# BPEL Research

Tuomas Piispanen

8.8.2006

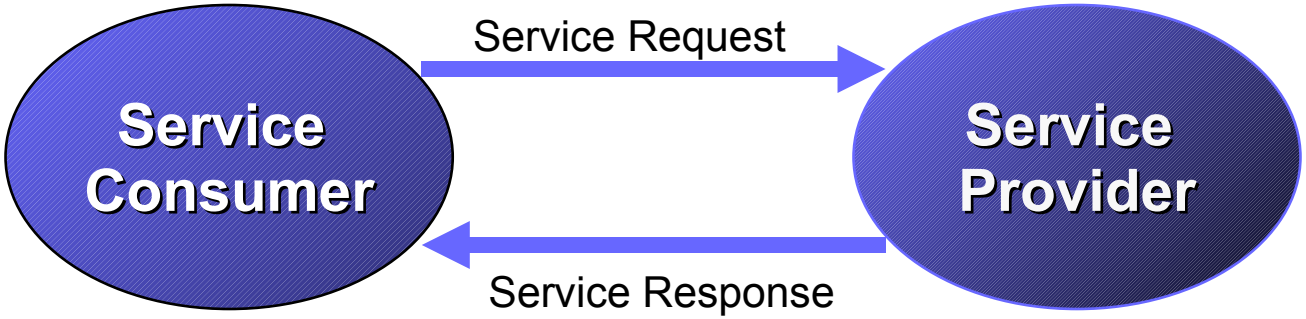
Comarch

# Presentation Outline

- SOA and Web Services
- Web Services Composition
- BPEL as WS Composition Language
- Best BPEL products and demo

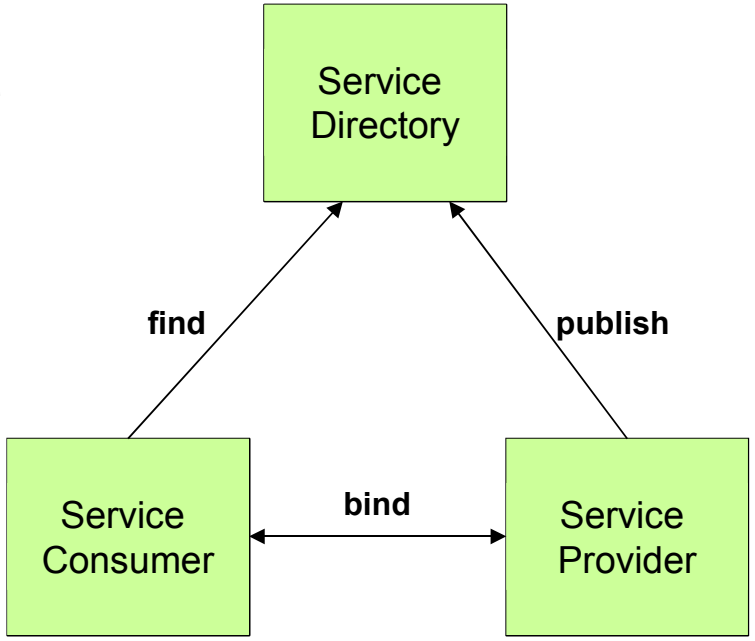
# What is a service?

“A unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software components”



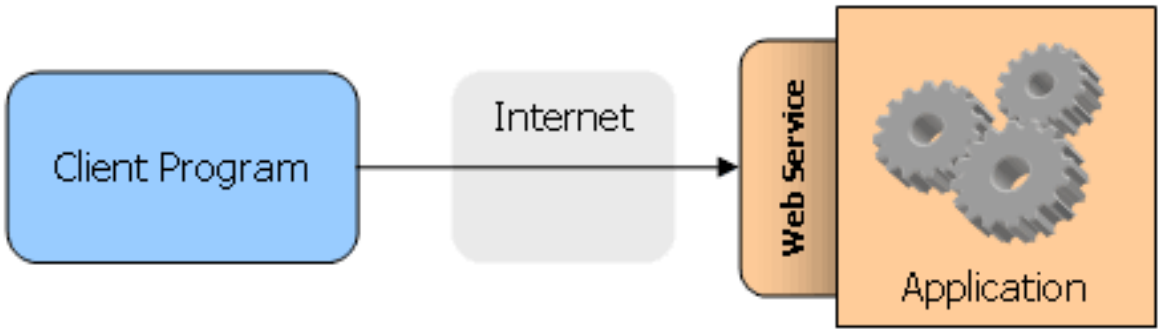
# SOA (Service-Oriented Architecture)

- Programming paradigm of the moment
- Architecture that represents software functionality as discoverable services on the network
- Vision of SOA
  - Many service providers
  - Many service consumers
  - And potentially many service directories
- Web Services are needed to realize the vision of SOA in practice
- Loose coupling of applications
- Communication through messages



# Web Services (1/4)

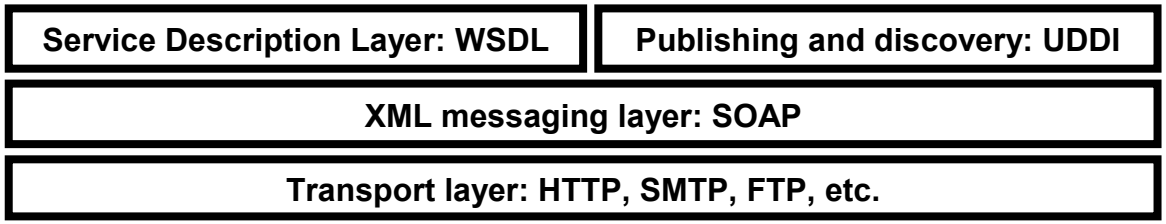
- Web Services are evolving, middleware platform that facilitate interactions between applications
- Applications are exposed to Internet/Intranet as services according to SOA architecture
- Easy invocation and location because of standard service descriptions and directories
- Aimed at solving the EAI and B2B integration problem
- The main enabler is the wide adoption of standards in the software industry



# Web Services (2/4)

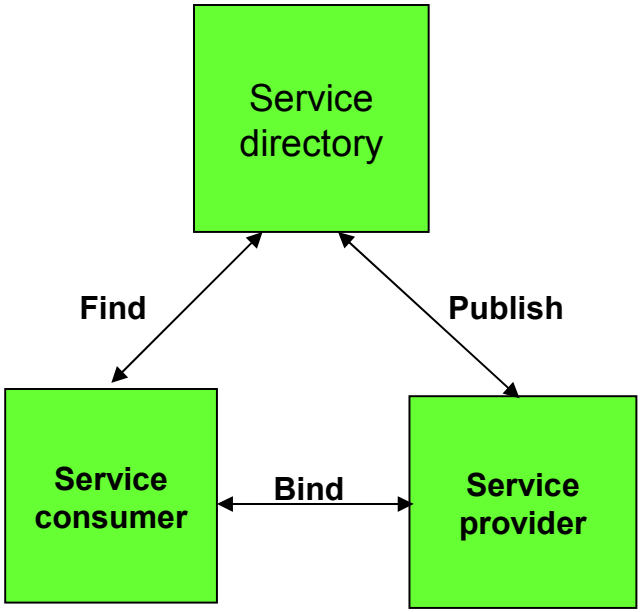
Basic Web Services technologies

- SOAP
  - Simple Object Access Protocol
  - Simple and platform-independent protocol to access Web Services
  - SOAP = XML + HTTP
  - Major advantage is the ability to penetrate firewalls
- WSDL
  - Web Services Description Language
  - Standard “interface” for Web Service
  - WSDL describes what WS can do, where it resides, and how to invoke it
- UDDI
  - Universal Description, Discovery and Integration
  - UDDI is a registry of Web Services (thus a database)
  - Basic idea is to connect producers and consumers of Web Services

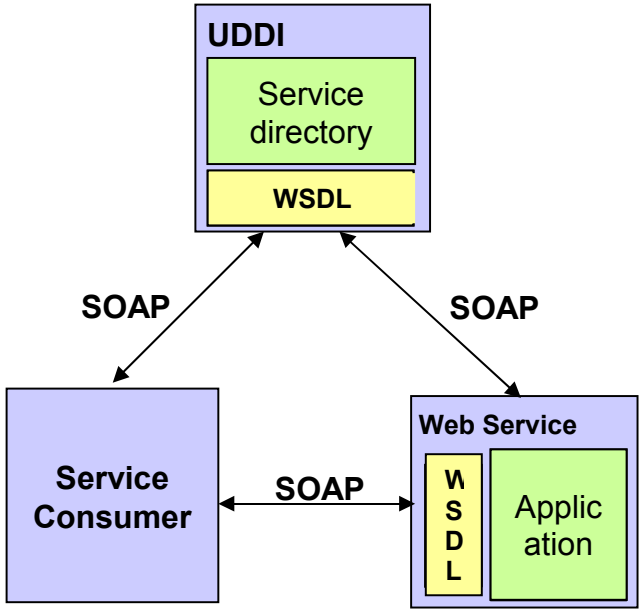


# Web Services (3/4)

SOA Architecture

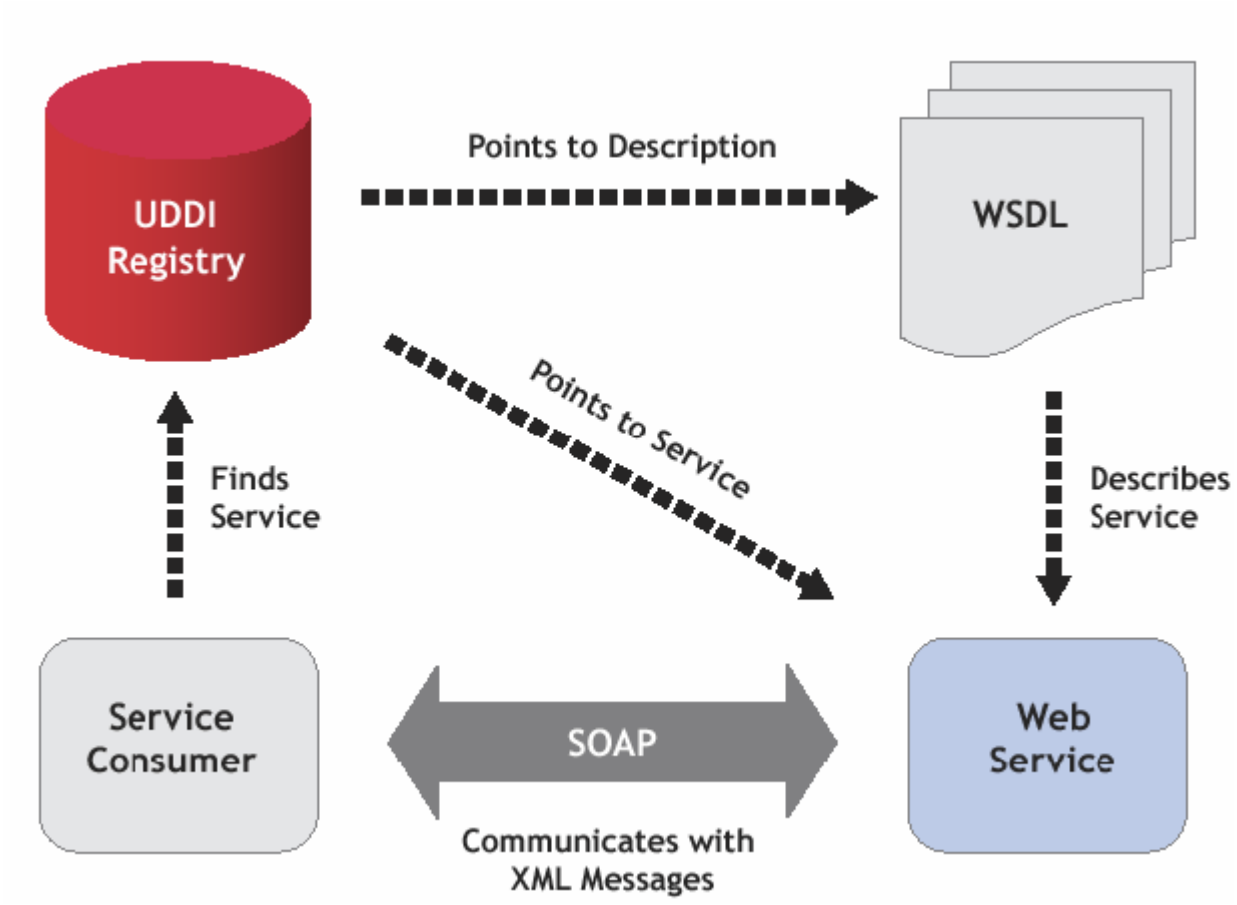



Implemented by Web Services



# Web Services (4/4)

SOA Architecture from the UDDI Perspective

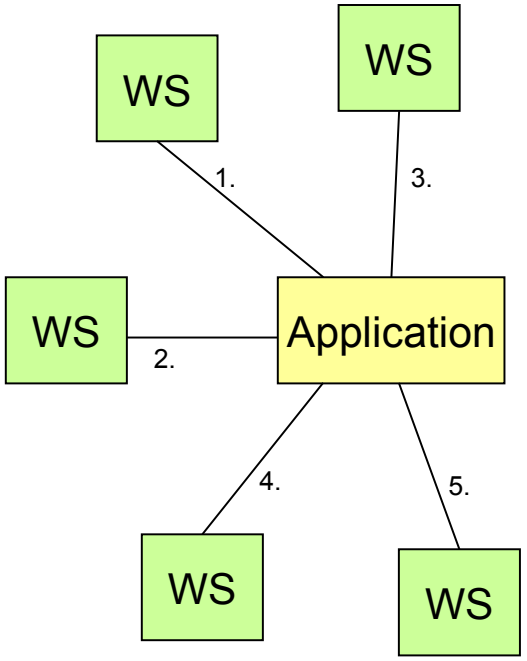


# Presentation Outline

- SOA and Web Services
- Web Services Composition
- BPEL as WS Composition Language
- Best BPEL products and demo

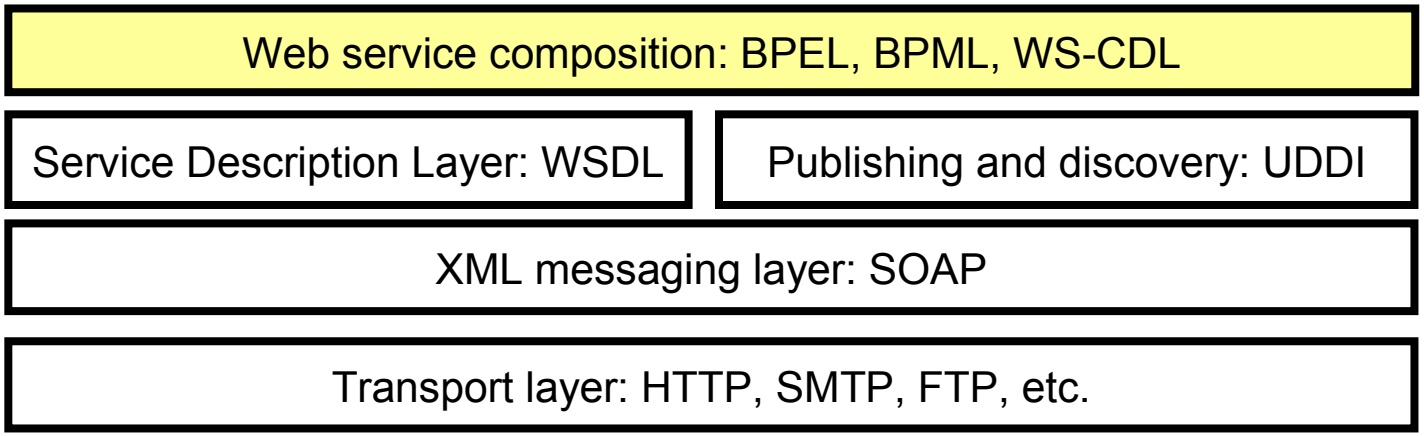
# Web Services Composition (1/3)

- Web Services composition is developing applications by composing Web Services as existing and reusable building blocks.
- It adds value to the collection of services, by combining them according to the requirements of the problem.
- WS composition is just a way to master complexity like using functions in conventional programming languages
- “Programming in the large” denotes to programming using enterprise systems as program components



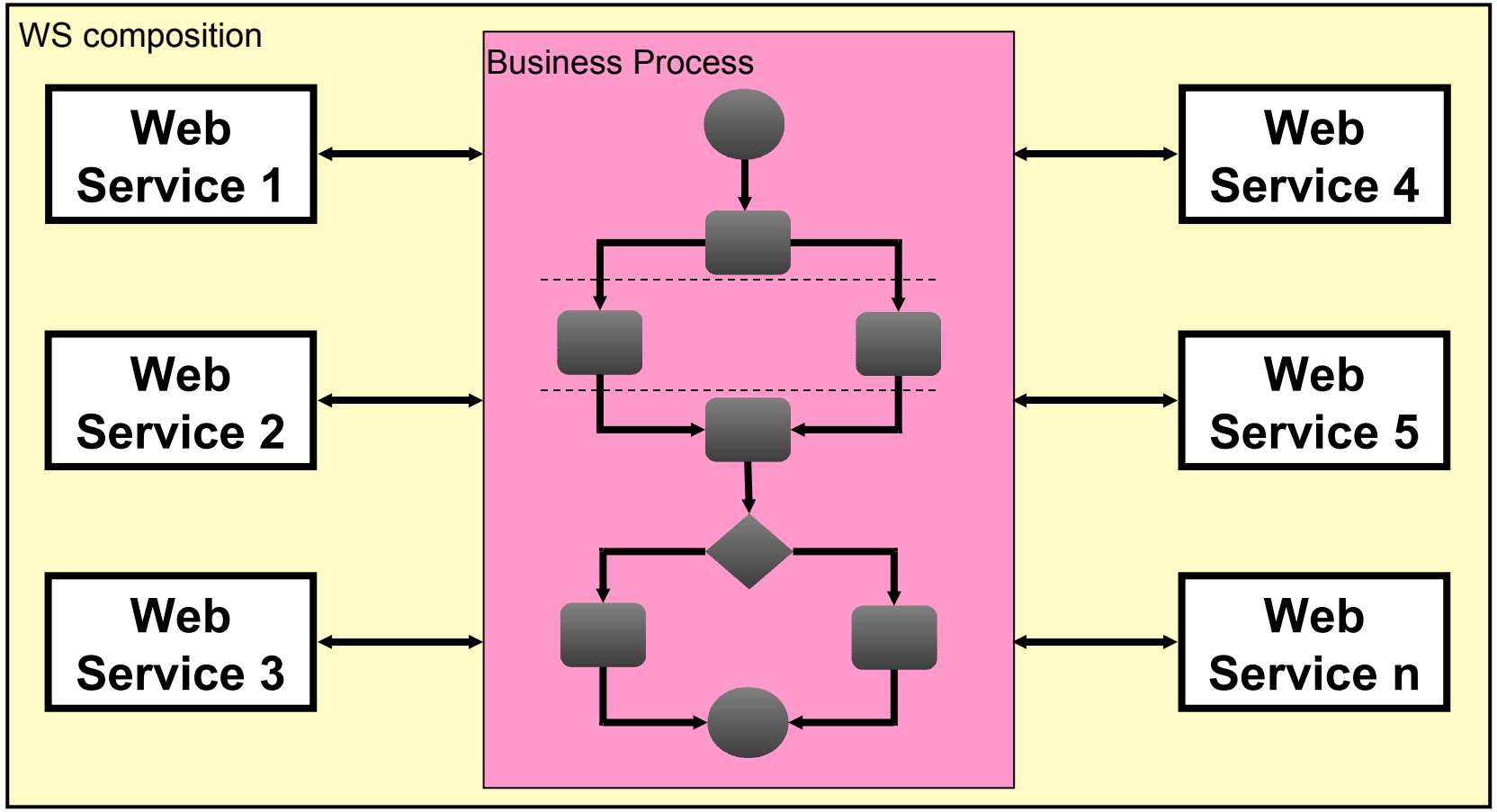
# Web Services Composition (2/3)

- New layer needed to the Web Services protocol stack
- WSDLs are the key components of WS composition because they enable a standard way to describe and access Web Services



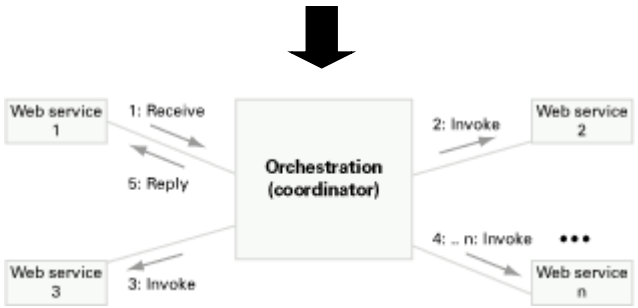
# Web Services Composition (3/3)

- The result of the composition is a work flow and called as a business process

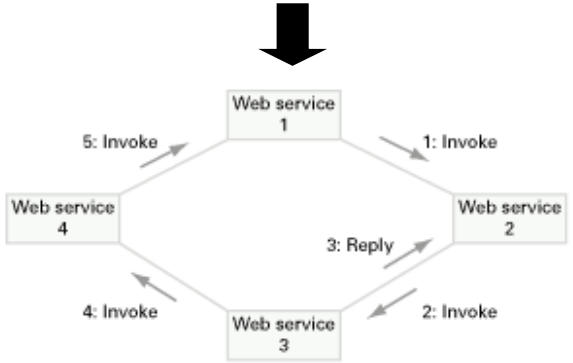


# Orchestration vs. Choreography

- **Orchestration**
  - Central coordinator is in control of the process
  - The involved Web Services do not know that they are involved into a composition



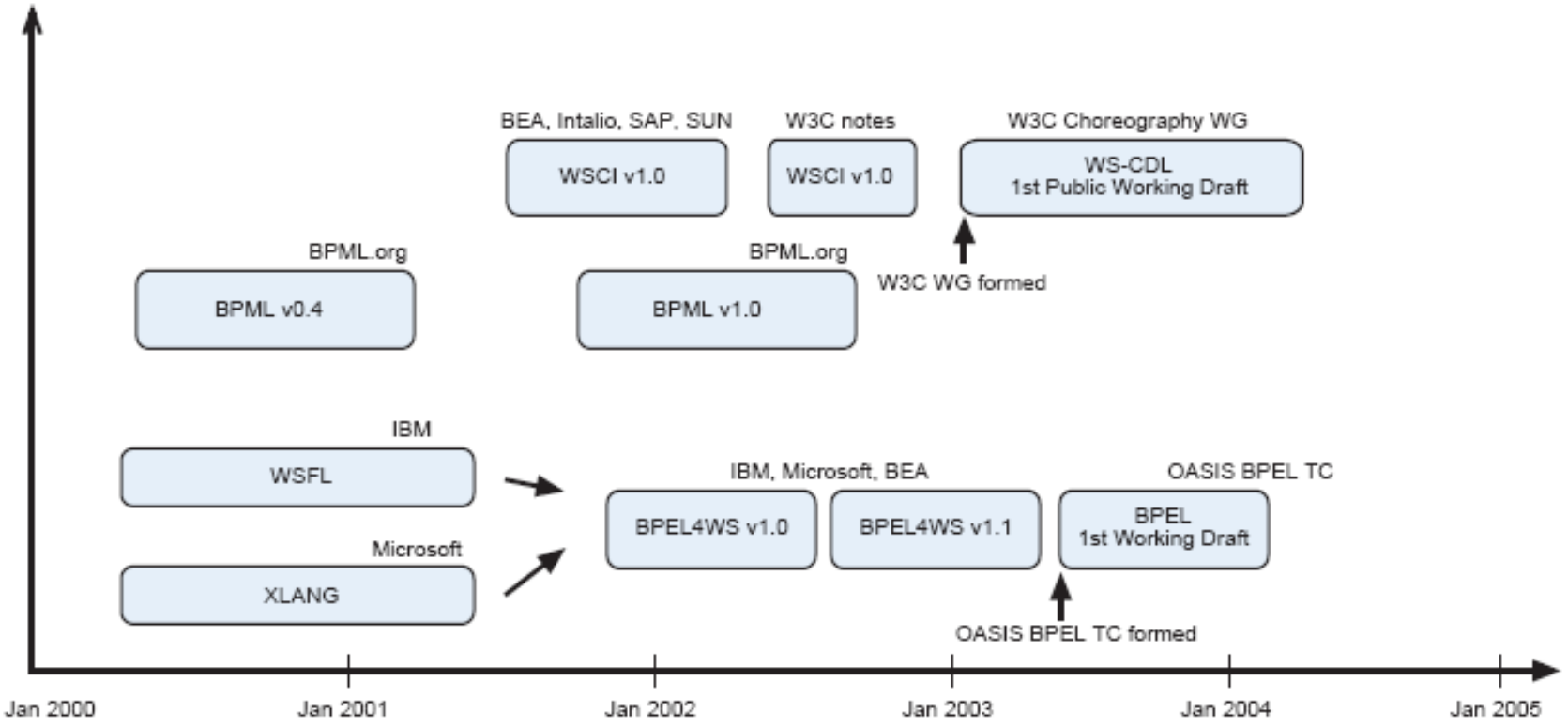

- **Choreography**
  - Does not rely on central coordinator
  - Defines the public message exchange between the Web Services
  - Each Web Service is aware of the composition and knows exactly when to execute its operations



# WS Composition Languages

- BPEL
  - Current de-facto orchestration language of the industry
- BPML
  - Another orchestration language but about to disappear because of BPEL, related to WSCI
- WS-CDL
  - The most current choreography language
- WSCI
  - Predecessor of the WS-CDL
- WSFL
  - Old IBM initiative for WS composition, superseded by BPEL
- XLANG
  - Former composition language of the MS Biztalk, superseded by BPEL

# Time-horizon of WS Composition Languages

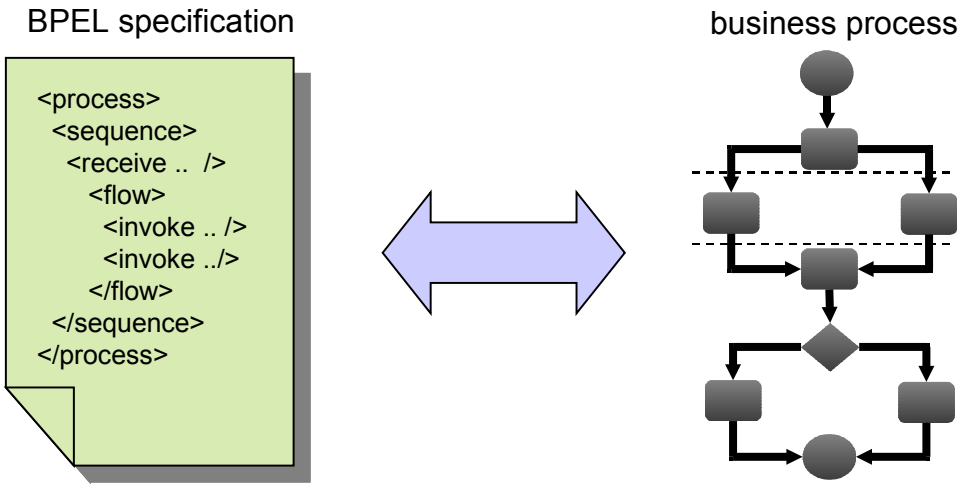


# Presentation Outline

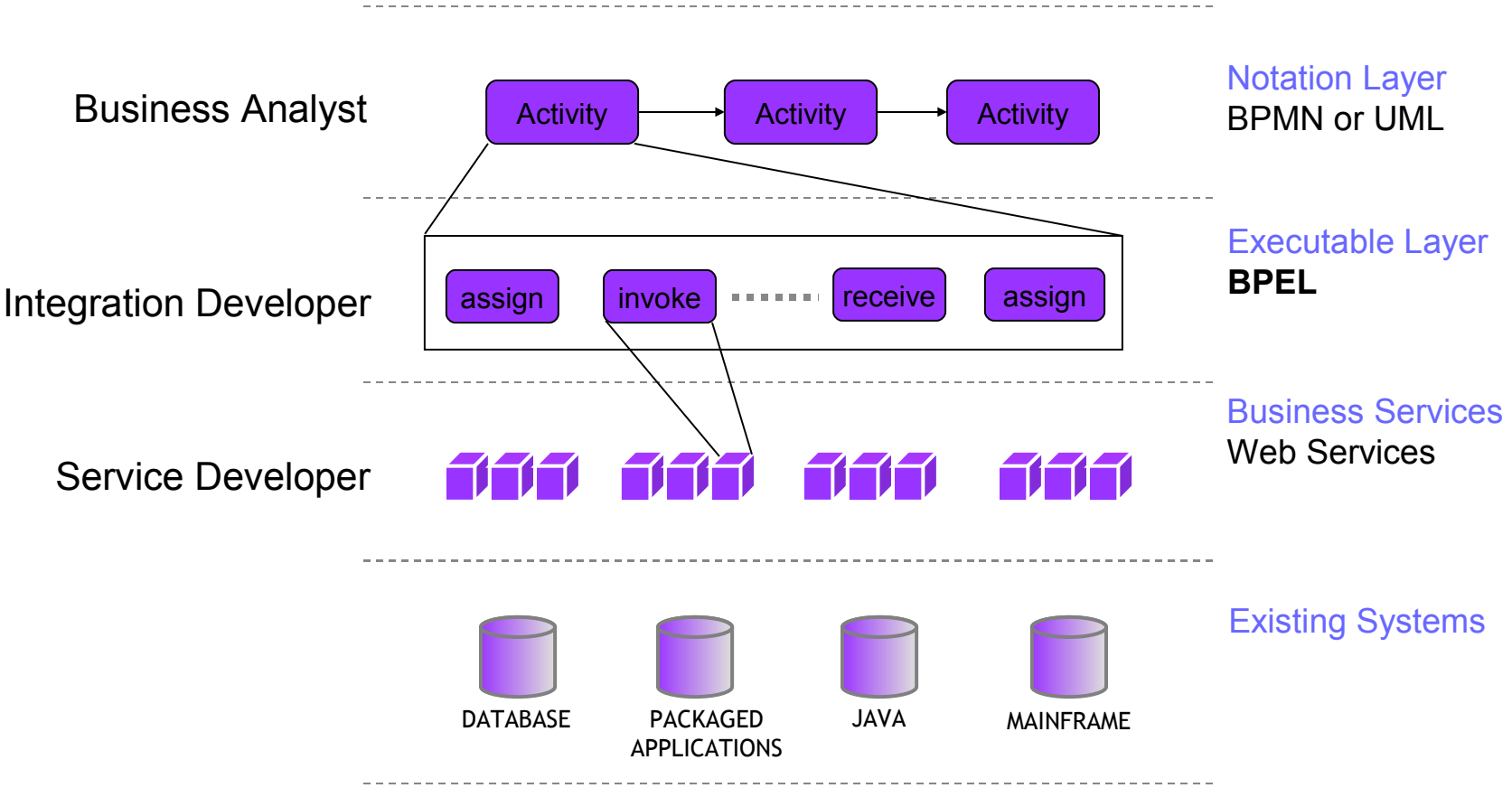
- SOA and Web Services
- Web Services Composition
- BPEL as WS Composition Language
- Best BPEL products and demo

# BPEL as WS Composition Language

- **B**usiness **P**rocess **E**xecution **L**anguage
- Current industry standard of how Web Services should be composed into business processes
- XML-based markup language that leverages XML Schema, XPath, XSLT, XQuery, WS-Security, WS-Addressing and WSIF
- Open standard under royalty-free terms

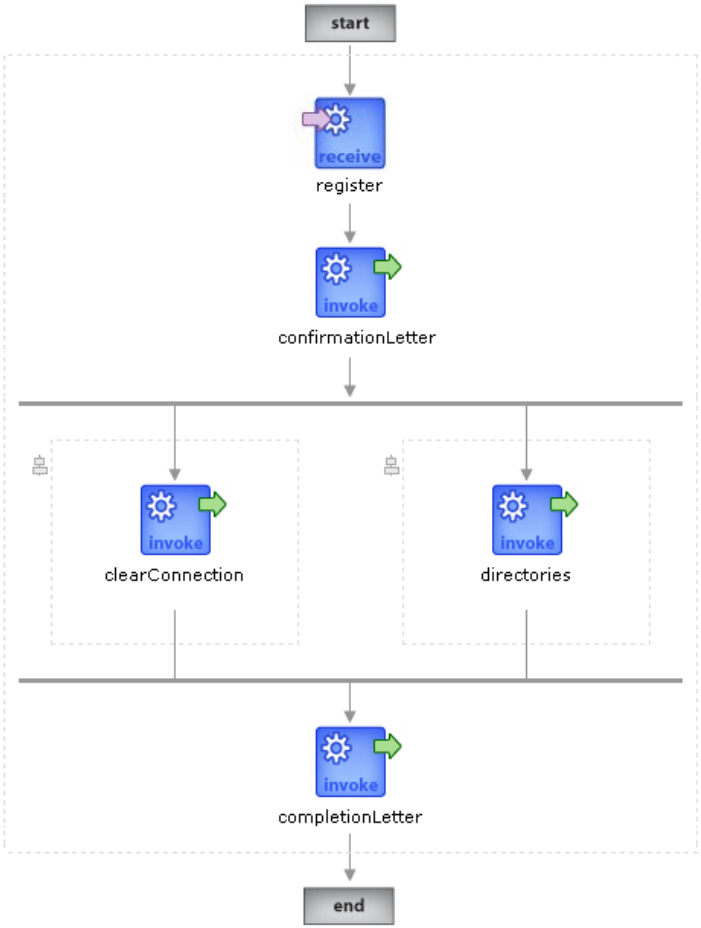


# Business Process Management with BPEL



# Motivating Example

Mail address clearance process




```
<sequence>

  <receive partnerLink="client" variable="input"
    operation="register"/>


  <invoke partnerLink="letterCenter"
    operation="confirmation" inputVariable="input"/>

  <flow>

    <invoke partnerLink="clearingCenter"
      operation="clearConnection" inputVariable="input"/>

    <invoke partnerLink="directories"
      operation="register" inputVariable="input"/>

  </flow>

  <invoke partnerLink="letterCenter"
    operation="completion" inputVariable="input"/>

</sequence>
```

# BPEL Basic Constructs

- Activities
  - The basic building blocks of BPEL processes
  - Activities are like steps in the process that we have to take to walk the process path from start to finish
  - <receive>, <invoke>, <reply>, <assign>, <sequence>, <flow>, ..
- Partner Links
  - Partner links are the entry and exit points of a BPEL process
  - All Web Services involved into the process are modeled as partner links
  - Partner links can also be links to clients which invoke the BPEL process
- Variables
  - Variables are used to hold the state of the process by storing messages
  - The variables can be defined in the terms of WSDL message types, XML Schema types, or XML Schema elements
  - All variables have to be declared in a certain scope which can be either global or local
- Namespaces
  - The namespaces enable the differentiation of element and attribute names if same names are used in many contexts

```
<sequence>

  <receive partnerLink="client" variable="input"
    operation="ns1:register"/>


  <invoke partnerLink="letterCenter"
    operation="ns2:confirmation"
    inputVariable="input"/>

  <flow>

    <invoke partnerLink="clearingCenter"
     operation="ns3:clearConnection"
    inputVariable="input"/>

    <invoke partnerLink="directories"
     operation="ns4:register"
    inputVariable="input"/>

  </flow>

  <invoke partnerLink="letterCenter"
    operation="ns5:completion"
    inputVariable="input"/>

</sequence>
```

# BPEL Features

- Scopes
  - Scope = **{ }** structure in conventional programming languages
  - defines a nested activity with its own associated variables, fault handlers, and compensation handler
- Transactions with compensation
  - In BPEL, a set of activities can be grouped in a single transaction with <scope>
  - Compensating actions are defined in the scope's exception handler
- Exception Handling
  - Exception handling works in BPEL in the same way as in Java
  - A fault handler is always defined for some scope
  - <faulthandler>, <throw>, <rethrow>, <catch>, <catchAll>, ..
- Correlation
  - With correlation we can make sure that we are always talking to the same instance of a service
  - WS-addressing is used to set tokens to headers that act like cookies in web applications
  - Optional feature and not needed with synchronous invocations
  - <correlation>, <correlationSets>
- Event Handlers
  - Process can react to two kind of events: message events or alarm events
  - <pick>, <onAlarm>, <onMessage>
- Concurrency and non-determinism
  - Concurrent activities can be defined in BPEL using the <flow> activity

# BPEL Advantages

- Widely approved standard
- Platform-independent
- Support for business process characteristics
- Well-suited for EAI and B2B integration
- Simple and easy to learn

BPEL vs. Java

- Everything that can be done with BPEL can be done with Java too
- BPEL is not as powerful programming language as Java but it is better suited for business process definition
- Disadvantages of Java
  - Platform-dependent
  - No support for business process characteristics

# BPEL Activities

- Basic activities
  - Simplest form of constructs that cannot contain other activities

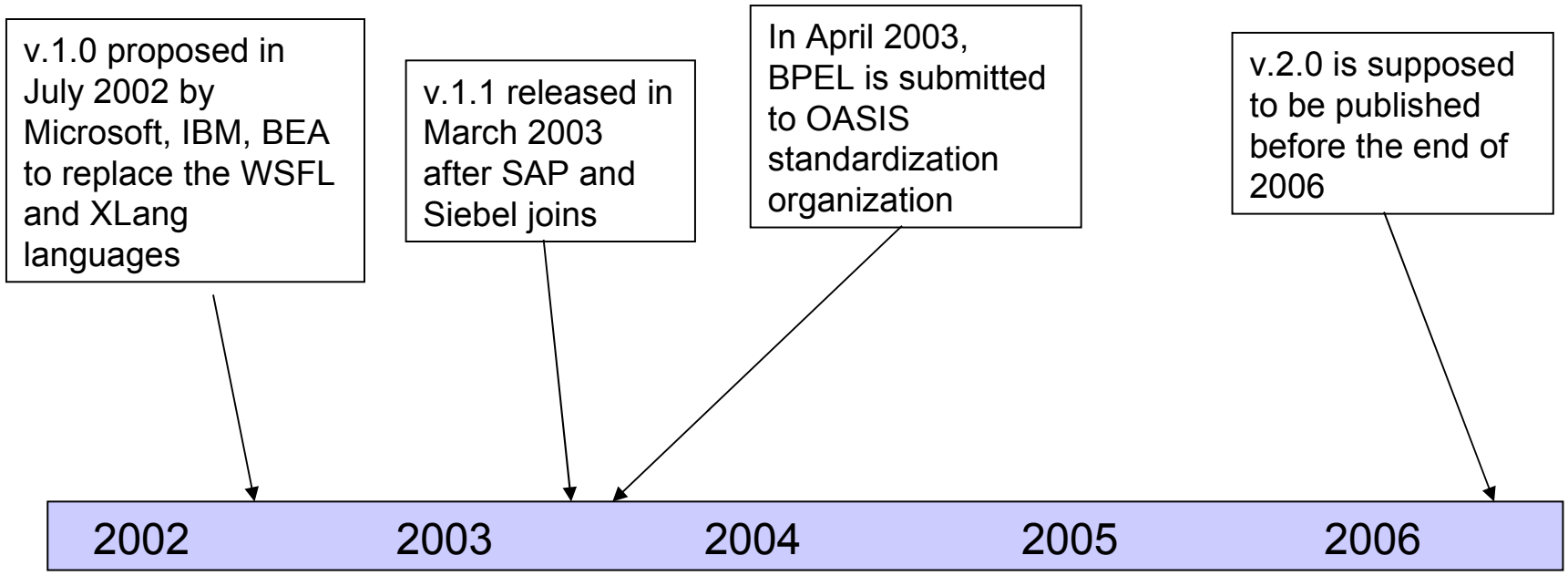
<invoke>
<receive>
<assign>
<reply>
<throw>
<terminate>
<wait>
..

- Structured activities
  - Offer a way to structure a BPEL process
  - Contain other activities (basic or structured)

<sequence>
<switch>
<pick>
<flow>
<link>
<while>
<scope>

# Evolution of BPEL

- BPEL is standardized by OASIS which is a coalition of major software vendors
- OASIS = Organization for the Advancement of Structured Information Standards



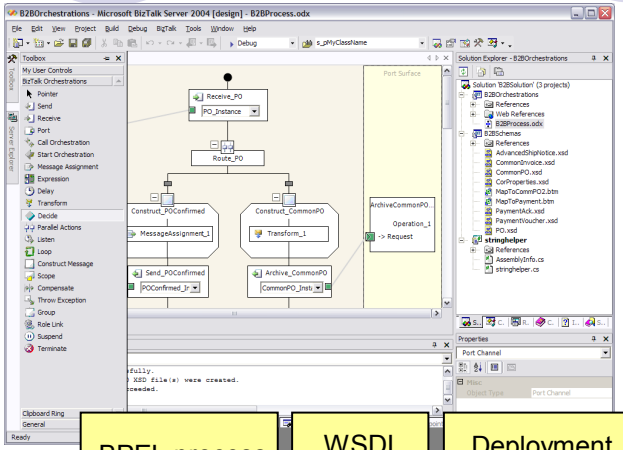
# Executable vs. Abstract Processes in BPEL

- **Executable process**
  - contains the exact details of execution
  - Can be executed by a BPEL engine
- **Abstract process**
  - Does not include internal details of the process and are nor executable
  - Abstract processes are used to describe the business process to another party who is willing to implement it
  - Abstract process are contracts of message sequencing between the interacting services

# BPEL in Practice


Business Analyst

**BPEL Editor**

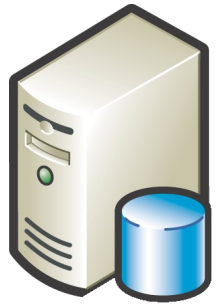



BPEL process Specification (.BPEL)

.WSDL

Deployment descriptor (.XML)

Web Services



```
<process>
  <sequence>
    <receive
      … />
    <invoke
      … />
  </sequence>
</process>
```

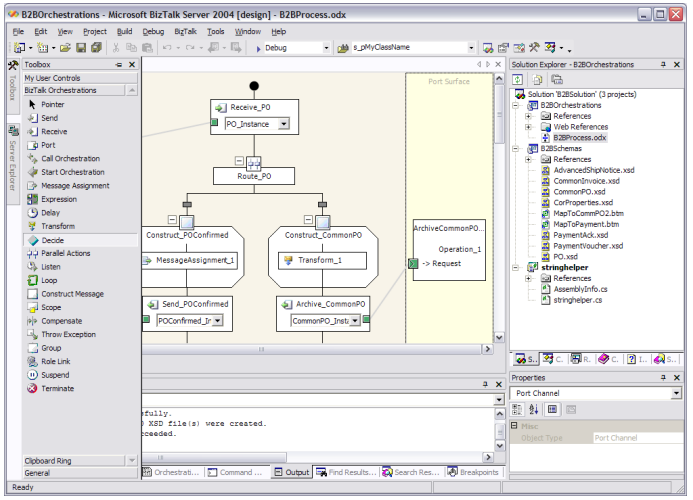
Is deployed to

**BPEL Engine**

# BPEL Editors

(BPEL designer, BPEL design tool, ..)

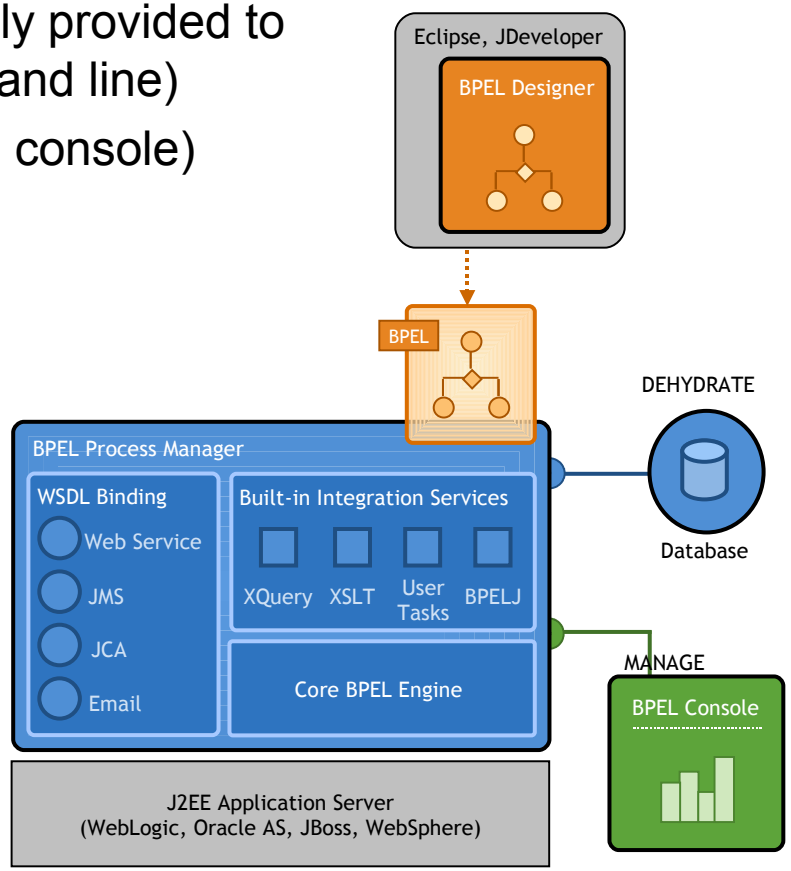
- Aimed not only at software engineers but also for non-technical business people unaware of programming languages
- Designing processes both in graphical and source code views
- With most editors there is no need to touch the source code
- Editors provide help tools to
  - deploy processes to BPEL engine
  - validate BPEL code
  - import Web Services
  - map data from variable to variable
  - debug processes



# BPEL Engines

(BPEL server, orchestration server, process manager, orchestrator, execution engine, business integration se

- Implemented as a Web app in Application Server (usually J2EE)
- Is used to execute the deployed BPEL specifications
- Web based administration console is usually provided to administer the engine (in addition to command line)
- Some common features of the engine (and console)
  - Deployment of processes
  - Process debugging
  - Process dehydration
    = storing long running process to database
  - Process monitoring
  - Version management
  - Exception management and error reporting




**Architecture of Oracle BPEL Process Manager**

# Presentation Outline

- SOA and Web Services
- Web Services Composition
- BPEL as WS Composition Language
- Best BPEL products and demo

# Tested BPEL Products and Recommendations

- BPEL Editors
  - Oracle BPEL Designer for JDeveloper ***RECOMMENDED***
  - Oracle BPEL Designer for Eclipse
  - ActiveBPEL Designer ***RECOMMENDED***
  - IBM BPWS4J Editor
  - Eclipse BPEL Designer
- BPEL Engines
  - Oracle BPEL Process Manager ***RECOMMENDED***
  - ActiveBPEL Engine ***RECOMMENDED***
  - IBM BPWS4J Engine
  - Bexee
  - Twister

# BPEL Engines

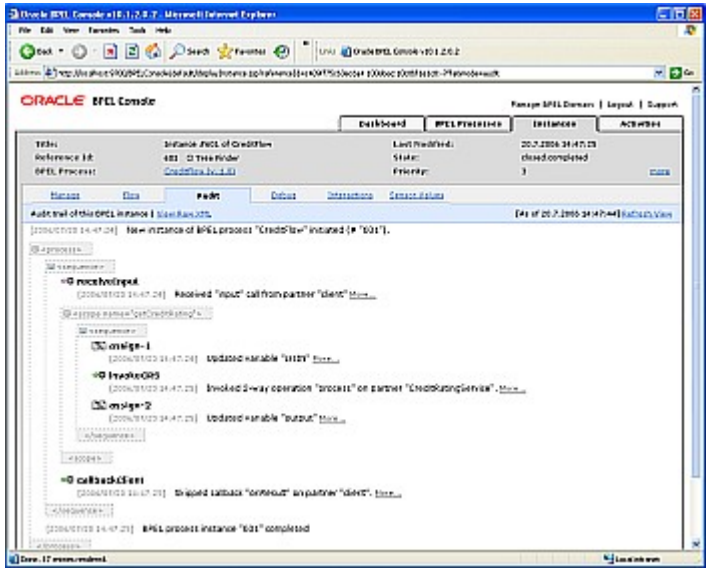
| | **Oracle BPEL Process Manager** | **ActiveBPEL Engine** | **ActiveBPEL Enterprise** | **IBM BPWS4J Engine** | **Bexee** | **IBM WebSphere Business Integration Server Foundation** | **Microsoft Biztalk 2006** |
|---|---|---|---|---|---|---|---|
| Vendor | Oracle | - | Active Endpoints | IBM | - | IBM | Microsoft |
| Platform | J2EE | J2EE | J2EE | J2EE | - | J2EE | .NET |
| Current status | In development | In development | In development | Discontinued | Discontinued | In development | In development |
| Latest version | 10.1.2 (2006) | 2.0 (2/2006) | ? | 2.1 (2004) | 0.1 (2004) | 4.4 (4/2005) | 2006 |
| Type of license | Commercial | Open Source | Commercial | Commercial | Open Source | Commercial | Commercial |
| Price | $7500-30000 / CPU | Free | Unknown | unknown | Free | $50000 / CPU | $25000 / CPU |
| BPEL version supported | 1.1 | 1.1 | 1.1 | 1.1 | - | 1.1 | ? |

# BPEL Editors

| | Oracle BPEL Designer for Eclipse | Oracle JDeveloper BPEL Designer | IBM BPWS4J Editor | ActiveBPEL Designer | Eclipse BPEL Designer |
|---|---|---|---|---|---|
| Vendor | Oracle | Oracle | Active Endpoints | IBM | - |
| Type of software | Eclipse plug-in | JDeveloper plug-in | Eclipse plug-in | Eclipse plug-in | Eclipse plug-in |
| Current status | Discontinued | In development | In development | In development | Discontinued |
| Latest version | 2.2 (2004) | 10.1.2.0.2 (2005) | 2.1 (5/2004) | 2.0 (2/2005) | 0.01 (2006) |
| Type of license | Freeware | Freeware | Commercial | Freeware | Open Source |
| BPEL version supported | 1.1 | 1.1 | 1.1 | 1.1 | - |
| Deployment of processes | Deployment to Oracle BPEL PM only | Deployment to Oracle BPEL PM only | Not possible from editor | Deployment to all different ActiveBPEL engines | - |

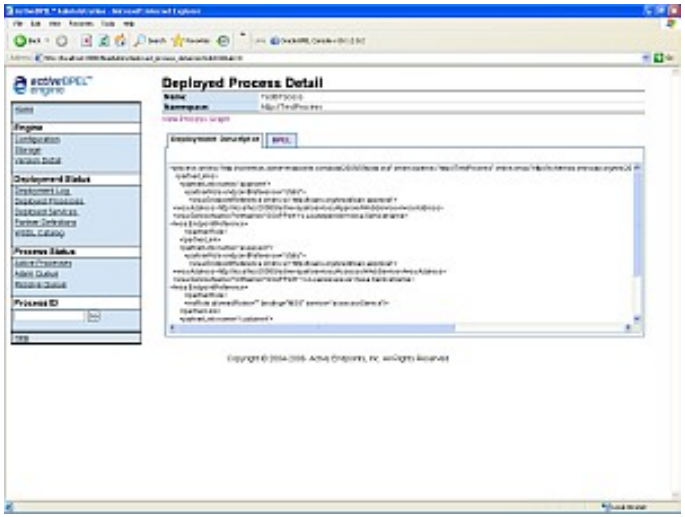
# Oracle BPEL Process Manager

- Result of the acquisition of Collaxa Inc. (2004) – who offered the leading BPEL implementation in the market
- High-end enterprise-level platform for designing, deploying and managing BPEL processes
- The most comprehensive BPEL platform currently on the market
- Commercial product which bundles many programs (Oracle BPEL Designer, Application Server, Oracle Database, JDeveloper IDE)
  - Price is lower if the buyer has existing Oracle products
- Advantages
  - Both BPEL Designers (Eclipse and JDeveloper) are very comfortable to use
  - Scalable and reliable engine
  - Easy one-click deployment from the Designer
  - Excellent process management, debugging and monitoring features through administration console
  - Easy installation to several J2EE app servers
  - Support and learning material
- Disadvantages
  - Cost - rather pricey
  - Some small bugs with deployment



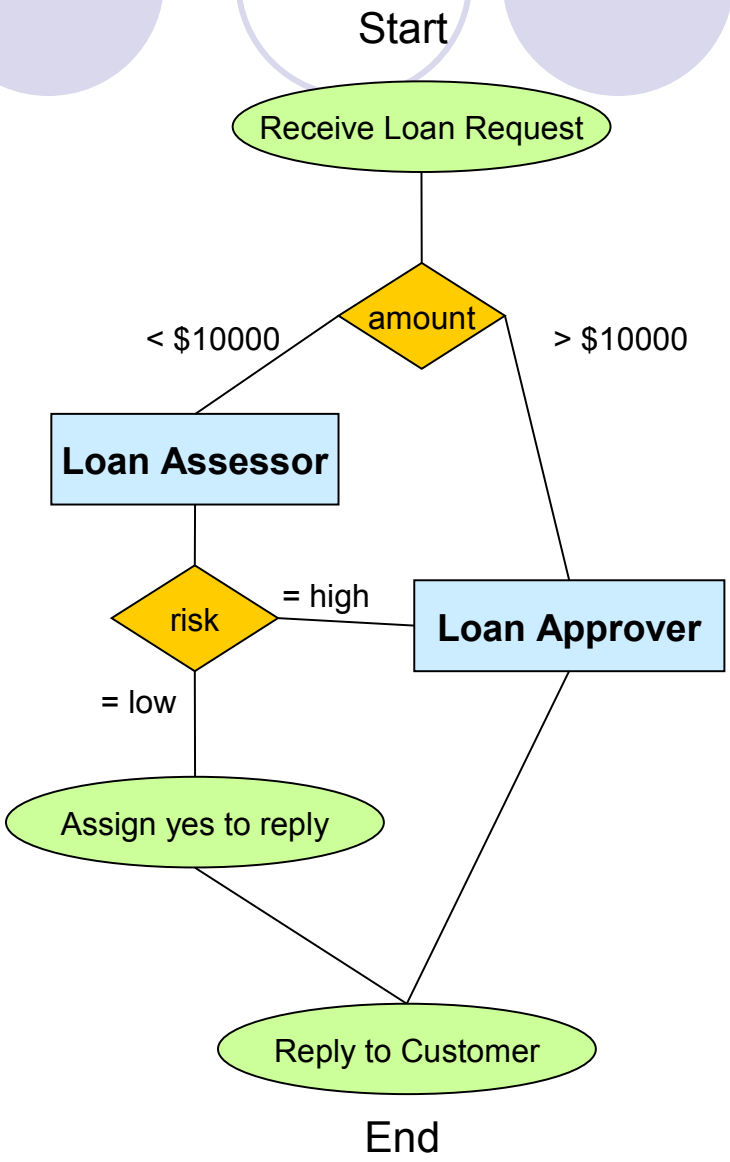
# ActiveBPEL Engine & Designer

- Open Source project in active development by the ActiveBPEL community
- Originally developed by Active Endpoints inc. but who decided to publish the core technology of their commercial engines as Open Source in 2004
- Basic enterprise-level BPEL engine for designing, deploying and managing BPEL processes
- Currently the best combination of free engine and editor
- Advantages
  - Number of features
  - Easy deployment to ActiveBPEL engines
  - Cost (free)
  - Easy deployment
  - Support and learning material
  - Very easy debugging of processes from the ActiveBPEL Designer
- Disadvantages
  - Can be difficult to learn
  - Some problems occurred in installation



# Loan Approval Demo

- The loan approval process starts by receiving a customer request for a loan amount.
- If the request is for less than $10,000, the assessor's Web service is invoked to assess the risk
- If the customer is low risk, the request is approved. Otherwise, the request is sent to the approver for review.
- If the amount is for $10,000 or more, the request is always sent to the approver for review.
- The customer can receive acceptance from the assessor or an accept/reject reply from the approver.



# Questions?