



WS-BPEL 2.0

Werden die Erwartungen nun erfüllt?

Torsten Winterberg
Sven Bernhardt
OPITZ CONSULTING GmbH

Agenda

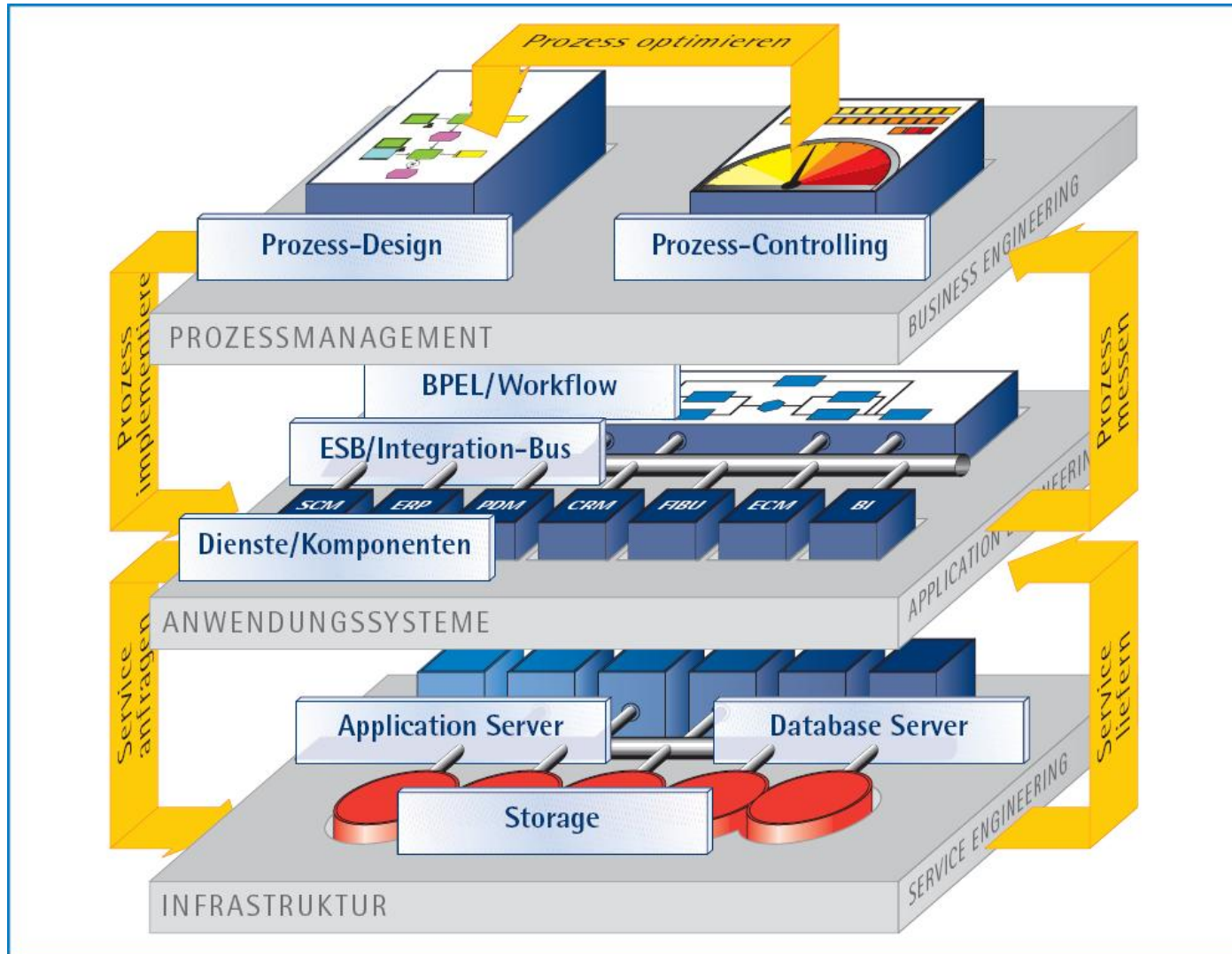
- Warum BPEL? Was ist BPEL?
- Architekturkonzepte
- Neues in BPEL 2.0
- Menschliche Interaktion / Human Workflow
- Tools inkl. Live-Demos
- Migration von BPEL 1.1 auf BPEL 2.0
- Bewertung des neuen Standards



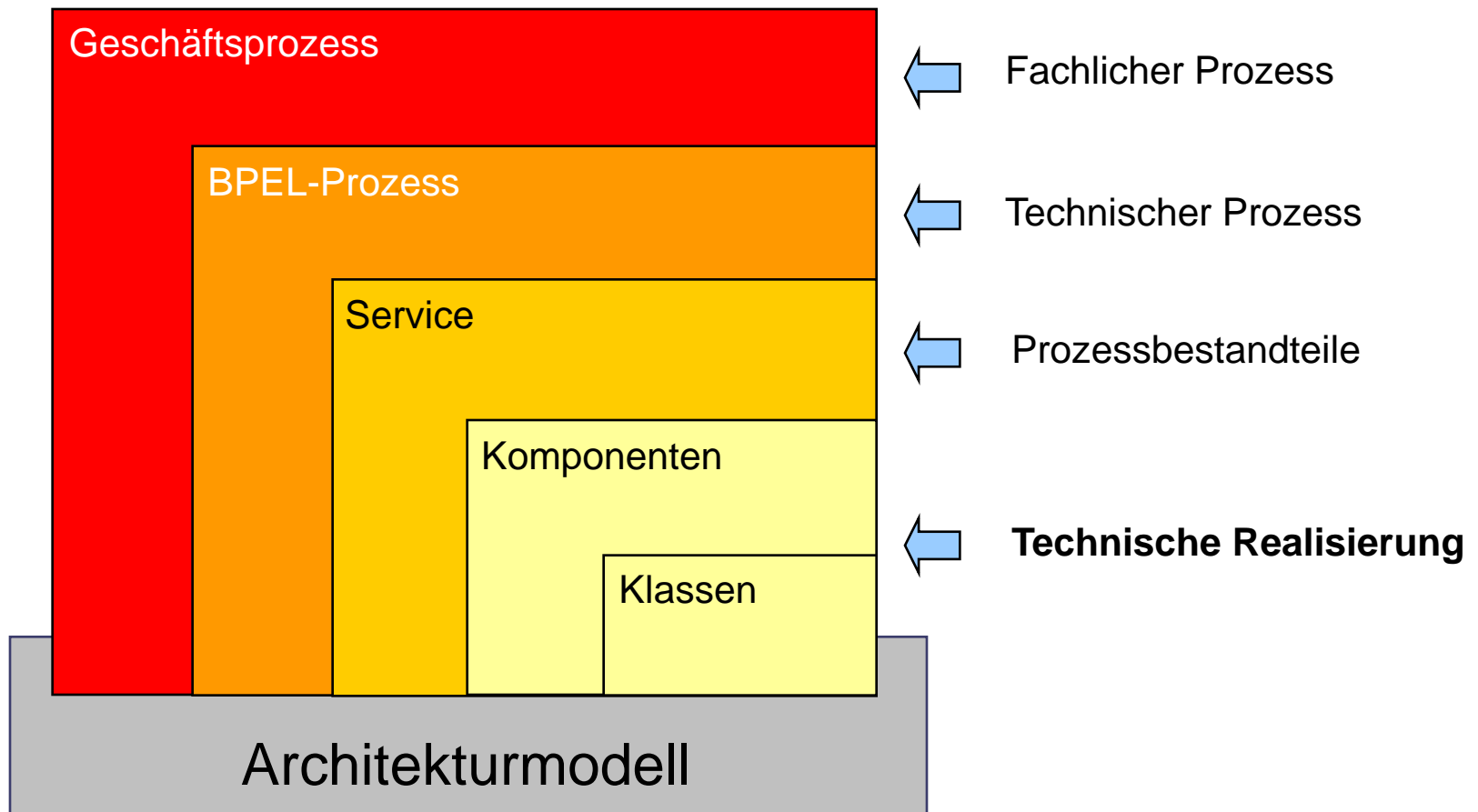
1

Warum BPEL, Was ist BPEL, Architekturkonzepte

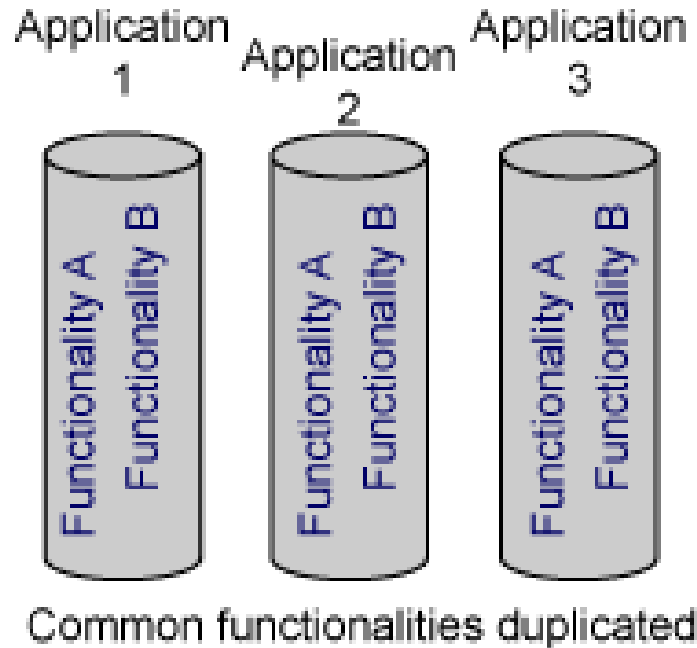




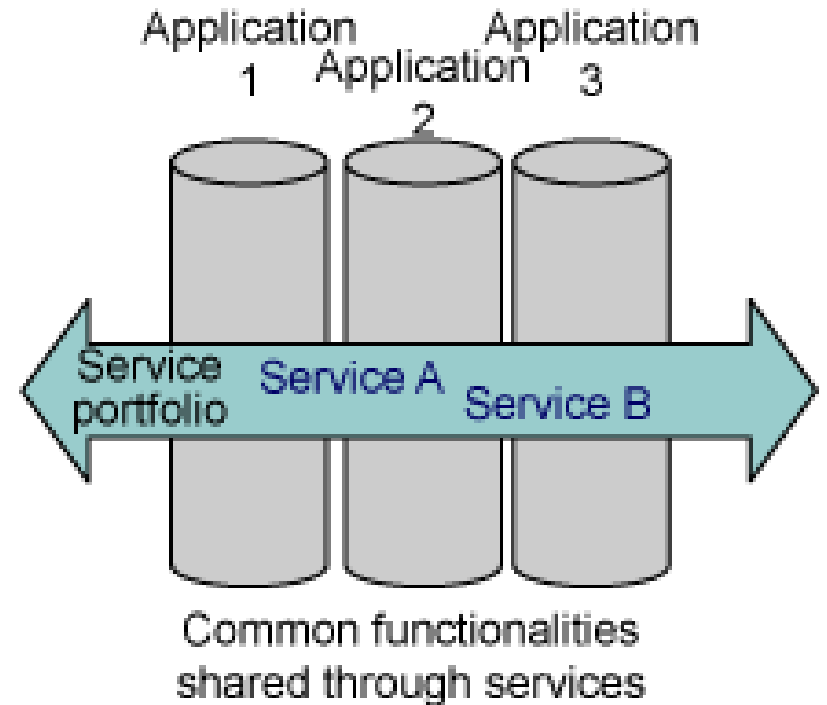
Unterschiedliche Design-Ebenen



The silo approach



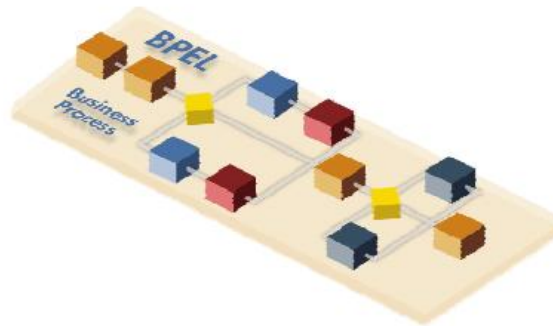
The SOA approach



Was ist BPEL überhaupt?

**BPEL is the Web Services
Orchestration standard from OASIS*:**

**An XML-based grammar for describing the
logic to orchestrate the interaction between
Web Services in a business process**



*Organization for the Advancement of Structured Information Standards
(<http://www.oasis-open.org>)

BPEL Historical Timeline

Dec 2000

Microsoft publishes XLANG

March 2001

IBM publishes WSFL

July 2002

IBM, Microsoft and BEA
converge WSFL & XLANG
into BPEL4WS 1.0

March 2003

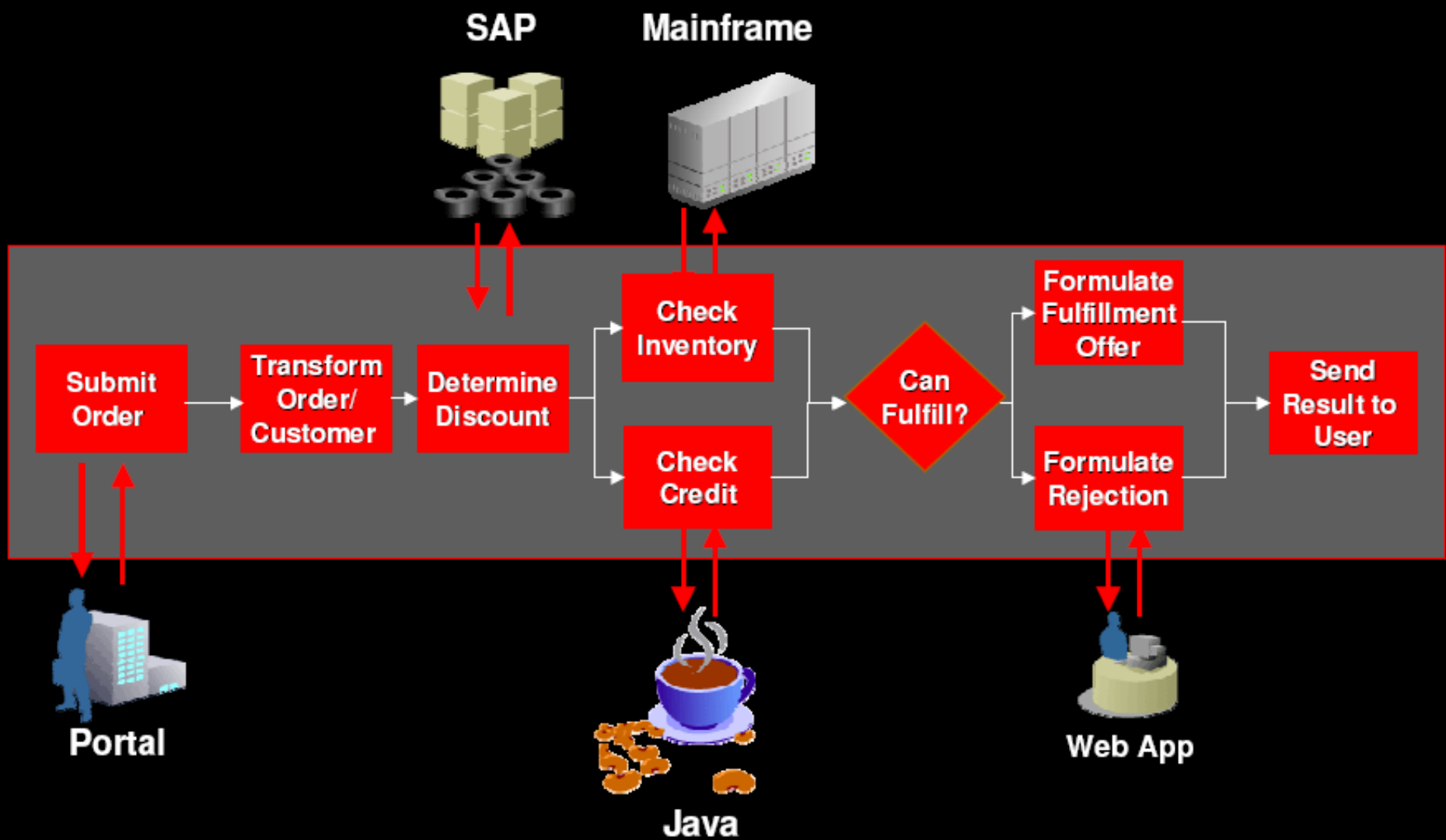
BPEL4WS is submitted to
OASIS

May 2003

OASIS publishes BPEL4WS
1.1

April 2007

WS-BPEL 2.0 released



BPEL ist gut für langlaufende Prozesse

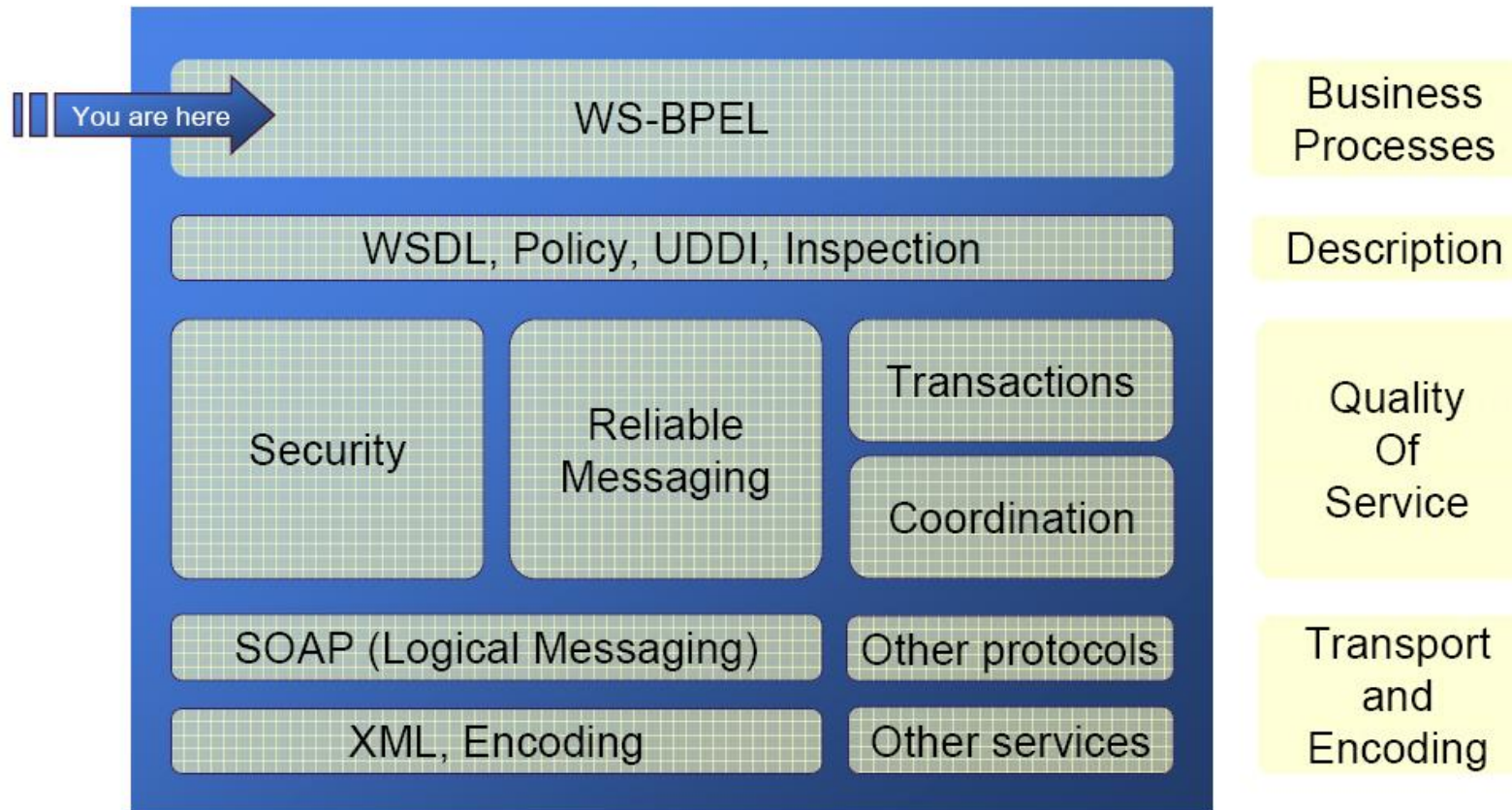
- WSDL-definierte Services haben ein zustandsloses Interaktionsmodell:
 - Messages werden über synchrone Aufrufe ausgetauscht
 - Asynchrone Aufrufe sind unkorreliert
- „Real-World“ Business Prozesse erfordern ein robusteres Interaktionsmodell:
 - Messages werden bidirektional ausgetauscht, in peer-to-peer Konversationen, die Minuten, Stunden oder Tage laufen können.
- BPEL erlaubt es, zustandsbehaftete, lang-laufende Interaktionen durchzuführen.

Beziehung von BPEL zu WSDL

- BPEL basiert auf und erweitert das WSDL Service Modell
- WSDL definiert die spez. erlaubten Operationen
- BPEL definiert, wie WSDL Operationen orchestriert werden, um einem Business Prozess zu genügen
- BPEL definiert weiterhin Erweiterungen zu WSDL, für den Support von langlaufenden, asynchronen Business Prozessen

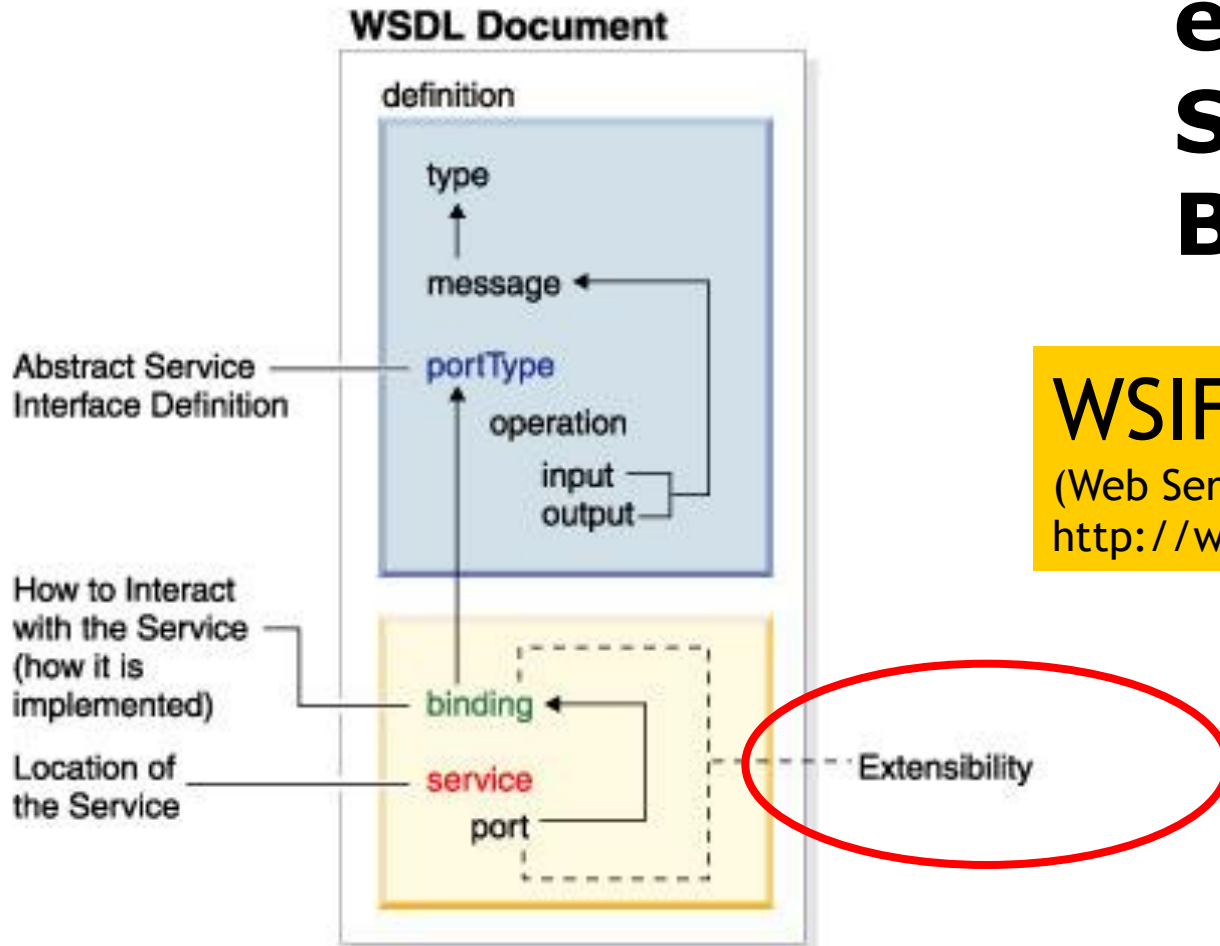


WS-BPEL im WS-* Stack



Der erweiterte Service-Begriff

WSIF !
 (Web Service Invocation Framework)
<http://ws.apache.org/wsif>



Composite Application

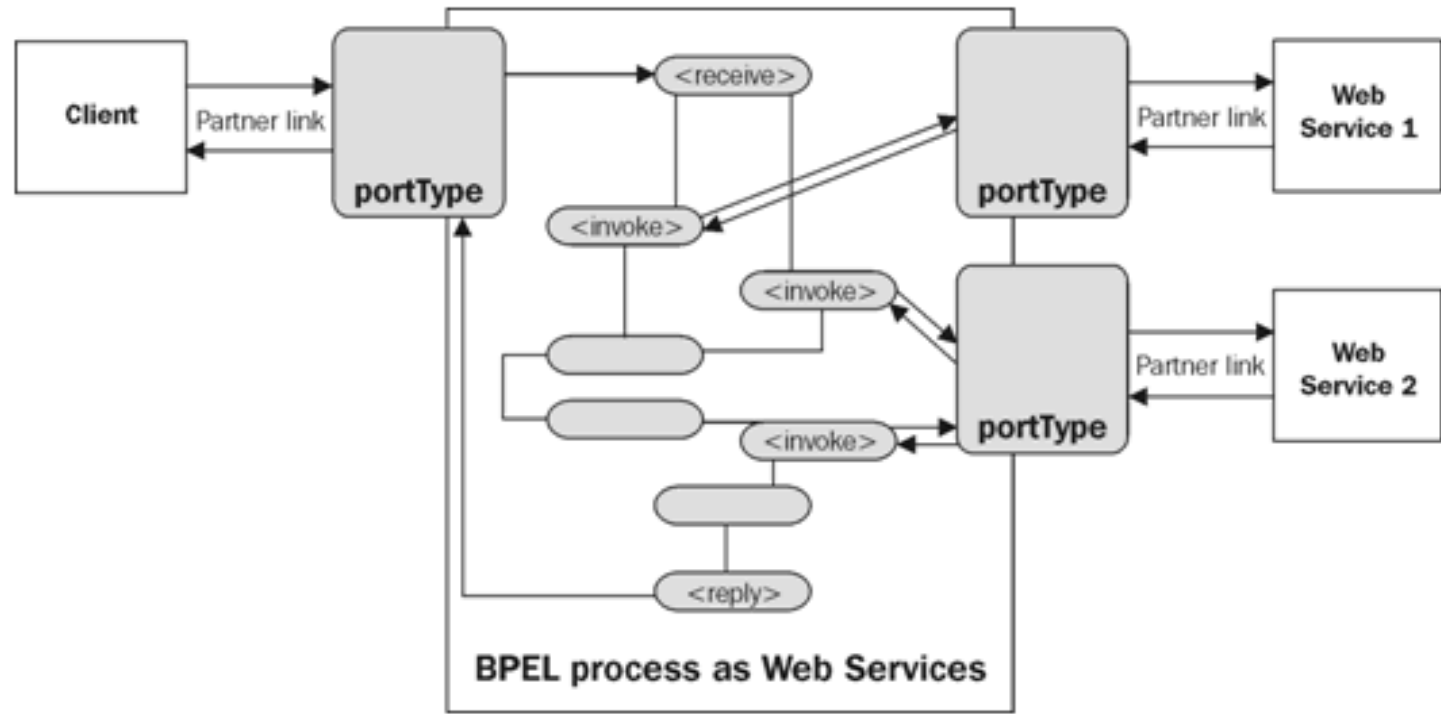


Figure: Example BPEL process



2 Änderungen in BPEL 2.0

- Überblick (Aktivitäten, BPEL-Syntax, Wesentliche Unterschiede zu 1.1)
- Syntaktische Veränderungen
- Data-Handling
- Fault-Handling
- Extension-Konzept
- Abstrakte BPEL-Prozesse

Others

- Partner Link
- Variable
- Properties
- Correlation Set

Basic Activities

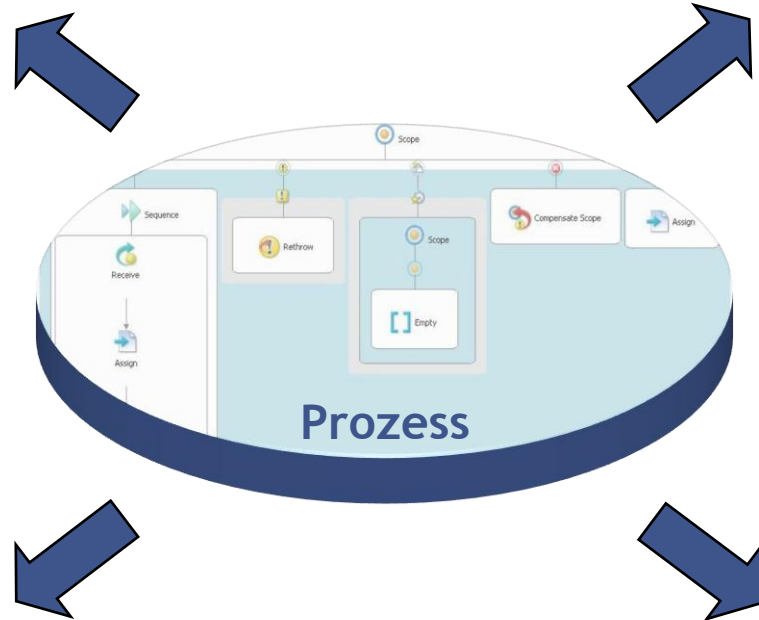
- Receive
- Reply
- Invoke
- Assign
- Throw
- Rethrow
- Exit
- Wait
- Empty
- Compensate
- Compensate Scope
- Validate
- Extension Activity

Handlers

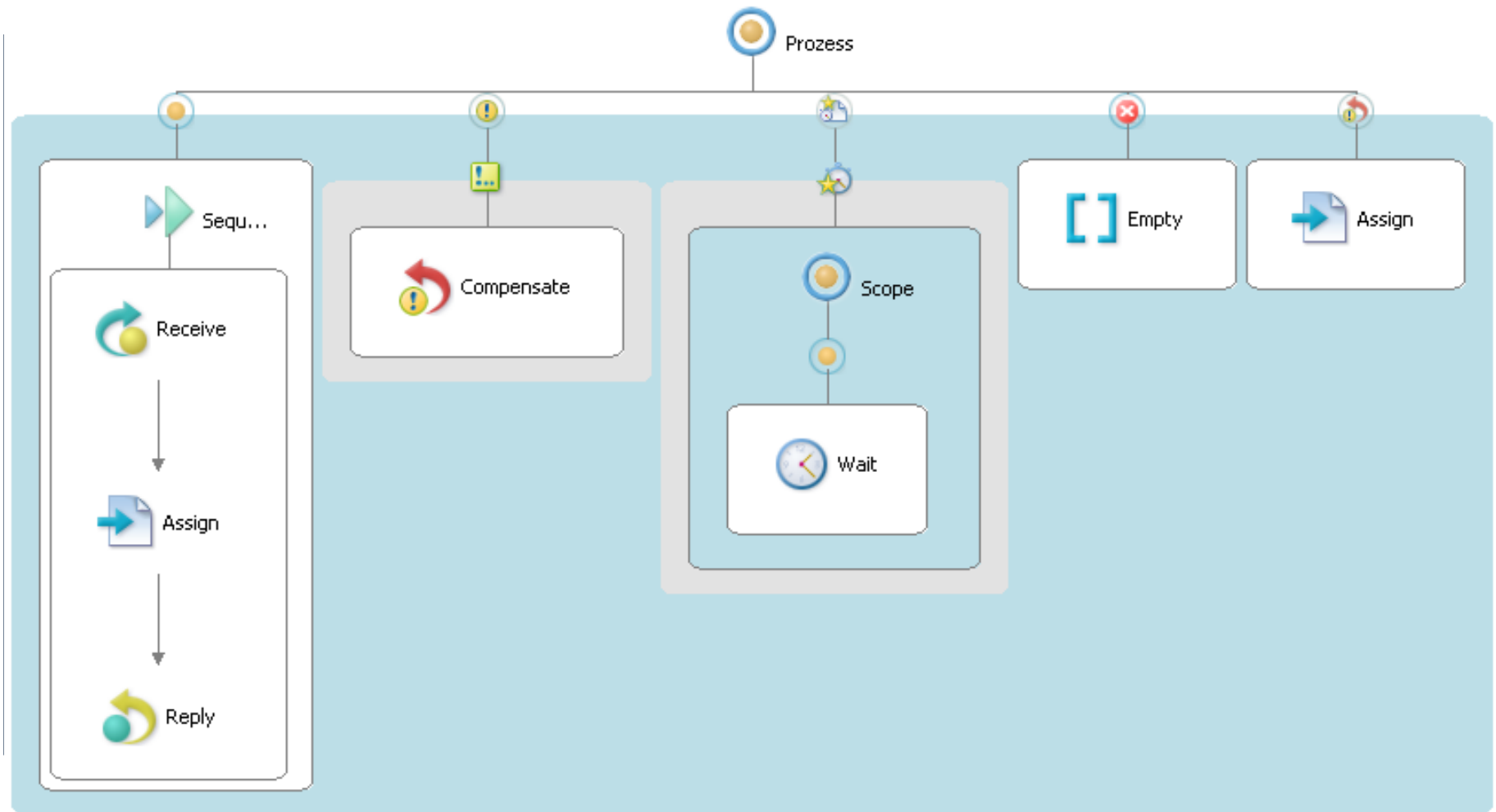
- Compensation Handler
- Event Handlers
- Fault Handlers
- Termination Handler

Structured Activities

- Sequence
- If
- While
- Repeat Until
- Pick
- Flow
- Scope
- For Each















BPEL-Syntax



BPEL4WS 1.1 vs. WS-BPEL 2.0

- Syntaktische Neuerungen
- Vereinfachtes Data-Handling
- Erweitertes Fault-Handling
- Extension-Konzept
- Abstrakte Prozesse

Veränderungen: Syntax

BPEL4WS 1.1	WS-BPEL 2.0
--	 Rethrow
 Compensate	 Compensate  Compensate Scope
--	 Validate
--	 Extension Activity
 Terminate	 Exit
 Switch	 If
--	 Repeat Until
--	 For Each
--	<documentation>, <import>



Veränderungen: Data-Handling

- Erweiterte `<assign>`-Funktionalität
 - Validierung von Variablen
 - Neue `<copy>`-Attribute
 - Verwendung von XSL-Transformationen
- XSD „Complex Type“-Variablen
- WSDL Message-Part-Mapping
- Mehrere Anfragen an ein `<receive>`
- Vereinfachter Variablenzugriff



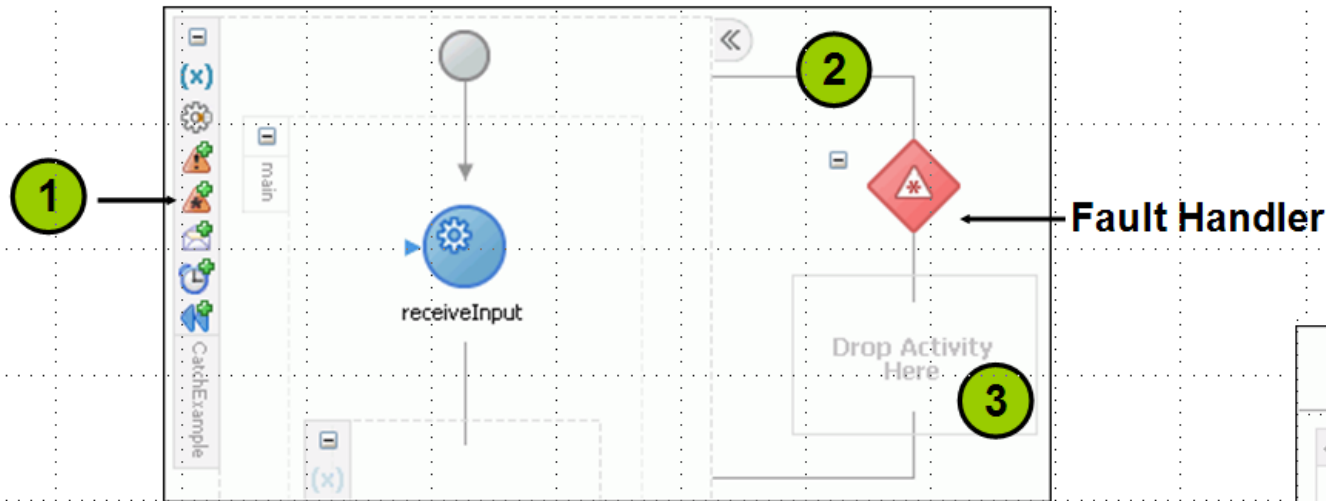
Veränderungen: Data-Handling

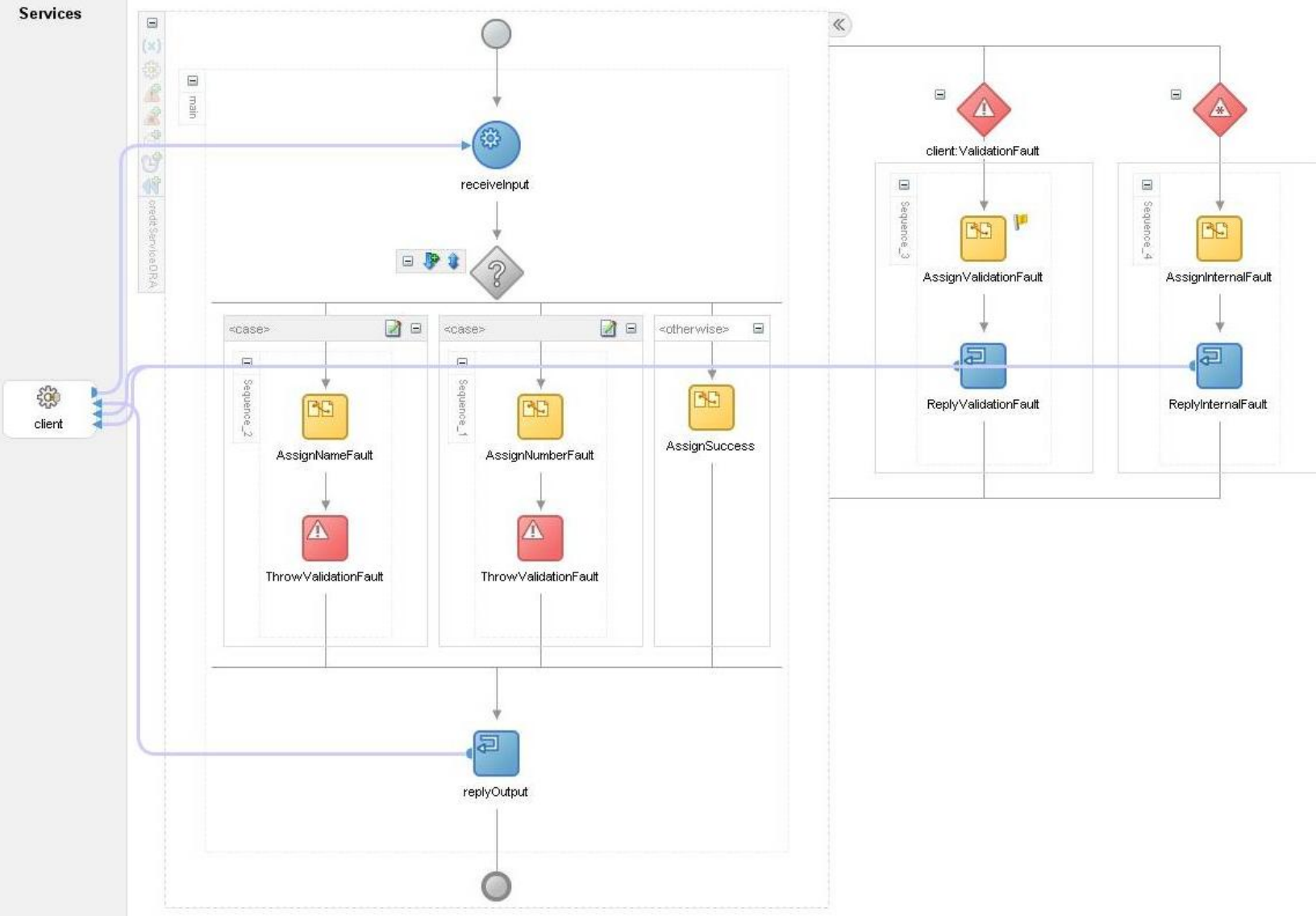
```
<a:foo xmlns:a="http://example.com">  
  <a:bar>23</a:bar>  
</a:foo>
```

```
...  
<variables>  
  <variable messageType="a:foo" name="FooVar">  
</variables>  
...  
<from expression="bpws:getVariableData('FooVar','bar')"></from>
```

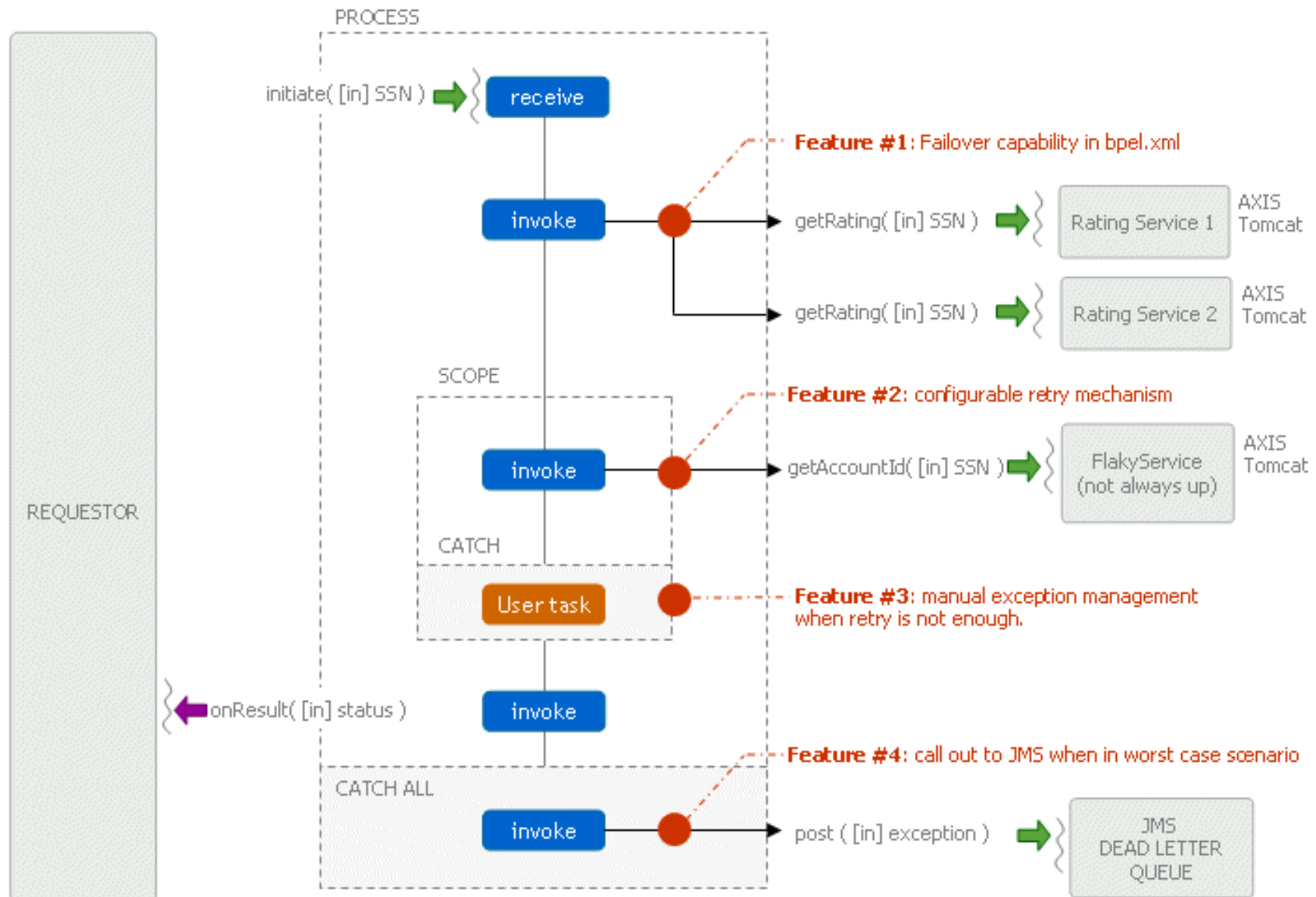
```
<from>$FooVar.bar</from>
```

Fehlerhandling in BPEL



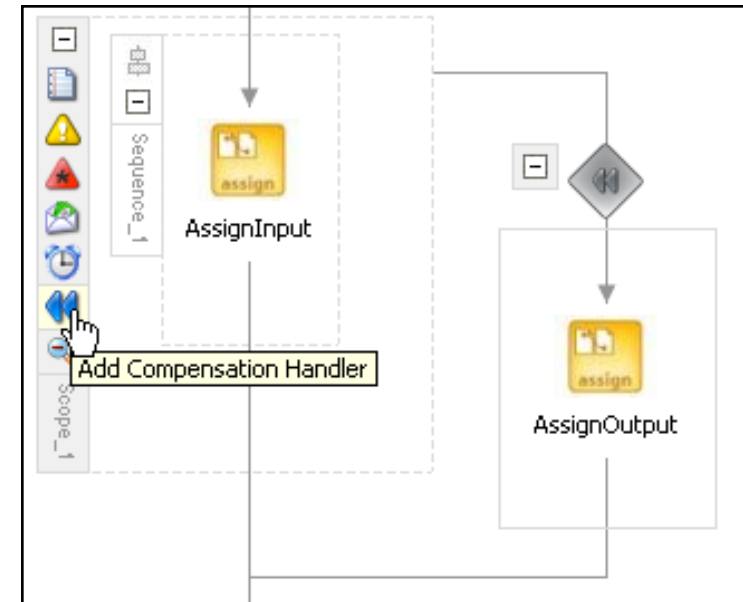


Services



compensationHandler in BPEL

- Eine Compensation Handler-Aktivität erfordert:
 - die Unterstützung einer Undo-Operation in den aufgerufenen Services
 - die Definition eines Compensation Handlers für einen Scope
 - das Auslösen einer Kompensation mit einer `Compensate`-Aktivität (häufig im Fault Handler eines anderen Scopes)





Veränderungen: Fault-Handling

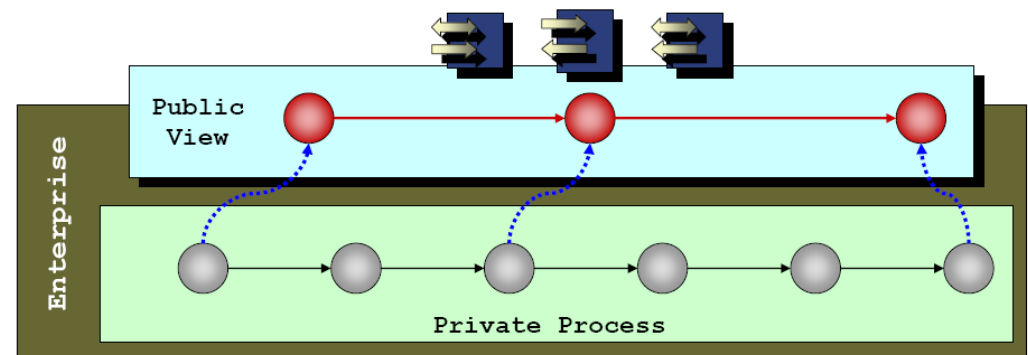
- „Fault-Propagation“ über `<rethrow>`
- Feinere Unterscheidung der Fehler:
 - `faultName`
 - `faultVariable`
 - `faultType`
- **Neuer Handler:** `<terminationHandler>`
- **Verbessertes Compensation-Handling**

Veränderungen: Extension-Konzept

- Bestandteile des Extension-Konzept:
 - `<extensions>`
 - `<extensionActivity>`
 - `<extensionAssignOperation>`
- Vorteile:
 - Erhöhte Modularität
 - Verbesserte Prozessportabilität
- Anwendungsbeispiel: B4P

Veränderungen: Abstrakte Prozesse

- Ausführbare Prozesse
 - Prozess ist sichtbar, Business-Logik ist verborgen
 - Private Sichtweise (BlackBox)
- Abstrakte Prozesse
 - Nicht ausführbar, nur teilweise spezifiziert
 - Öffentliche Sichtweise
 - Profile:
 - Template
 - Observable

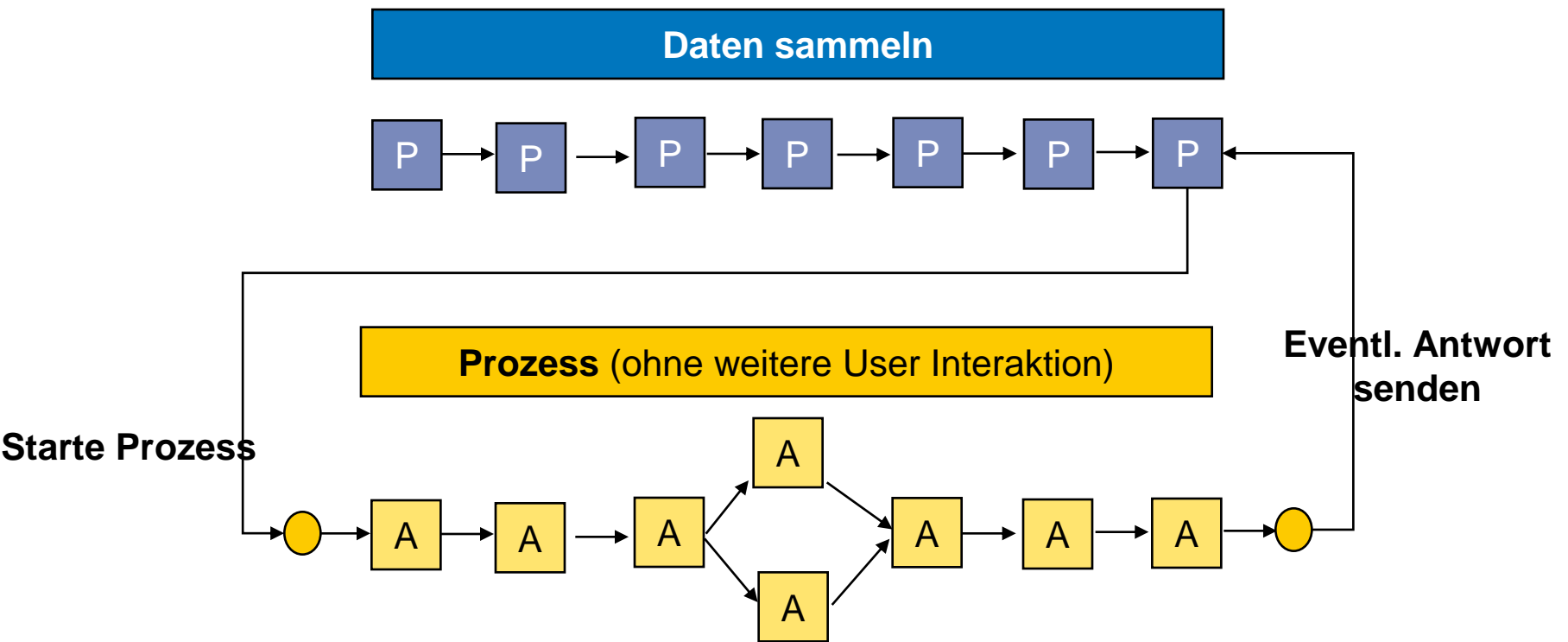





3 **Stand Workflow-Erweiterung**



Standard-Ansatz: Daten sammeln und Prozess anstoßen



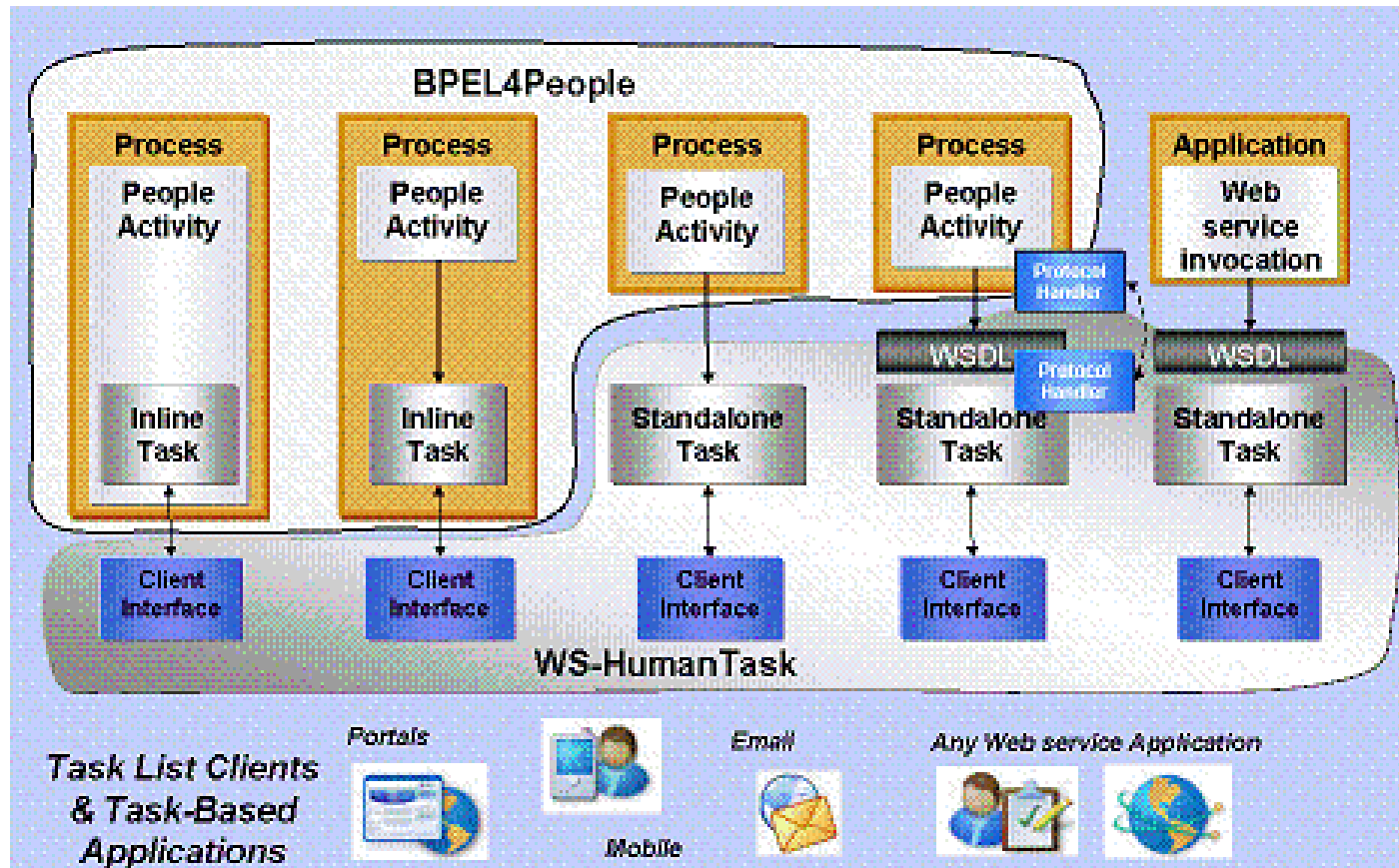
Human Interactions & BPEL

- Auch WS-BPEL 2.0 betrachtet nur vollautomatisiert ablaufende Prozesse
- Keine „Human Interactions“ im Standard berücksichtigt
- Whitepaper *BPEL4People* von IBM und SAP schon vom Juli 2005:
 - <ftp://www6.software.ibm.com/software/developer/library/ws-bpel4people.pdf>
-  **„Human Interactions“ als wesentlicher Bestandteil langlaufender Prozesse**

Human Interactions & BPEL

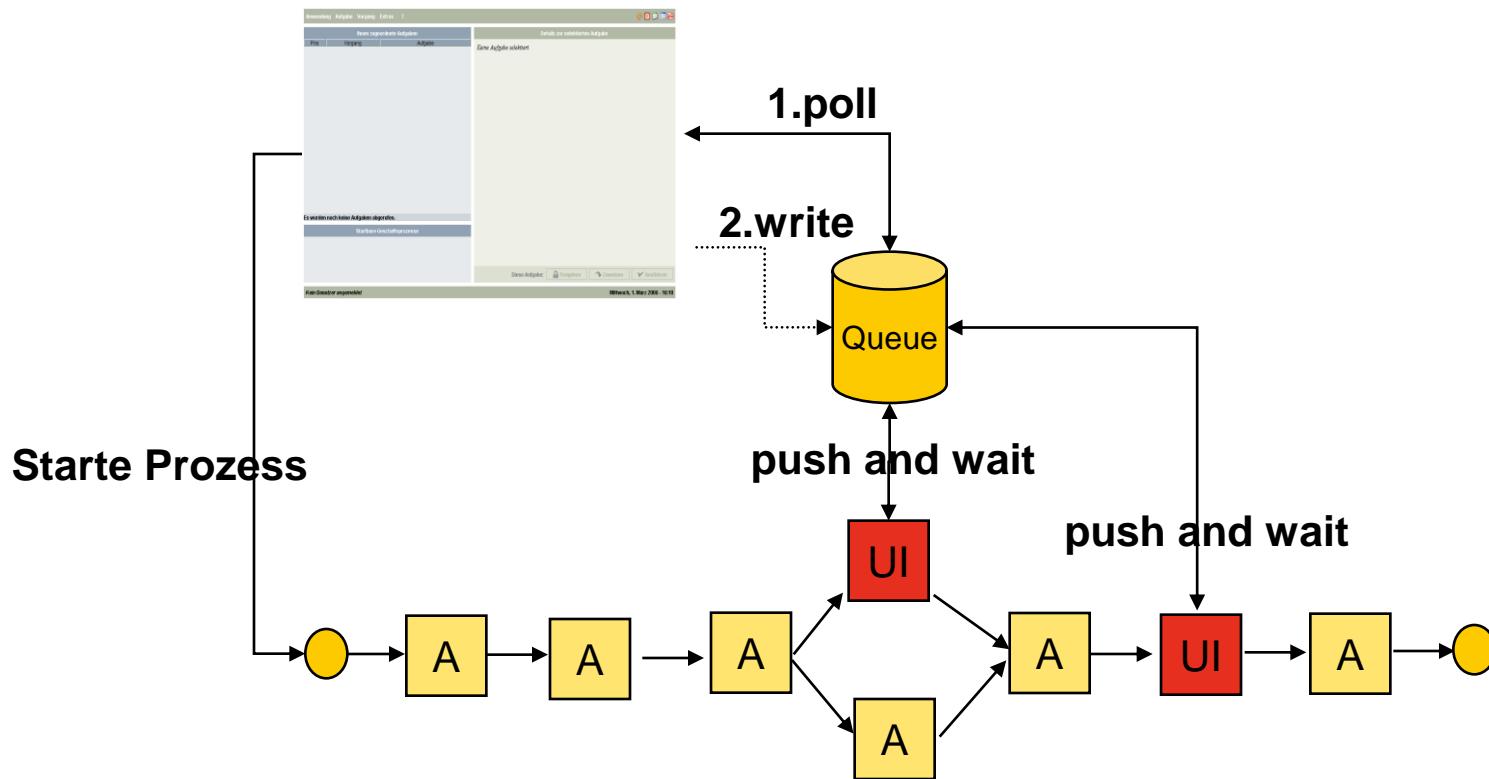
- Integration von Human Interactions über
 - WS-Human Task (WS-HT)
 - WS-BPEL4People (WS-B4P)
- Benefits
 - Administrative Interventionen möglich
 - Eindeutige Beziehung Mensch ↔ Prozess
 - „Mächtiger“ Prozesse

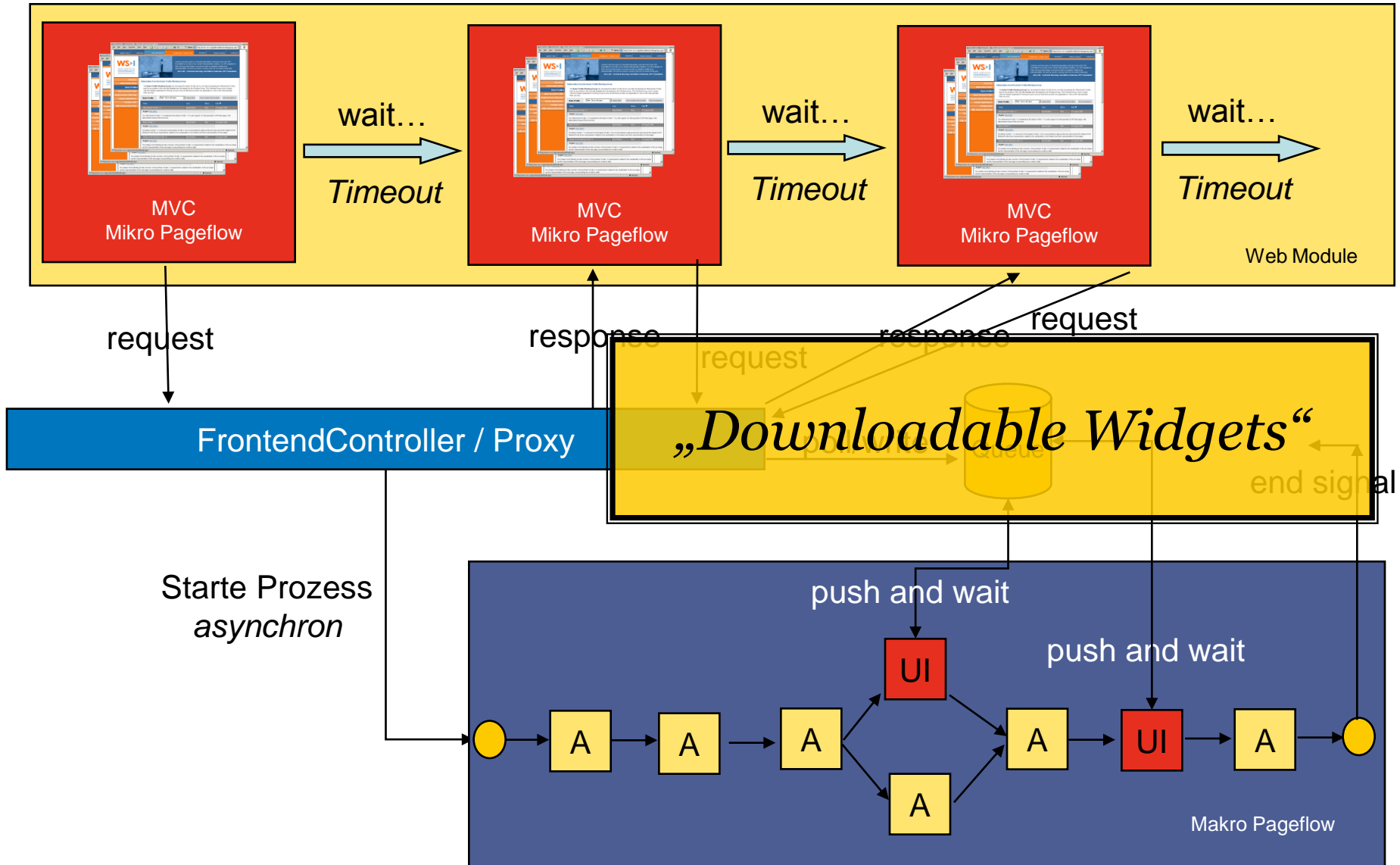
BPEL 2.0-Extensions: WS-HT & WS-B4P

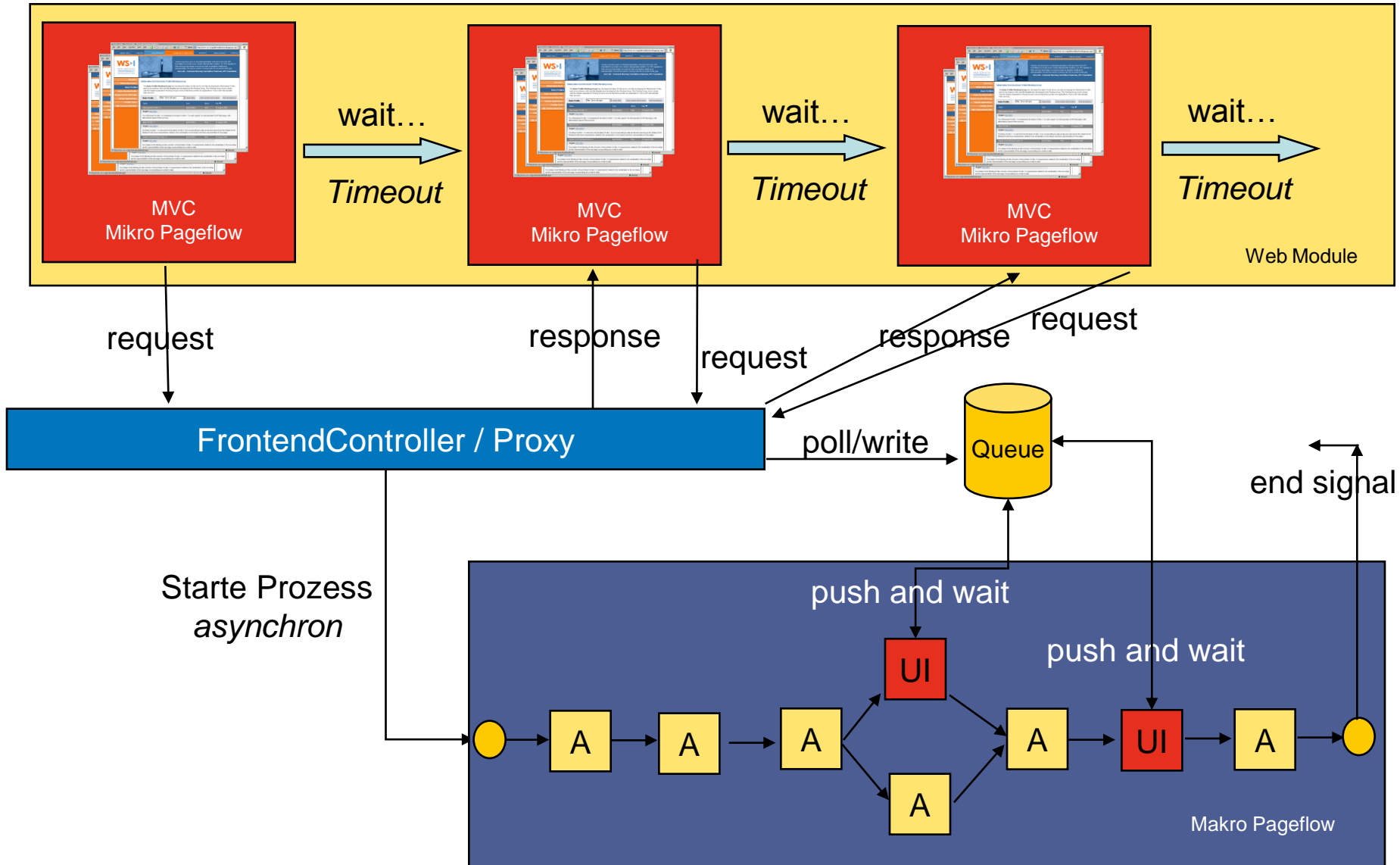


Bildquelle: <https://weblogs.sdn.sap.com/pub/wlg/6790>; abgerufen am 08.10.07

Arbeitsliste für langlaufende Prozesse









4 Tools für BPEL 2.0

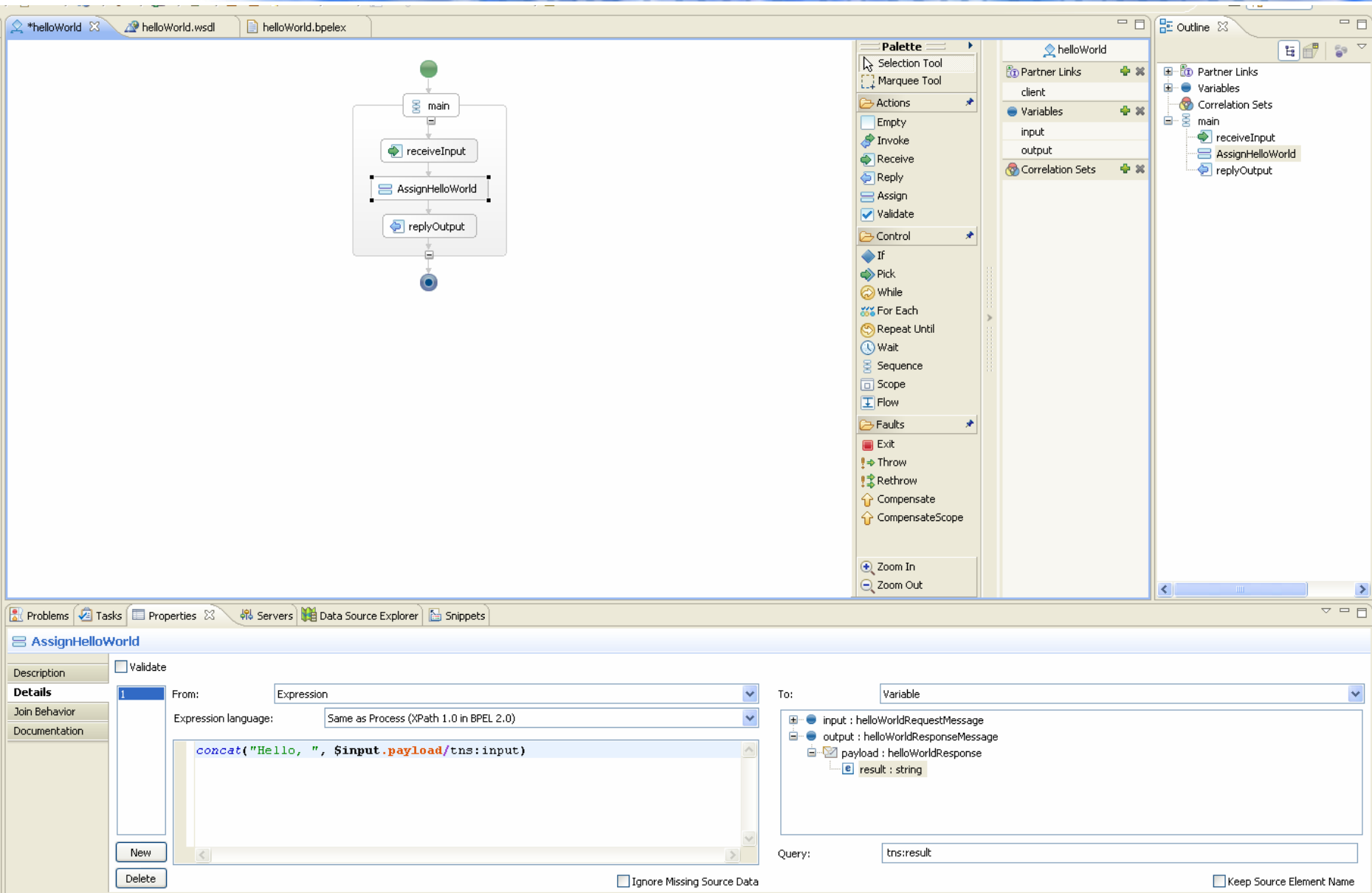


Tools für BPEL 2.0

- Eclipse.org
 - Plugin: BPEL Designer
- Active Endpoints
 - Active BPEL Designer 4.1, AE Enterprise/OpenSource
- Oracle
 - SOA Suite 11, Jdeveloper 11, Eclipse Plugin
- Weitere:
 - Sun, Application Server 9.1, Glassfish v2 (OpenESB)
 - Bea, Aqualogic BPM 6.0
 - Intalio, Apache ODE, Eclipse-basierter BPEL Designer

Eclipse BPEL-Plugin

- Kooperation von IBM und Oracle
- BPEL 2.0-Support
 - Fehlende Einstellungsmöglichkeiten für Aktivitäten
 - Prozessvalidierung
- Features:
 - Deployment zu unterschiedlichen Engines
 - Diverse Editoren (u.a. für XSD, WSDL, XPath)
 - Synchronisation Design ↔ Sourcecode



AssignHelloWorld

Validate

Description

Details

Join Behavior

Documentation

From: Expression

Expression language: Same as Process (XPath 1.0 in BPEL 2.0)

```
concat("Hello, ", $input.payload/tns:input)
```

To: Variable

- input : helloWorldRequestMessage
- output : helloWorldResponseMessage
 - payload: helloWorldResponse
 - result : string

Query: tns:result

Ignore Missing Source Data Keep Source Element Name

Demo

The screenshot displays a BPEL editor interface. The main workspace shows a process diagram with a 'main' block containing 'receiveInput', 'AssignHelloWorld', and 'replyOutput' tasks. The 'AssignHelloWorld' task is selected, and its configuration is shown in the bottom panel. The configuration includes the following details:

- From:** Expression
- Expression language:** Same as Process (XPath 1.0 in BPEL 2.0)
- To:** Variable
- Expression:** `concat("Hello, ", $input.payload/tns:input)`
- Variables:**
 - input : helloWorldRequestMessage
 - output : helloWorldResponseMessage
 - payload : helloWorldResponse
 - result : string
- Query:** tns:result

The interface also features a 'Palette' on the right with various BPEL activities and a 'Outline' on the far right showing the process structure.

ActiveBPEL: Überblick

- For free Design-Tool ActiveBPEL™ Designer
 - Eclipse Ready Technology (Eclipse 3.2)
 - Standard BPEL-Support
- BPEL-Engine
 - OpenSource: ActiveBPEL™ Engine
 - Kommerziell: ActiveBPEL™ Enterprise

Verwendung verschiedener J2EE-Applikationsserver möglich

ActiveBPEL: Features

- Prozessdebugging/-simulation zur Design-Zeit
- Drag and Drop Prozesserstellung
- Automatisiertes Deployment
- Abwärtskompatible BPEL-Engine
- Automatische Prozessmigration
- Integration mit CentraSite (Software AG)



ActiveBPEL: Features

- BPEL-Source read-only
- Kein integrierter WSIF-Support
- Integration von Human Interactions nur mit Enterprise Engine
- Dokumentation



ActiveBPEL - tutorial.bpel - ActiveBPEL Designer

File Edit Navigate Search Project Run Process FieldAssist Window Help

100% Check for Software Updates

test.bpel tutorial.bpel tutorial1_1.bpel test1_1.bpel test2_0.bpel Präsentation.bpel

ActiveBPEL J2EE

Navigator

- ActiveBPEL_Tutorial
 - bpr
 - WSDL
 - .project
 - errorCode.xml
 - Präsentation.bpel
 - test.bpel
 - test.pdd
 - test1_1.bpel

Web References

Web References - WSDL

- echoDefs.wsdl
- testService.wsdl

Process Variables

- request
- approval
- risk
- error

Palette

- Selection
- Marquee
- Link
- Activity
 - Operation Wizard
 - Receive
 - Reply
 - Invoke
 - Assign
 - Throw
- Container
 - Sequence
 - If
 - While
 - Repeat Until
 - Pick
 - Flow
 - Other
 - If Condition
 - Else
 - Catch
 - Catch All
 - On Event
 - On Alarm
- Drawing
 - Custom

Process Activities | Fault Handlers | Event Handlers | Source

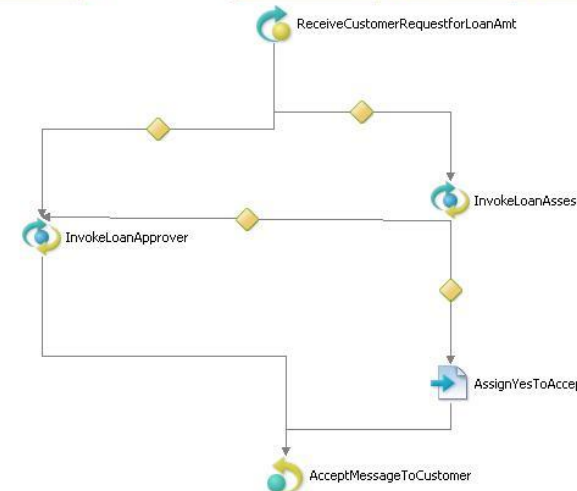
Outline

- tutorial
 - Extensions
 - Imports
 - Namespaces
 - Partner Links
 - Variables
 - Process Activities

Problems Tasks Error Log Properties

Property	Value
Info	
derived	false
editable	true
last modified	3/30/07 3:36 PM
linked	false
location	C:\svb\ActiveBPELDesigner\Designer\workspace\ActiveBPEL_Tutorial
name	ActiveBPEL_Tutorial
path	/ActiveBPEL_Tutorial

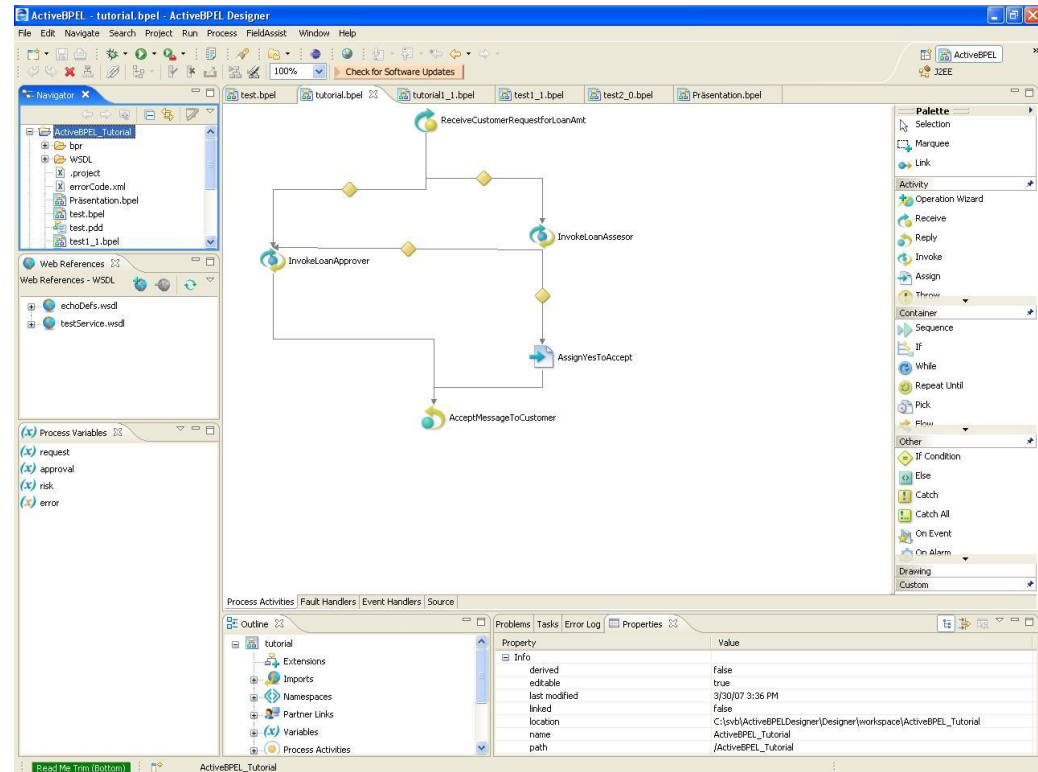
Read Me Trim (Bottom) ActiveBPEL_Tutorial



```

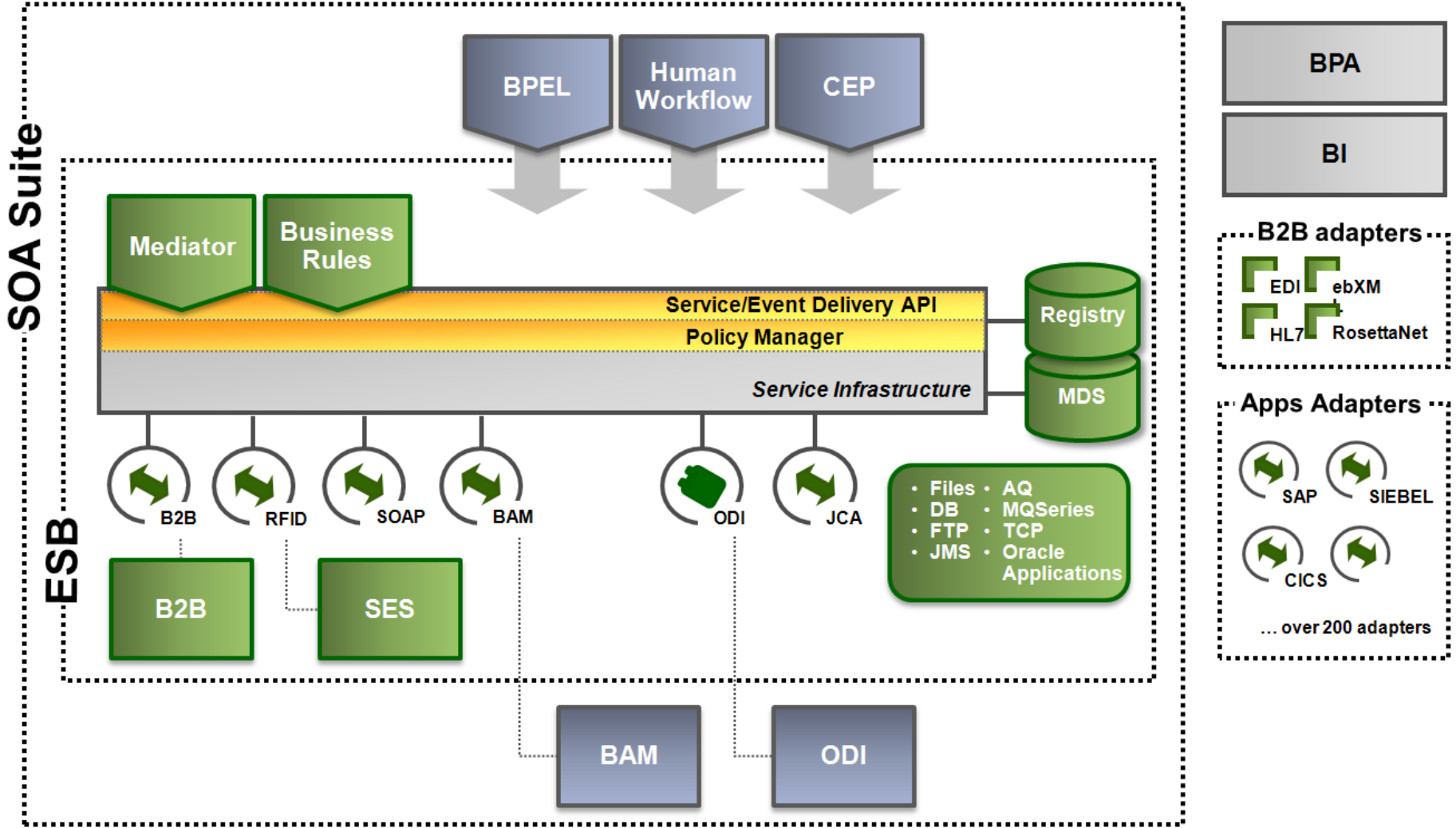
    graph TD
      Start(( )) --> Receive[ReceiveCustomerRequestForLoanAmt]
      Receive --> Diamond1{ }
      Diamond1 --> InvokeLoanAssesor[InvokeLoanAssesor]
      InvokeLoanAssesor --> Diamond2{ }
      Diamond2 --> AssignYesToAccept[AssignYesToAccept]
      AssignYesToAccept --> AcceptMessageToCustomer((AcceptMessageToCustomer))
      Diamond2 --> InvokeLoanApprover[InvokeLoanApprover]
      InvokeLoanApprover --> Diamond3{ }
      Diamond3 --> Diamond1
  
```

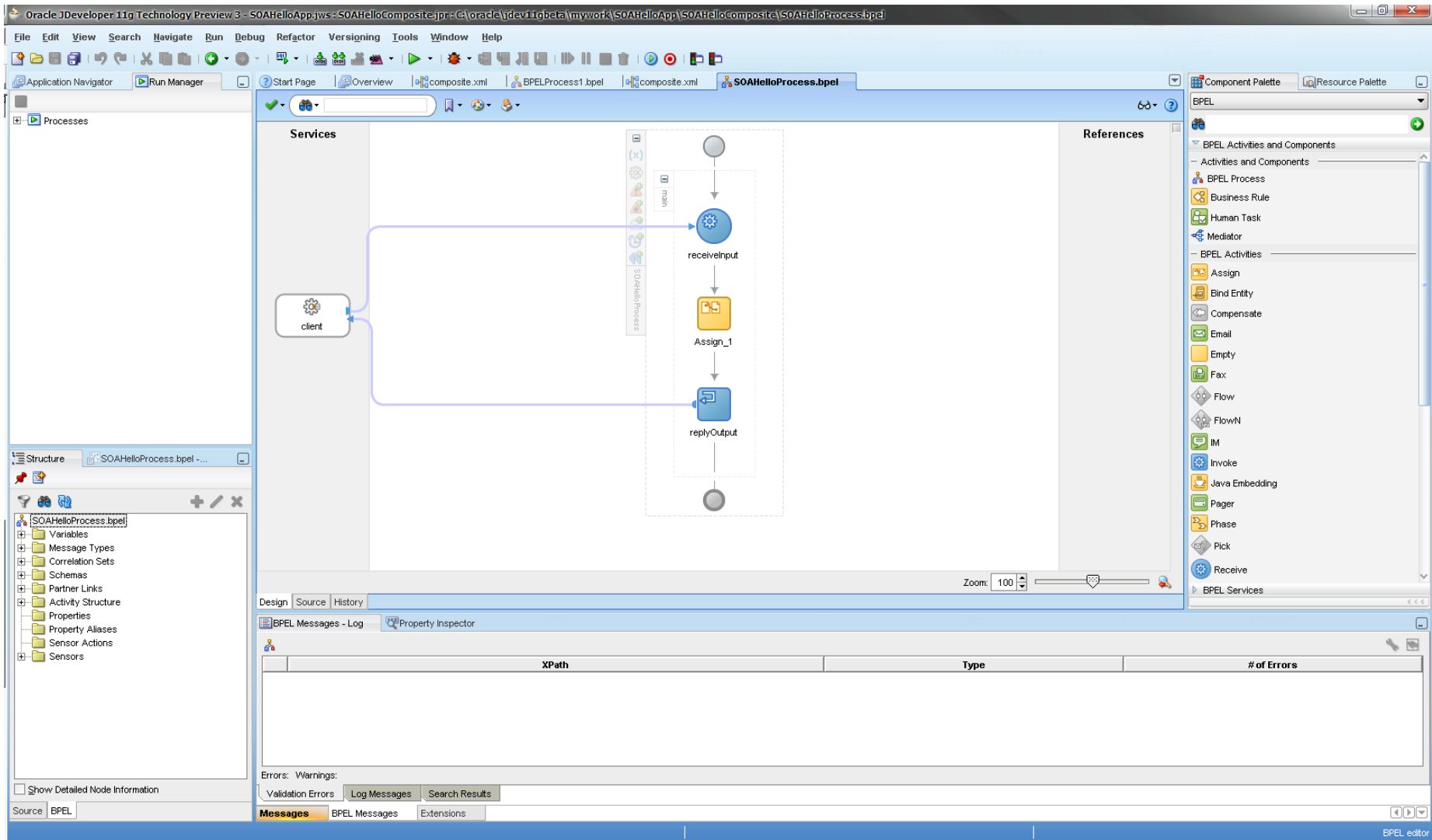
Demo



Oracle SOA Suite 11g

- Ganzheitliche Suite mit div. Komponenten:
 - BPEL, ESB/Mediator, Rules, EDN, CEP, BAM, WSM, Adaptoren
- Ausgereiftes BPEL-Werkzeug:
 - Oracle BPEL Process Manager (ehemals Collaxa, seit BPEL4WS 1.1 bis 10g)
 - WS-BPEL 2.0 ab 11g (Aktuell Beta; Release in 2008)
 - Eingebettet in SCA Infrastruktur
 - Unterstützung Human Tasks, Business Rules, BAM
 - Error Hospital
 - SOA Console für ganzheitliches Monitoring
 - Clusterbetrieb
 - Dehydration über Coherence/Tangosol

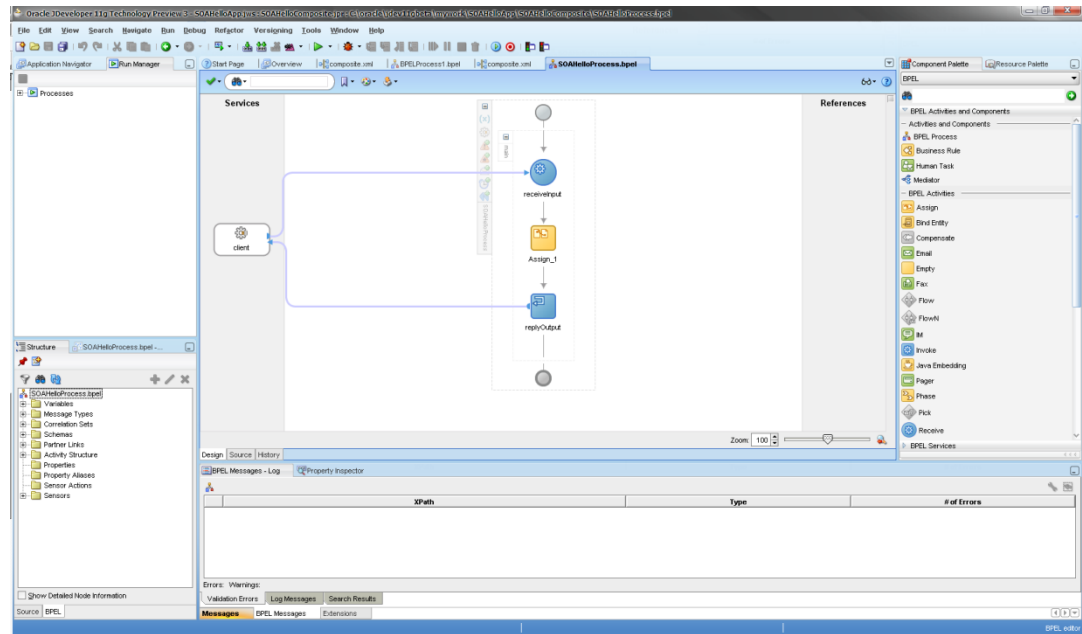




The screenshot displays the Oracle JDeveloper 11g IDE interface for editing a BPEL process. The main workspace shows a process diagram with a 'client' activity connected to a 'main' process block. Inside the 'main' block, the flow is: receiveInput (Receive) -> Assign_1 (Assign) -> replyOutput (Reply). The Component Palette on the right lists various BPEL activities and components. The Structure view on the left shows the project hierarchy. The Messages view at the bottom is currently empty.

XPath	Type	# of Errors

Demo





5 Migration von BPEL4WS 1.1 auf WS-BPEL 2.0



Toolunterstützung: ActiveBPEL™ Designer

- Umwandlung von BPEL4WS 1.1 zu WS-BPEL 2.0
 - Automatisierter, intuitiver Umwandlungsprozess
- Migrationsschritte:
 - Syntaxkonvertierung
 - Aktivitätenumwandlung
 - Konvertierung Variablenzugriffsfunktionen
 - Modifizierung `<catch>`-Aktivitäten
 - Konvertierung Event-Handler



6 Bewertung

- Was ist mit Hilfe der Neuerungen möglich?
- Praktischer Nutzen für die Prozesse in Unternehmen?

Vorteile einer BPEL-basierten Anwendung

- **Prozessorientierte Anwendung**
 - Gesamte Ablauf-Logik ist im BPEL-Prozess definiert
 - Gesamte Business-Logik ist in Services hinterlegt
 - Prozess wird in IT-Lösung "sichtbar": BPEL-Flow
- **Verwendung von Standards**
 - Investitionssicherheit, Herstellerunabhängigkeit
- **Einbindung beliebiger Services möglich**
 - Alle Komponenten mit Web Service Interface
 - Über WSIF aber auch: EJB, JCA, JMS, Local Java
- **Technisches Monitoring über BPEL-Console**
 - Prozesszustände jederzeit instanzgenau abrufbar

Bewertung der Neuerungen in WS-BPEL 2.0

- Abstrakte Prozesse
 - Bereitstellung von Prozesstemplates/Best-Practises
 - Erhöhte Wiederverwendbarkeit
 - Effizientere/Schnellere Prozesserstellung
 - Verbergen geschäftskritischer Prozesselemente
- Bessere Dokumentationsmöglichkeiten
- Variablen
 - Übersichtlicherer Sourcecode durch neue Möglichkeiten mit XPath
 - Variablentransformationen
- Feinschliff, BPEL wird „runder“ und exakter!!
- „Clarifications“ gegenüber BPEL 1.1

Weitere Möglichkeiten I:

- **Überbau mit fachlicher Prozessmodellierung**
 - Bspw. EPK (ARIS, Oracle BPA Suite)
 - Generierung der BPEL-Prozesse aus den fachlichen Prozessen (Roundtrip, IDS SOA Architect, Oracle BPA Suite)
- **Fachliches und technisches Service Portfolio**
 - Service Lifecycle Management
 - Gemeinsames Repository für Prozesse und ihre Services
 - Governance

Weitere Möglichkeiten II:

- **Fachliches Monitoring (BAM)**
 - Echtzeit Prozess-Controlling
 - Messung prozessbasierter KPIs in Echtzeit
 - Darstellung in Dashboards (Management-Cockpit)
 - Versenden von „Alerts“ bei Grenzwertverletzungen
 - Mehrdimensionale Auswertung der Kennzahlen (vgl. BI, DW)



Fazit

BPEL ist für die Prozess-/Workflow-Technologie, was SQL für relationale Datenbanken ist !

- BPEL ist die technisch ablauffähige Umsetzung fachlicher Prozesse
- BPEL hilft bei Verkleinerung des Business-IT-Gap
- BPEL ist ein robuster, modularer Standard
- WS-HT & WS-B4P integrieren Menschen in Prozesse
- Jetzt ist die richtige Zeit, um einzusteigen!

Vielen Dank für Ihre Aufmerksamkeit!

Kontaktinformationen:



OPITZ CONSULTING

Sven Bernhardt
Dipl.-Wirt.-Inf. (FH)
Anwendungsentwickler

OPITZ CONSULTING Gummersbach GmbH

Kirchstr. 6 - 51647 Gummersbach
Phone +49(0)2261/6001-0 - Fax -4200 - Mobile +49(0)172/2193529
sven.bernhardt@opitz-consulting.de - www.opitz-consulting.de



OPITZ CONSULTING

Torsten Winterberg
Dipl.-Ing. / Dipl.-Wirt.-Ing.(FH)
Bereichsleiter Anwendungsentwicklung – Oracle ACE Director

OPITZ CONSULTING GmbH

Kirchstr. 6 - 51647 Gummersbach
Phone +49(0)2261/6001-0 - Fax -4200 - Mobile +49(0)173/5479302
torsten.winterberg@opitz-consulting.de - www.opitz-consulting.de