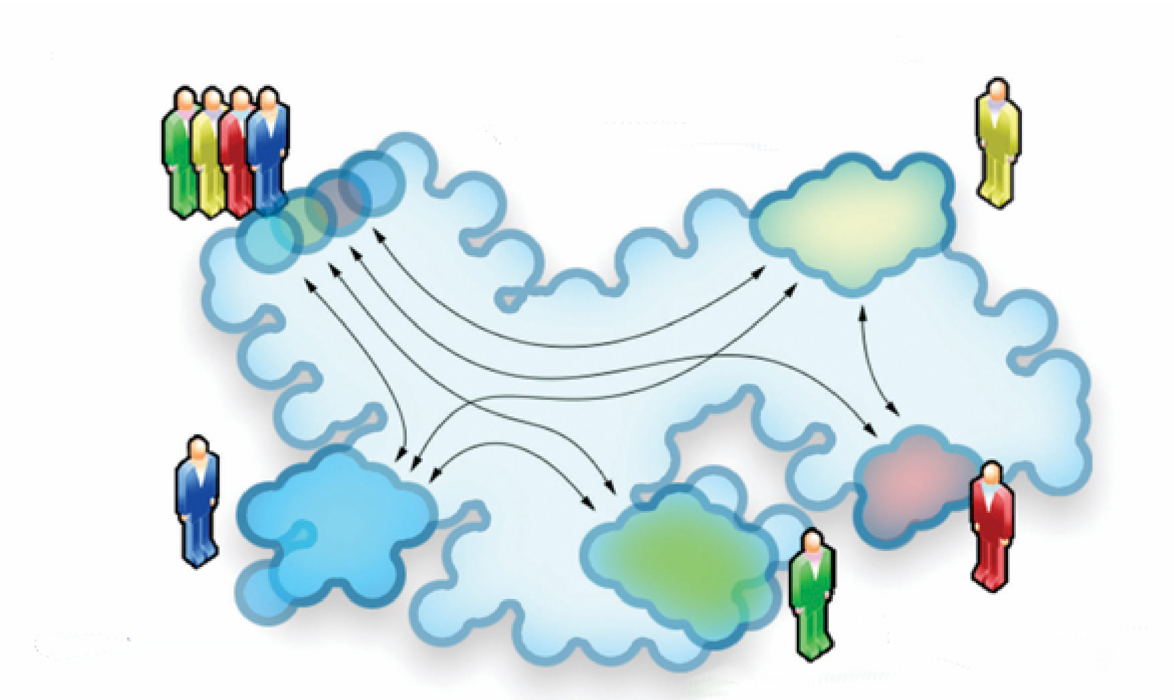
User Interfaces

- Visualization/Modeling of Sense & Respond Processes (data flows, control flows, dependencies) (→ Visualization and manipulation of Graphs, GUI design, Visual Studio 2005, C#)

Q&A

Building Standard-Based Business Processes – Part 2

with Web Services



Josef Schiefer

Vienna, Dezember 2004