Oracle BPEL Training

Basheer Khan
Innowave Technology

Agenda

- Concepts
- Architecture
- Installation
- Hello World BPEL Process
- Synchronous Web Service
- Asynchronous Web Service
- Parallel Processing
- Conditional Branching

Agenda (cont'd)

- Database Connections
- File Load BPEL Process
- Polling Tables
- Changing Payload
- Notifications
- Sequences
- Stress Testing
- Application Integration

Concepts

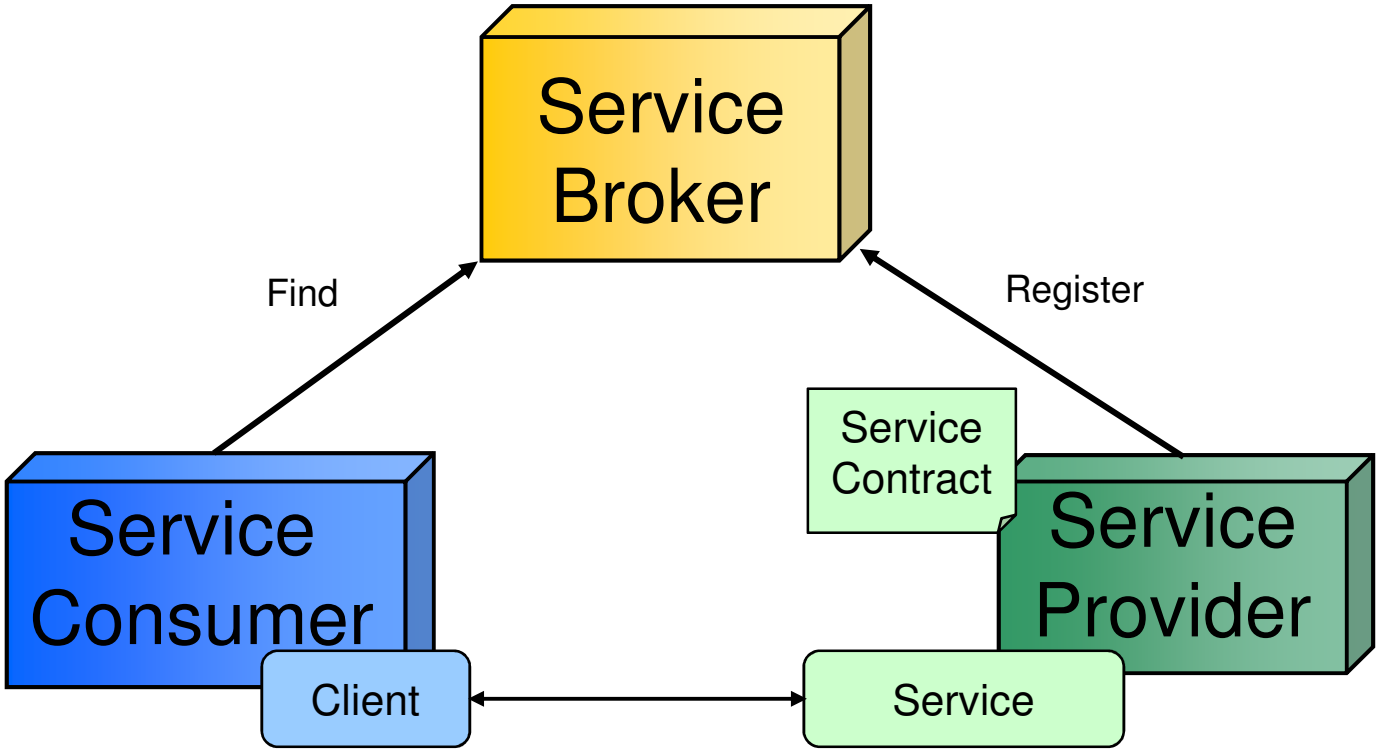
Concepts – Web Services

- C2C – Consumer to Consumer
  - Person (consumer) to person
  - Email (SMTP, RFC-822)
- B2C – Business to Consumer
  - Person to Application
  - Web (HTTP, HTML)
- A2A – Application to Application
  - Application to Application
  - **Web services (XML, SOAP, UDDI, WSDL)**

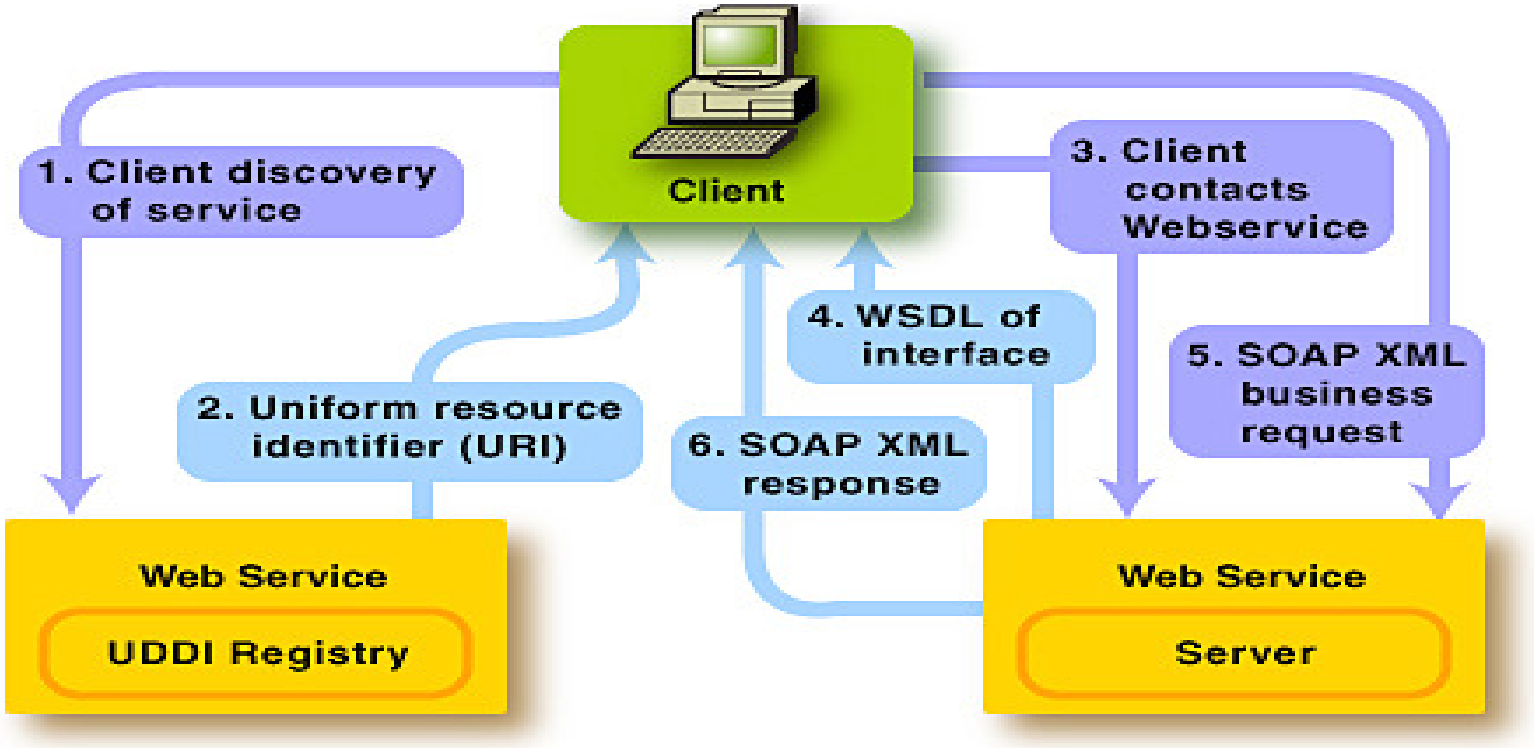
Concepts – Web Services

- **A Web service …**
  - Exposes and describes itself
  - Allows other services to locate it on the Web
  - Can be invoked
  - Must return a response

Concepts – Web Services



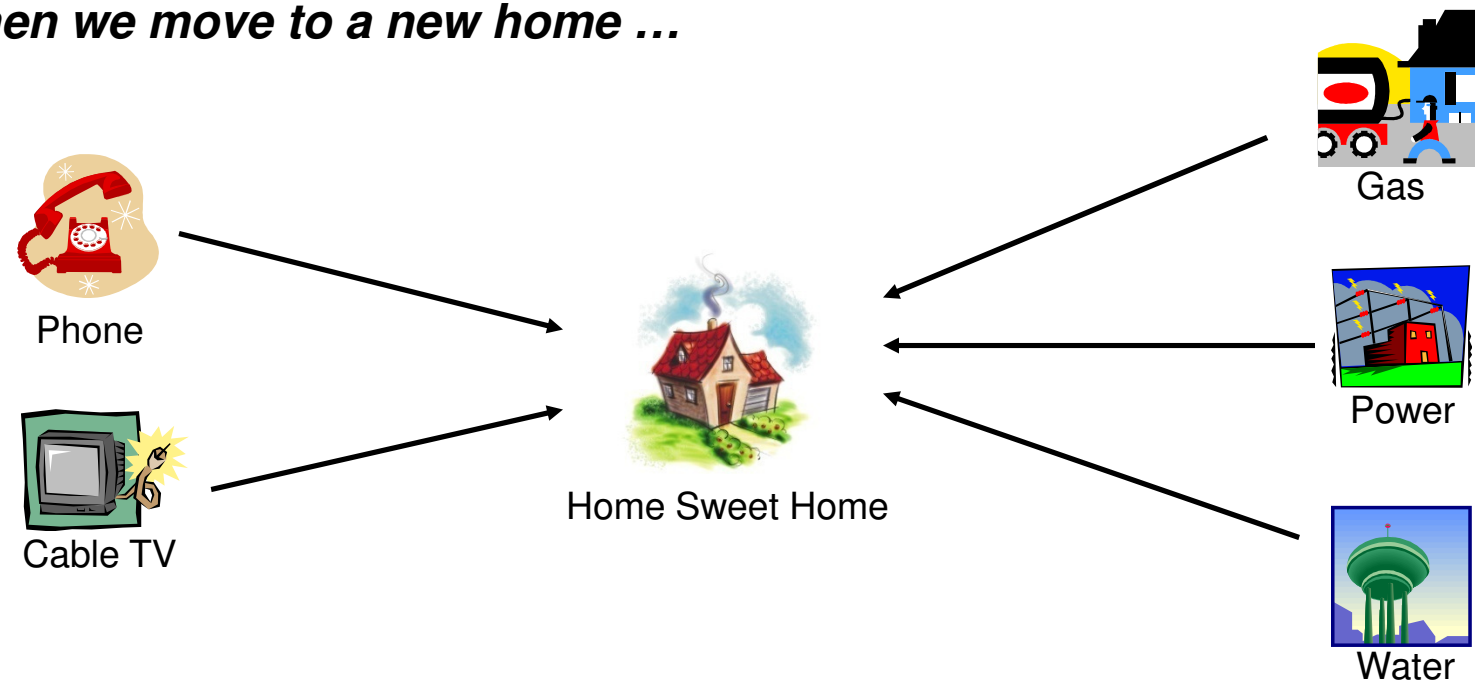
Concepts – Web Services



Diagram, courtesy of IBM - 2005

Concepts – SOA

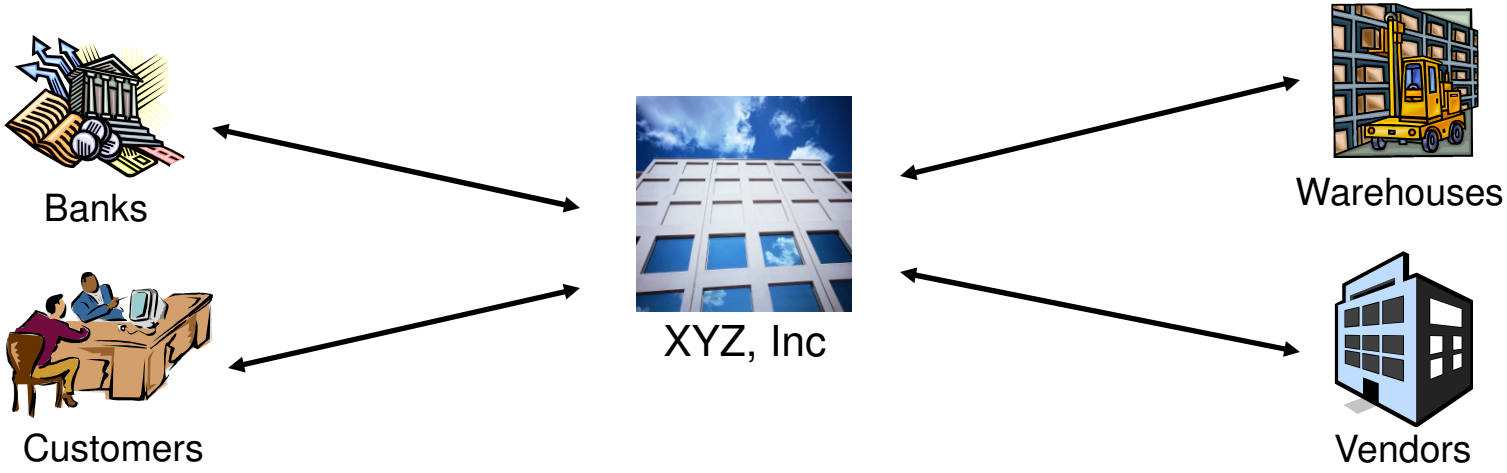
When we move to a new home …



… we establish and use (everyday) a Service-Oriented Architecture!

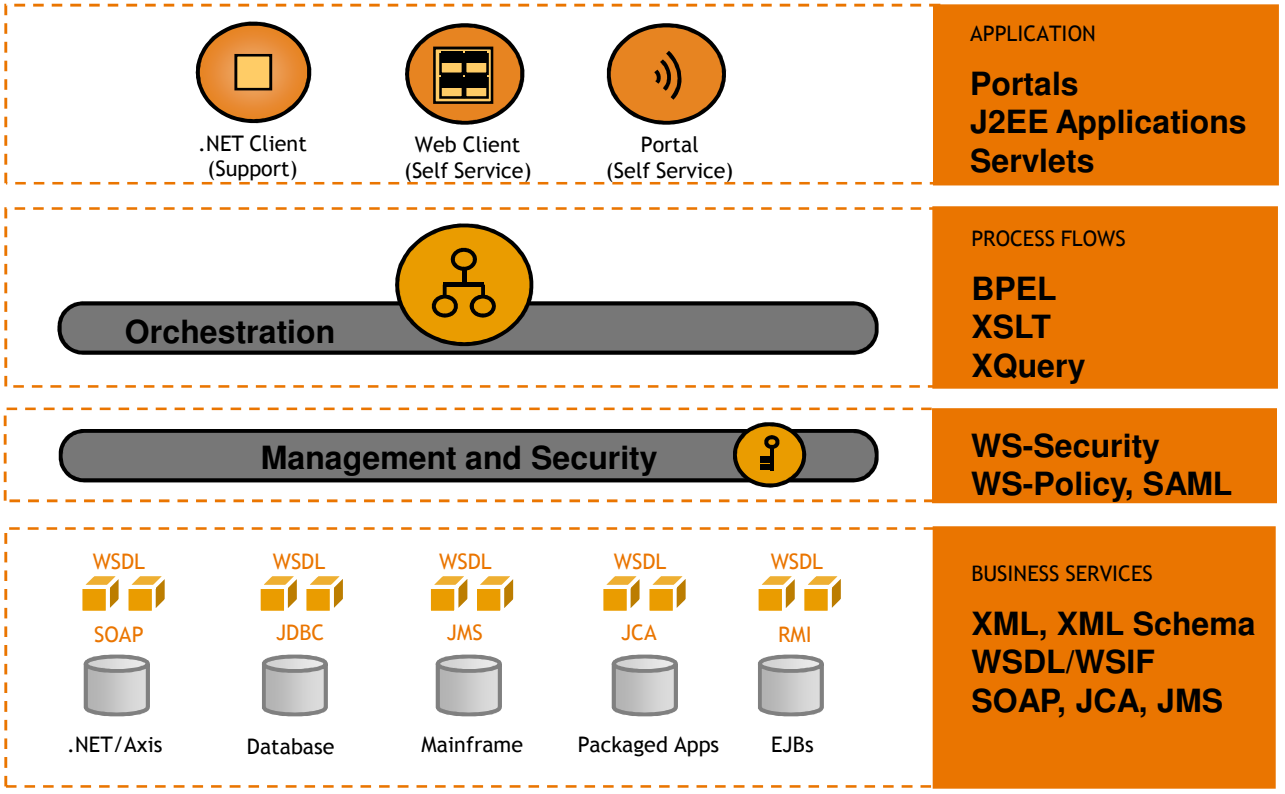
Concepts – SOA

Yet, why is it that most organizations today …



… build their own bridges to interact between applications or with partners?

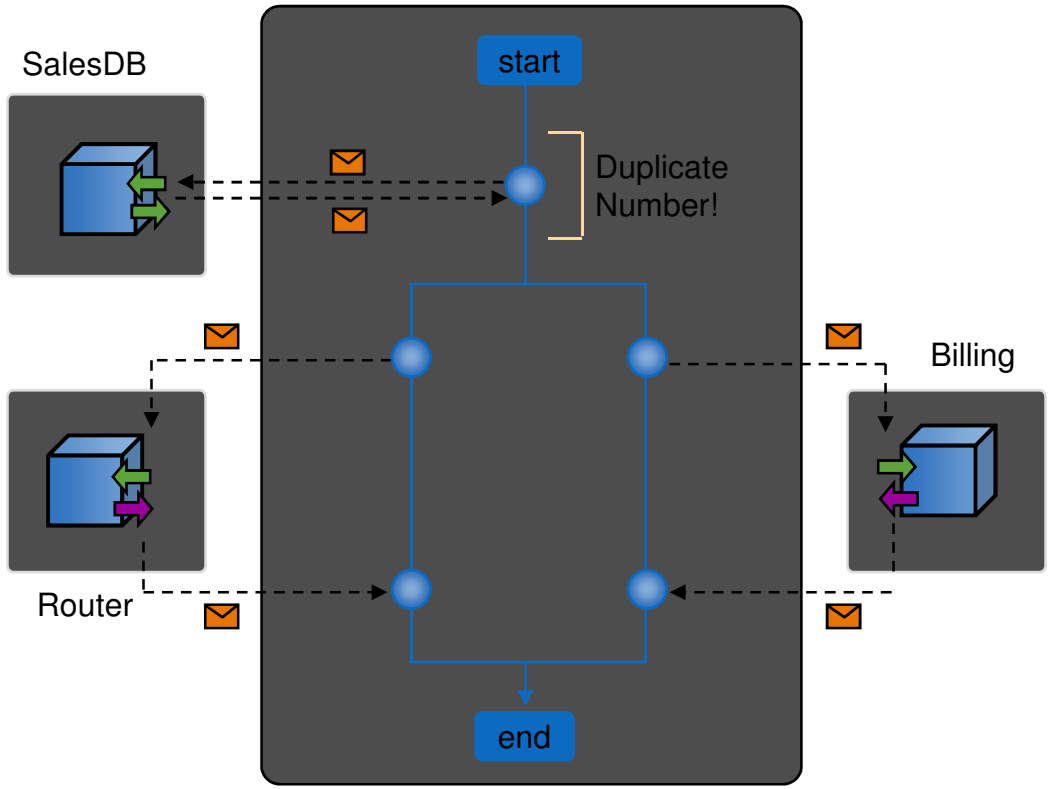
Concepts – SOA



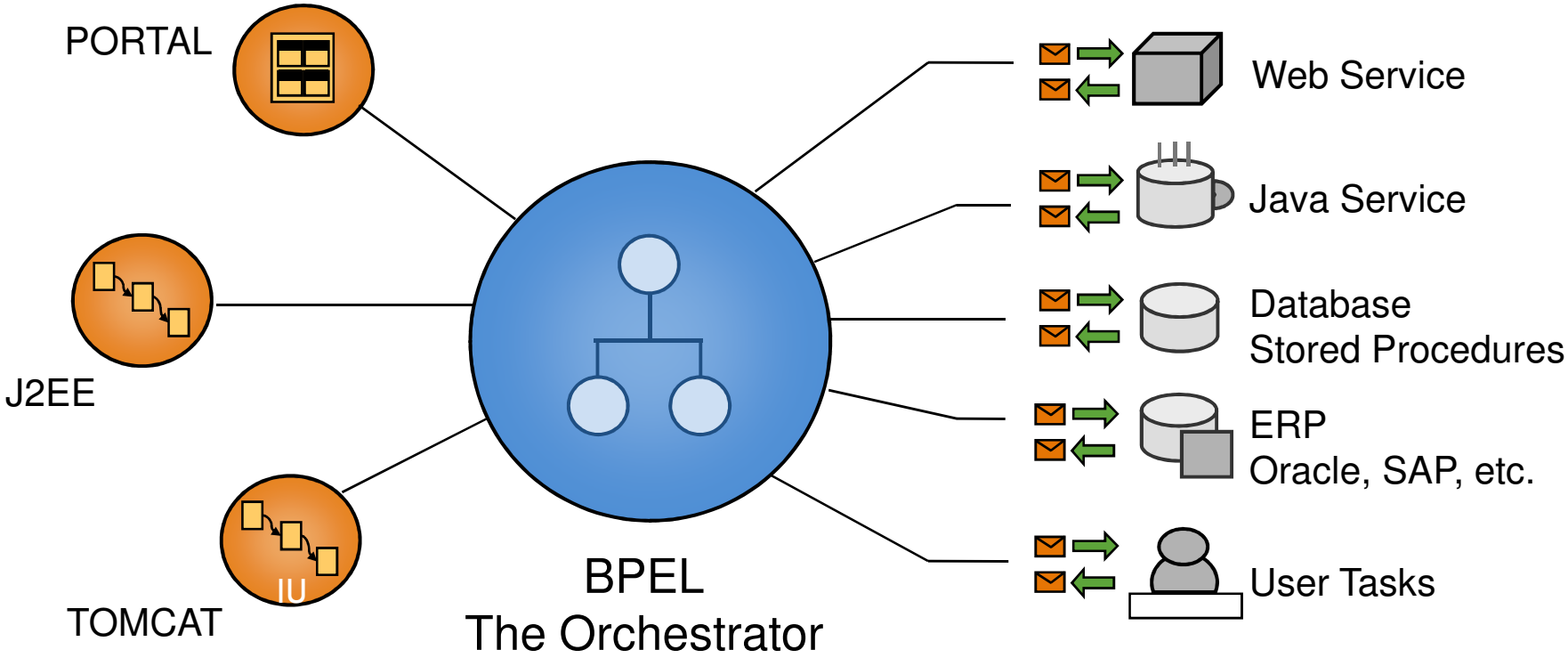
Concepts – BPEL

Business Process Execution Language: Markup language for composing a set of discrete services into an end-to-end process flow

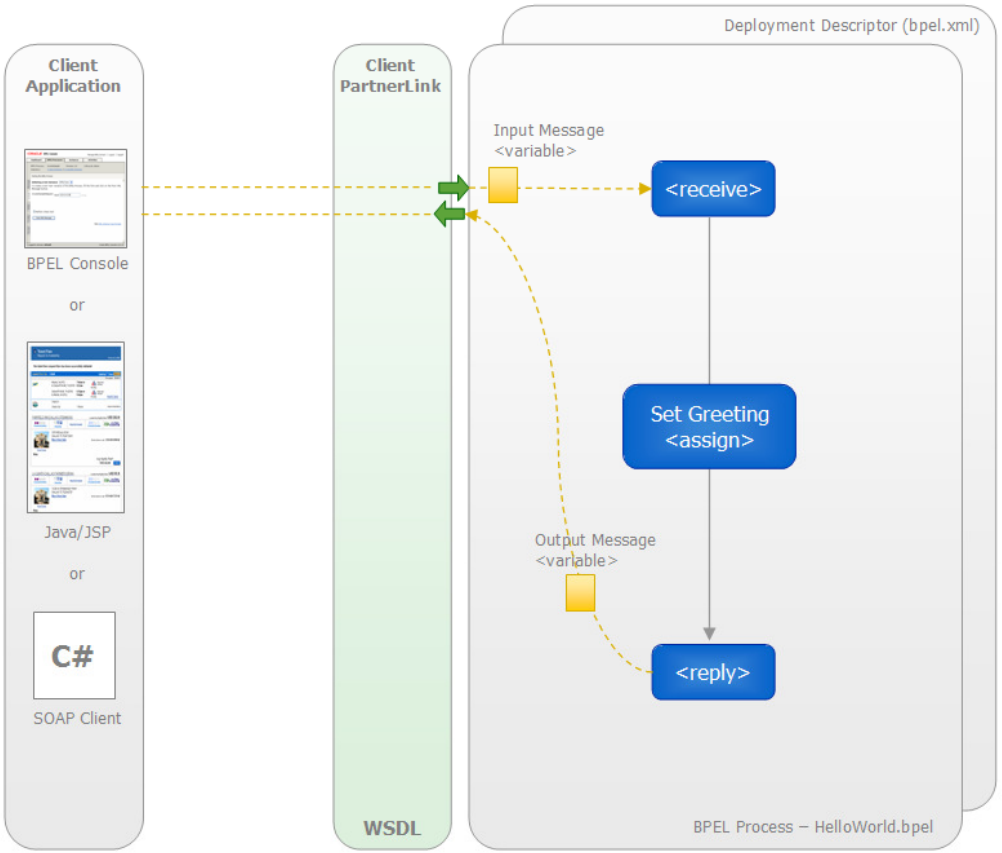
- 10+ years of R&D from MSFT and IBM
- SOAP but also Java, JCA
- Rich Flow Semantics
- Optimized Bindings
- XPATH+XSLT+XQuery
- WS-Security
- A Process is a Service



Concepts – BPEL



Concepts – BPEL



Concepts – BPEL

```
1 <!-- HelloWorld BPEL Process [Generated by the Oracle BPEL Designer] -->
2 <process name="HelloWorld" targetNamespace="http://acm.org/samples" suppressJoinFailure="yes" xmlns:tns="http://acm.org/
3     <!-- ------------------------------------------------------------ -->
4     <!-- PARTNERLINKS                                                 -->
5     <!-- List of services participating in this BPEL process          -->
6     <!-- ------------------------------------------------------------ -->
7     <partnerLinks>
8         <!-- The 'client' role represents the requester of this service. -->
9         <partnerLink name="client" partnerLinkType="tns:HelloWorld" myRole="HelloWorldProvider"/>
10    </partnerLinks>
11    <!-- ------------------------------------------------------------ -->
12    <!-- VARIABLES                                                    -->
13    <!-- List of messages and XML documents used within this BPEL process  -->
14    <!-- ------------------------------------------------------------ -->
15    <variables>
16        <!-- Reference to the message passed as input during initiation -->
17        <variable name="input" messageType="tns:HelloWorldRequestMessage"/>
18        <!--
19          Reference to the message that will be returned to the requester
20          -->
21        <variable name="output" messageType="tns:HelloWorldResponseMessage"/>
22    </variables>
23    <!-- ------------------------------------------------------------ -->
24    <!-- ORCHESTRATION LOGIC                                          -->
25    <!-- Set of activities coordinating the flow of messages across the    -->
26    <!-- services integrated within this business process             -->
27    <!-- ------------------------------------------------------------ -->
28    <sequence name="main">
29        <!-- Receive input from requester.
30             Note: This maps to operation defined in HelloWorld.wsdl
31             -->
32        <receive name="receiveInput" partnerLink="client" portType="tns:HelloWorld" operation="process"
33        variable="input" createInstance="yes"/>
34
35        <!-- Generate reply to synchronous request -->
36        <assign name="assign-1">
37            <copy>
38                <from expression="concat(&quot;Hello &quot;, bpws:getVariableData(&quot;input&quot;,
39                                         &quot;payload&quot;, &quot;/tns:HelloWorldRequest/tns:input&quot;))">
40                </from>
41                <to variable="output" part="payload" query="/tns:HelloWorldResponse/tns:result"/>
42            </copy>
43        </assign>
44        <reply name="replyOutput" partnerLink="client" portType="tns:HelloWorld" operation="process" variable="output"/>
45    </sequence>
46 </process>
```

<process> is the top level element

<partnerLink>, channel use to interact with client (and services integrated in process)

<variable>: reference to an XML message receive or sent to the <partnerLink>s.

Process flow: sequence of activities defining the process logic.

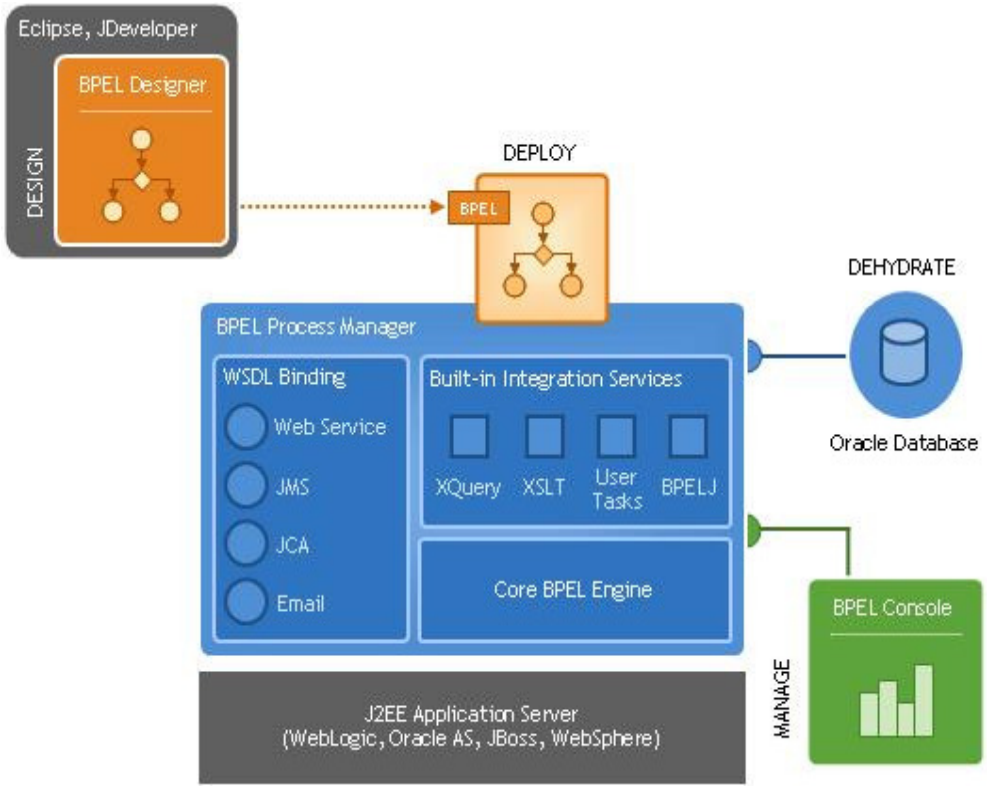
Initiate a new instance of the process when a process request is received

Create and assign greeting to output message.

Send synchronous reply to client passing output variable as response

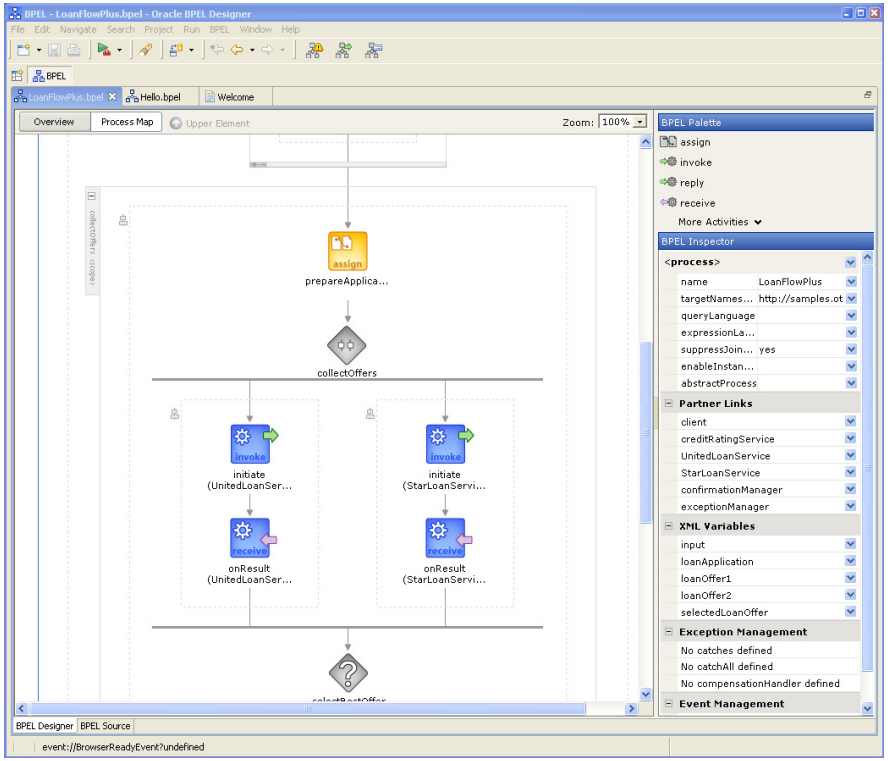
Architecture

BPEL Process Manager



- Comprehensive and native BPEL implementation
- Easy-to-use modeling tool
- Scalable and reliable engine
- Flexible binding framework
- Rich management and monitoring
- Support for Oracle AS, WebLogic and WebSphere
- Get up and running in less than 15 minutes!

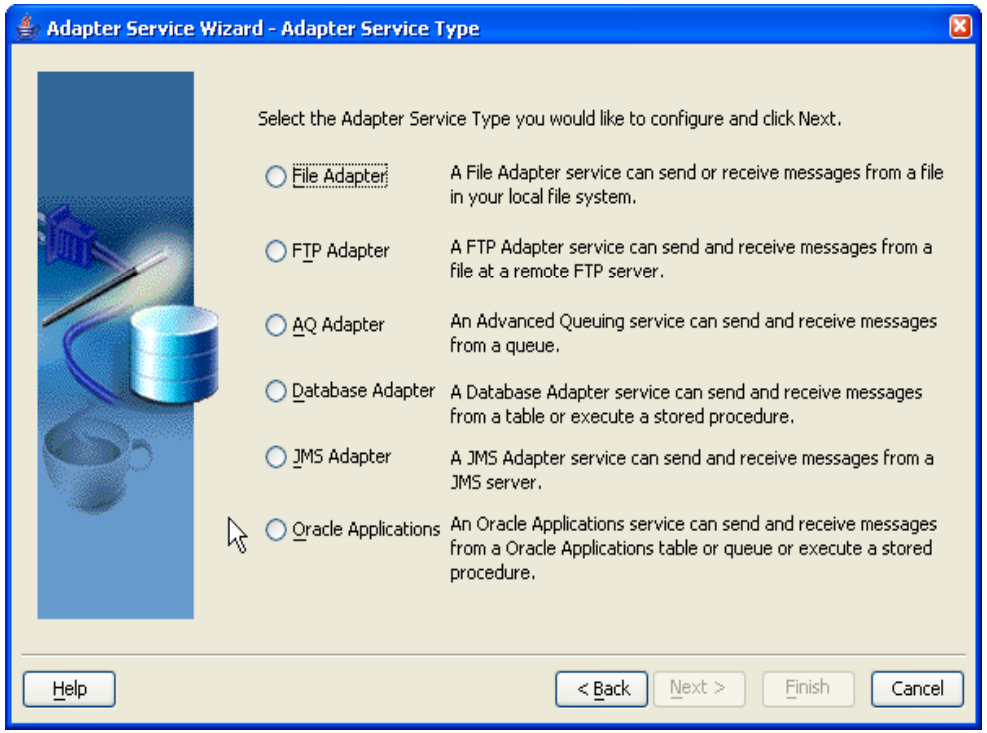
BPEL Designer



KEY FEATURES

- Native BPEL Support
- Drag-and-drop process modeler
- UDDI and WSIL service browser
- Visual XPATH editor
- Visual Assign editor
- One-click build and deploy

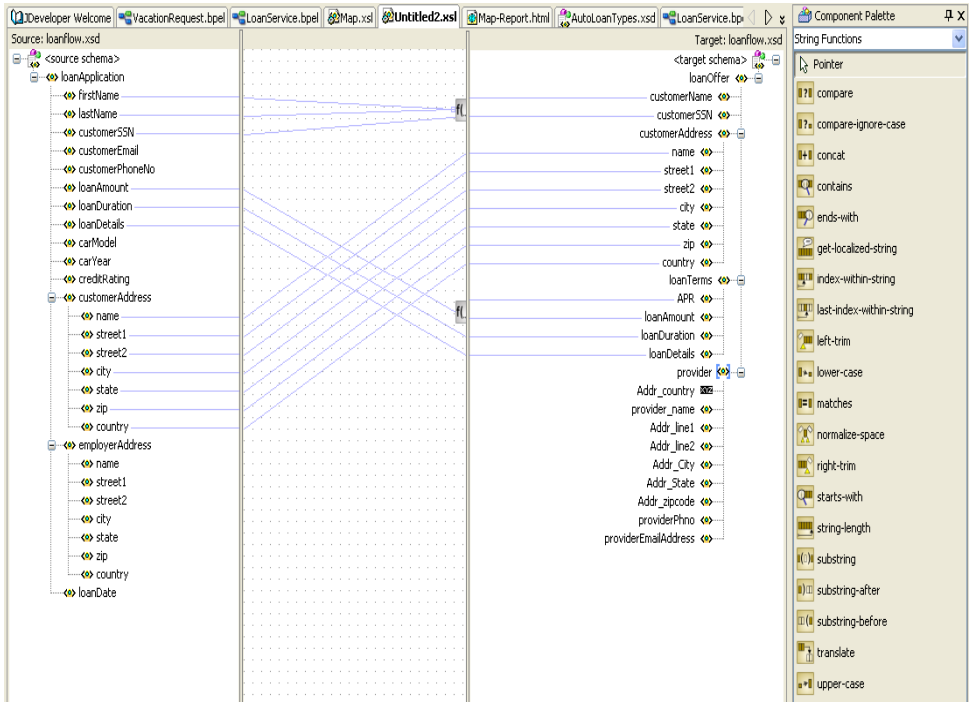
BPEL Adapters



KEY FEATURES

- WSIF + JCA + XML
- Rich Metadata
- Requests and Events
- Optimized Bindings
- 200+ Systems
- Fail Over Management

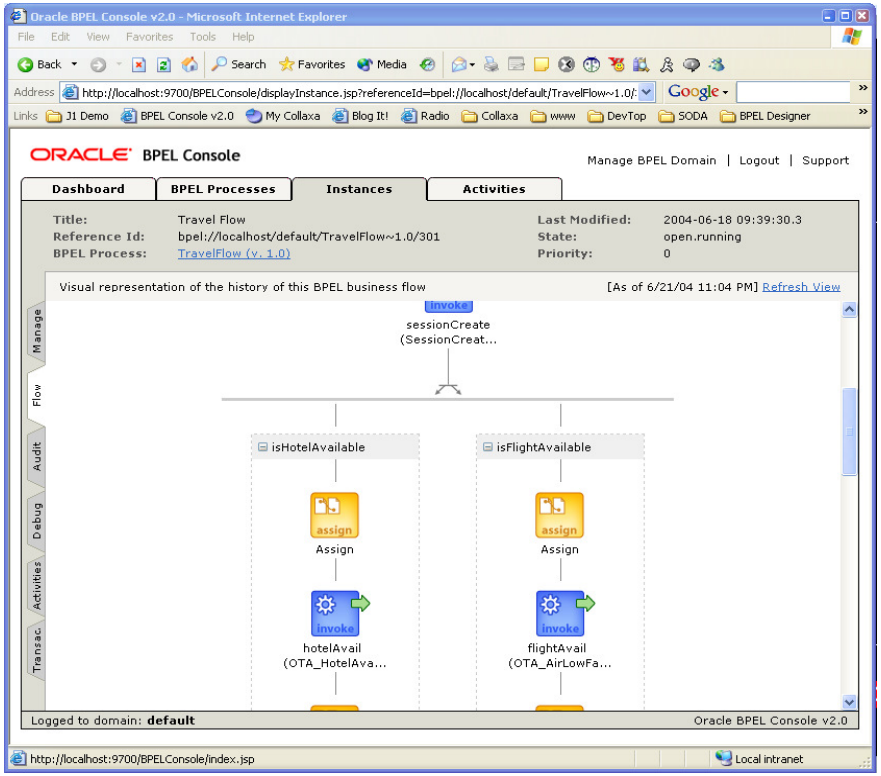
BPEL Transformations



KEY FEATURES

- Drag-and-drop Interface
- Built-in Library of Functions
- Support for Lists and Iterations
- Auto-mapping
- Version Resiliency

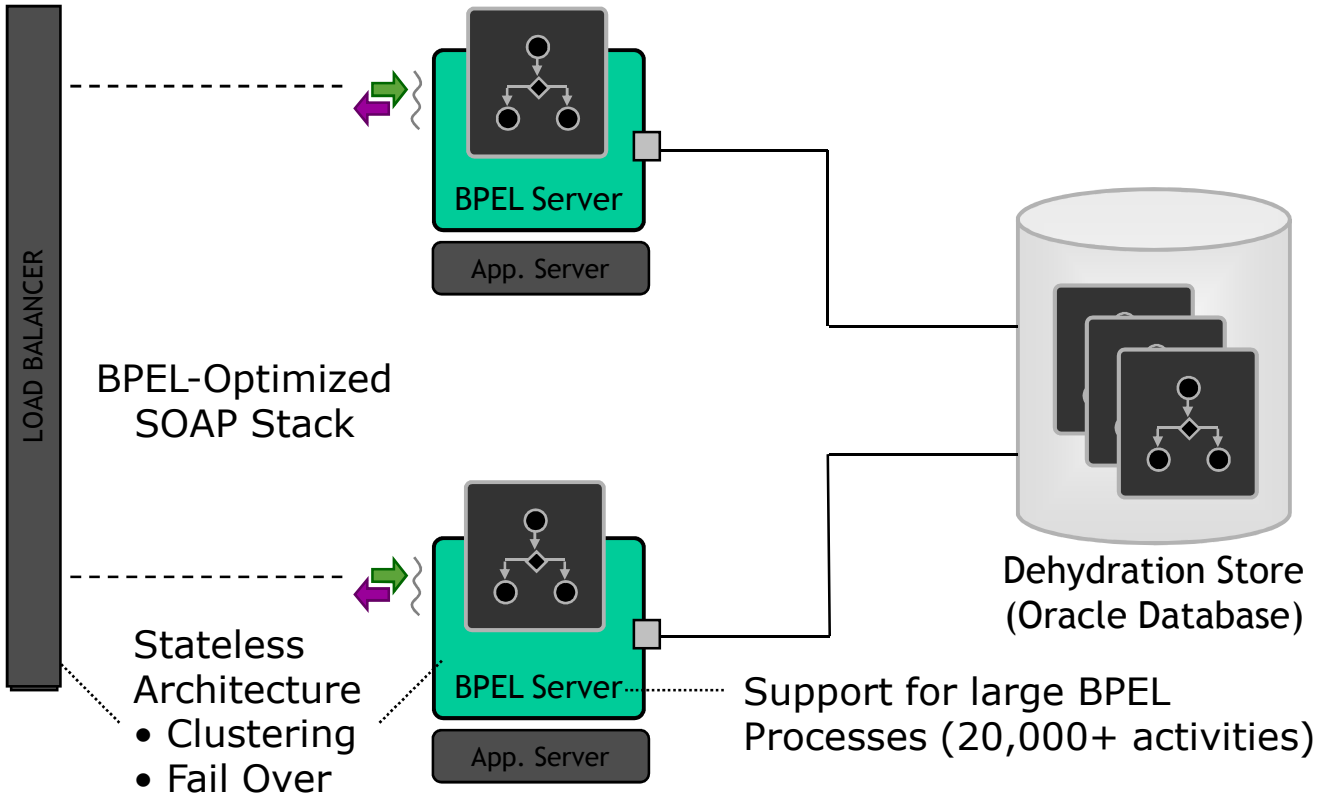
BPEL Console



KEY FEATURES

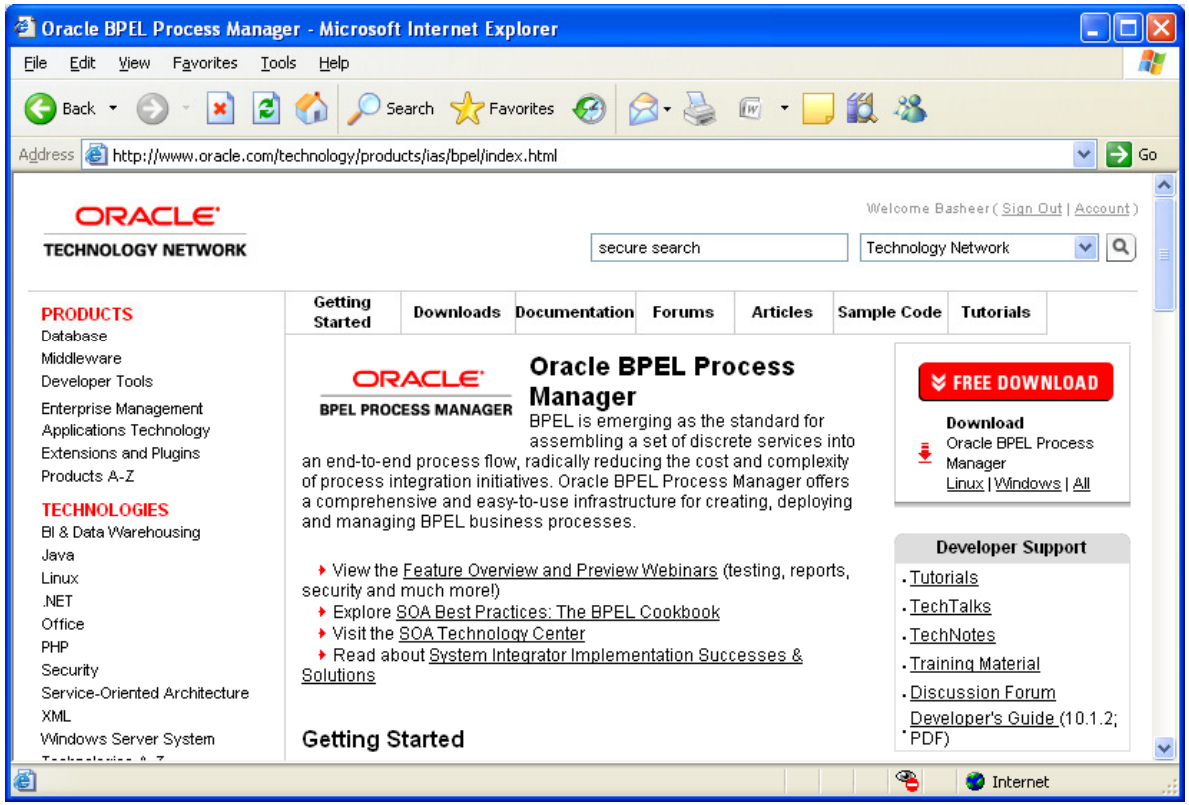
- Visual Monitoring
- Auditing
- BPEL Debugging
- In-flight Instance Administration
- Performance Tuning
- Partitioning/Domains

BPEL High Availability



Installation

BPEL Download



Hello World!

Problem/Use Case

- How do I implement, compile, deploy and run my first BPEL Process? I would like that BPEL Process to generate and return a greeting

Lab 1

- Create a new synchronous BPEL process named MyHelloWorld.
- Add “Hello World” logic to it to return the string “Hello ” plus the input string submitted to the process.

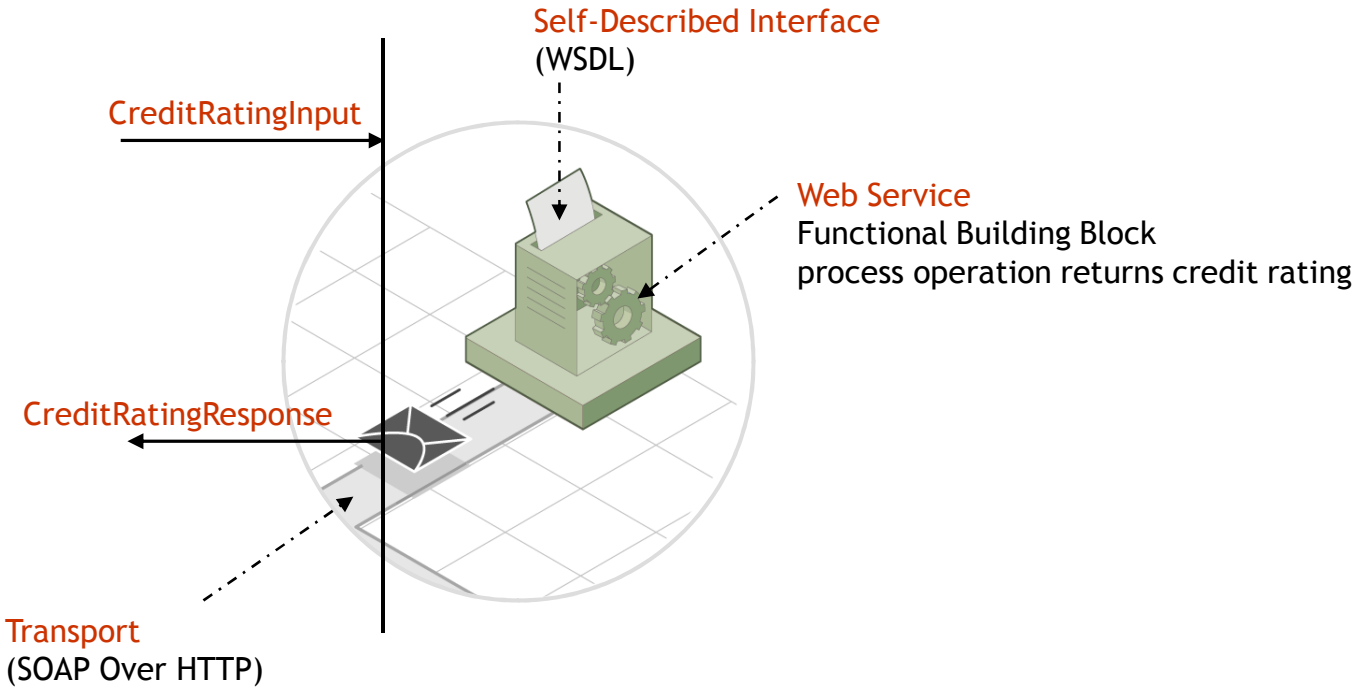
What Did We Learn?

- How to create a BPEL project (for a synchronous BPEL flow)
- How to assign a value to an XML message/variable
- How to build and deploy a BPEL process
- How to initiate and test a BPEL process

Synchronous Web Service

Problem/Use Case

- How do I invoke a synchronous credit rating web service from within a BPEL process?



Lab 2

- Create a new *asynchronous* BPEL project named MyLoanFlow.
- Change the interface to your BPEL process to accept a LoanApplication document as input and return a LoanOffer.
- Add the logic to MyLoanFlow to invoke the CreditRatingService in C:\orabpel\samples\utils\CreditRatingService.
- Make sure that the input to the credit rating service comes from your LoanApplication input variable and the credit rating response is placed back in the loan application as well.

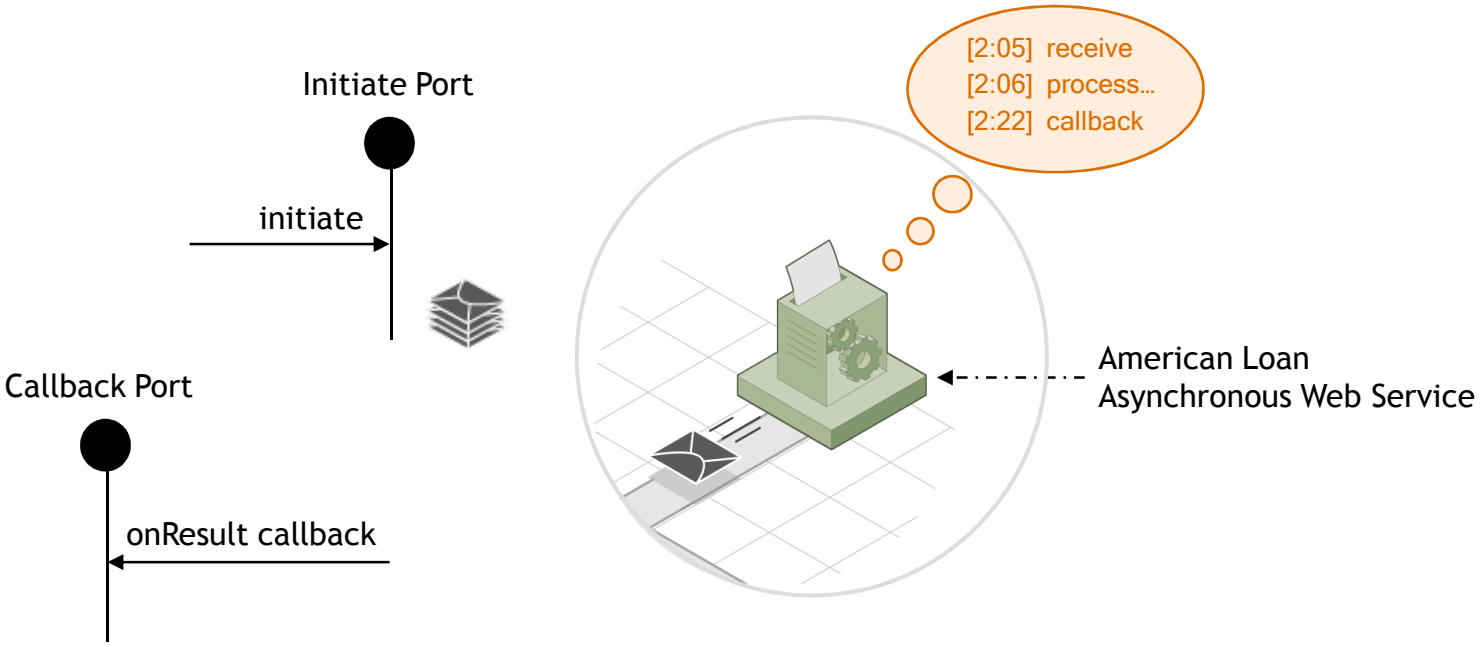
What Did We Learn?

- How to create an asynchronous BPEL process
- Changing the input and output types associated with a BPEL process
- How to add a new partnerLink to a BPEL process
- How to invoke that partnerLink from within the process flow
- How to create the variables referencing the messages that will be sent to and received from the BPEL process
- How to initialize a variable (input and output mapping)

Asynchronous Web Service

Problem/Use Case

- American Loan exposes a web service that can take anywhere from a couple of minutes to a couple days to process a loan application into a loan offer. How can I leverage that asynchronous loan processor service as part of my BPEL Process?



Lab 3

- Add the logic to your flow to invoke the asynchronous UnitedLoan service.

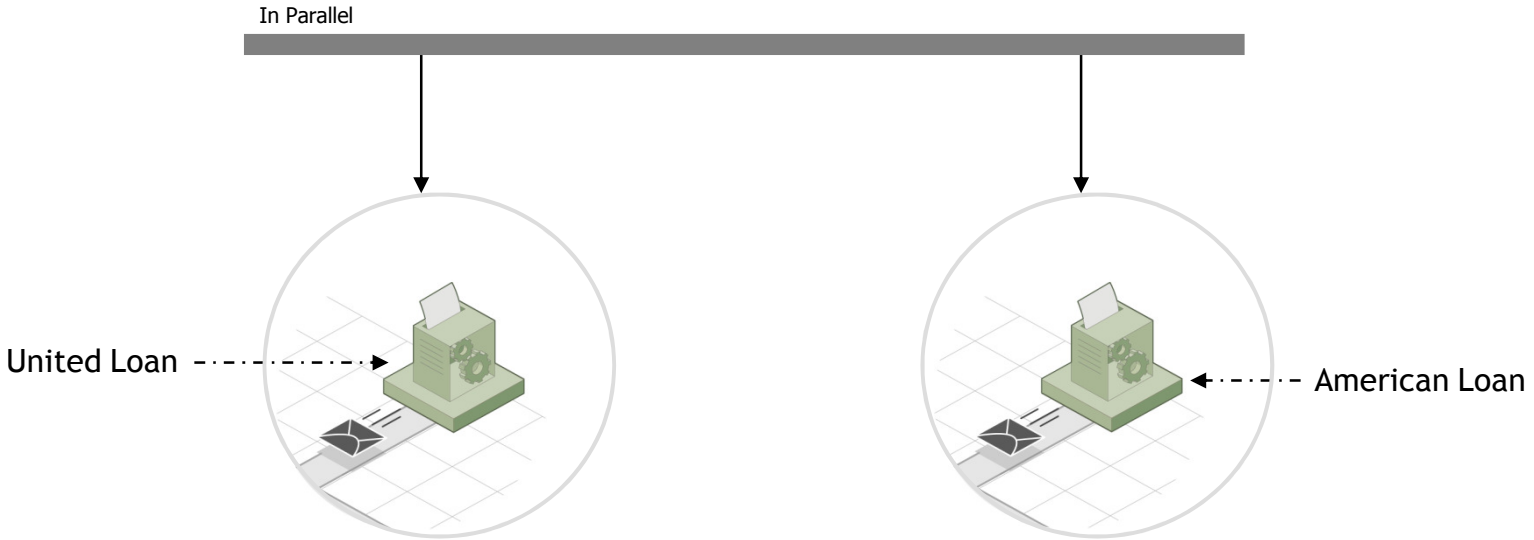
What Did We Learn?

- Asynchrony increases reliability and scalability
- How to combine two WSDL port types into an asynchronous conversation
- How to initiate an asynchronous web service (<invoke>)
- How to wait for an asynchronous web service to callback (<receive>)
- How WS-Addressing is used to exchange correlation Id and callback location information

Parallel Processing

Problem/Use Case

- Given that AmericanLoan and UnitedLoan can take up to 5 days to process a loan request, is it possible to invoke those services in parallel?



Lab 4

- Enclose your invocation of the credit rating service in a <scope> activity (just for readability, as your flow grows).
- Add a <flow> with two branches/<sequences>'s. Move your UnitedLoan invocation to one branch and add the logic to invoke the StarLoan service to the other branch.

What Did We Learn?

- How to split the flow of execution into two parallel branches

Conditional Branching

Problem/Use Case

- I have received two loan offers (one from UnitedLoan and one from StarLoan). How do I select the one with the lowest rate?

Lab 5

- Add a <switch> activity so that your process returns the better loan offer (the one with the lower APR).

What Did We Learn?

- How to add a <switch> activity to the process flow
- How to define a case using an XPATH Boolean expression

Database Connections

Problem/Use Case

- How to connected to a database?

Lab 6

- Create a new database connection

What Did We Learn?

- How to create and test a new database connection

File Load

Problem/Use Case

- How do I load data from a text file to a database table?

Lab 7

- Create a new process to read a text file and insert rows into a database table

What Did We Learn?

- How to read a text file and load it to a database table

Polling Tables

Problem/Use Case

- How do I poll a database table a read new or changed rows?

Lab 8

- Create a new process to poll a database table

What Did We Learn?

- How to poll a database table to read new or changed rows

Changing Payload

Problem/Use Case

- How do I change the default input payload

Lab 9

- Import a schema
- Replace the default payload with the new schema structure

What Did We Learn?

- How to import schemas & replace default payload

Notifications

Notifications

- Exception & Fault Handling
- Events & Timeouts

Problem/Use Case

- The Credit Rating service throws a NegativeCredit fault under certain conditions. How do I catch and manage a fault from within a BPEL process?
- StarLoan is supposed to call us back with a LoanOffer within an hour. What happens if it doesn't? How can I handle this time-out within my BPEL process?

Lab 10

- Add a <faultHandler> to your scope enclosing the credit rating service to handle the NegativeCredit faults which may be thrown by the service.
- Handle the faults in an automated fashion (here, just set the creditRating value to -1000).
- Use <pick> / <onMessage> / <onAlarm> so that your process waits a maximum of 1 minute for the StarLoan response.

What Did We Learn?

- How to wrap an <invoke> activity into a scope to be able to catch and manage a fault
- How to set the duration of a timeout

Sequences

Problem/Use Case

- How do I generate a unique sequence ID?

Lab 11

- Create a database sequence, say, ITEM_SEQ
- Add a datasource entry in data-sources.xml file

<BPEL_HOME>\integration\orabpel\system\appserver\oc4j\j2ee\home\config

```
<data-source class="com.evermind.sql.DriverManagerDataSource"
      name="BPELDemo"
      location="jdbc/BPELDemo"
      xa-location="BPELDemo"
      ejb-location="jdbc/BPELDemo"
      connection-driver="oracle.jdbc.OracleDriver"
      max-connections="50"
      min-connections="10"
      connection-retry-interval="30"
      max-connect-attempts="10"
      url="jdbc:oracle:thin:demo/demo@bkhan:1521:DEMO"/>
```

Lab 11 (cont'd)

- Stop and restart the BPEL Process Manager
- In BPEL Designer, use Database Function: sequence-next-val('ITEM_SEQ','jdbc/BPELDemo') to map the sequence to the appropriate ID column

What Did We Learn?

- How to generate unique sequence IDs

Stress Testing

Problem/Use Case

- How do I stress test my processes?

Lab 12

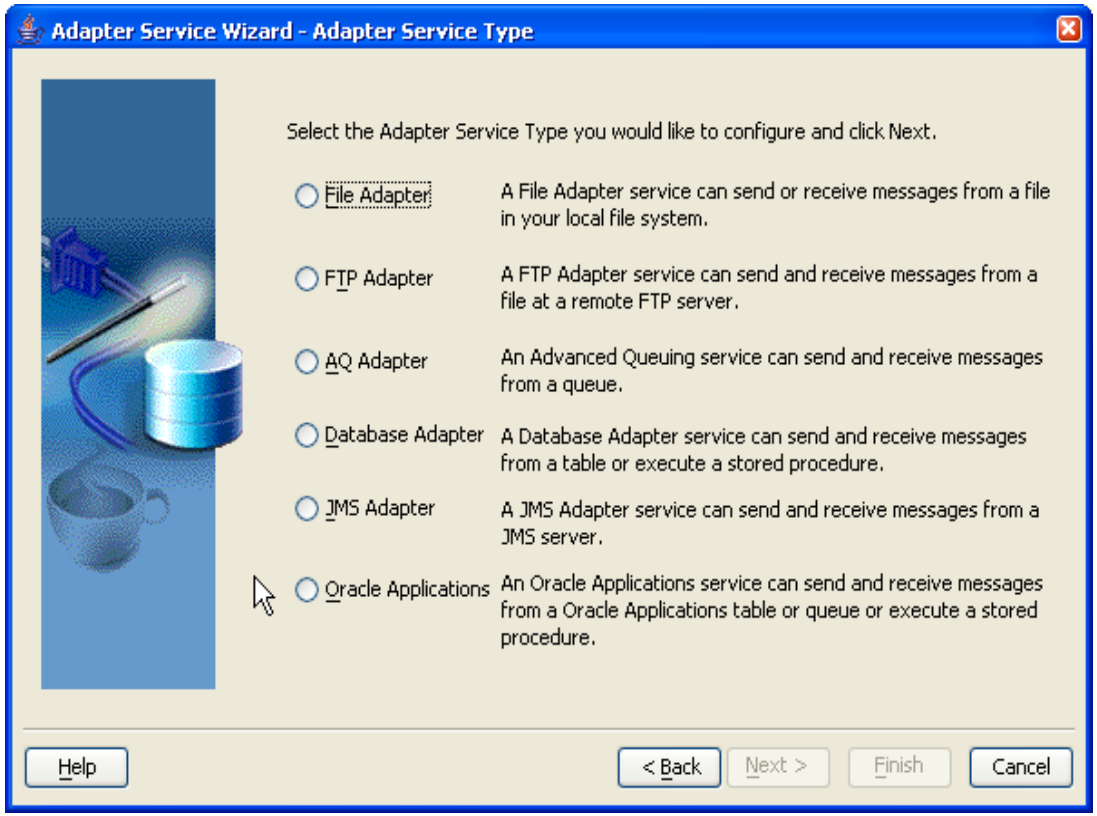
- Incorporate stress test logic in a process

What Did We Learn?

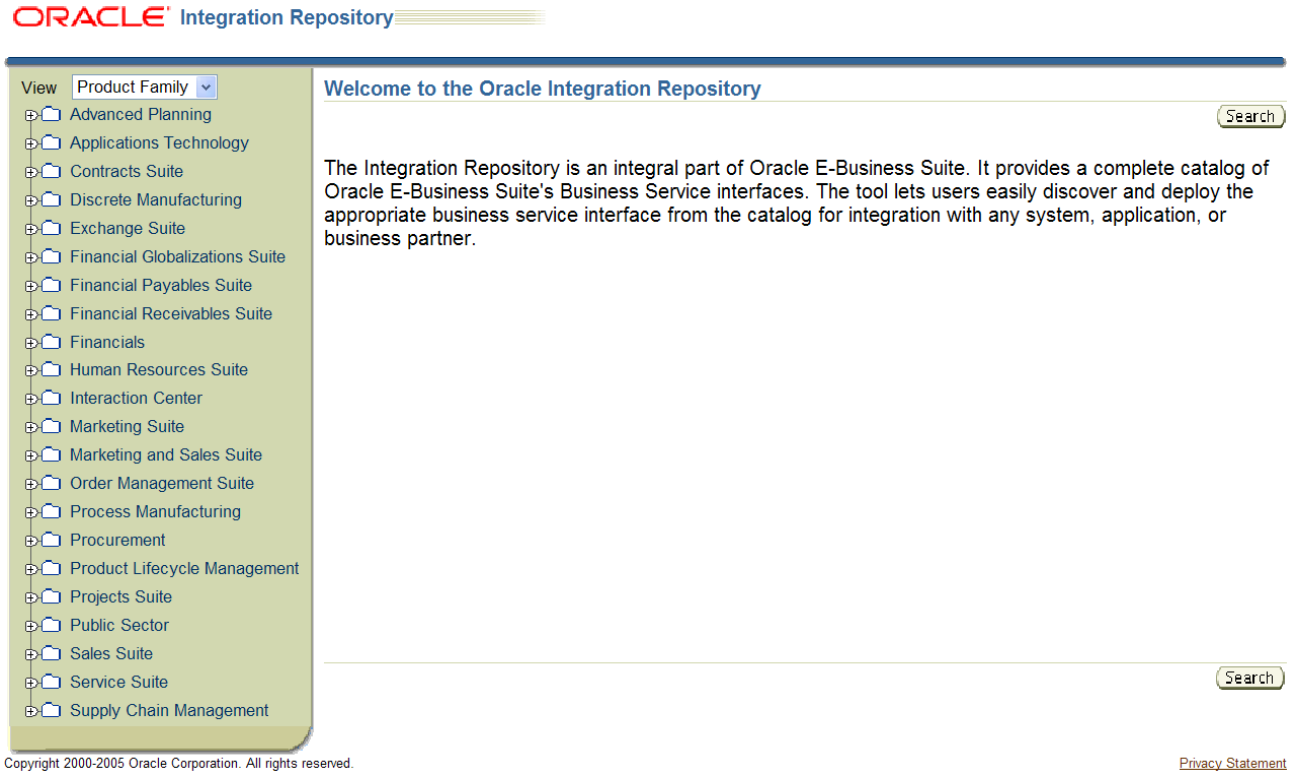
- How to stress test processes

Application Integration

Application Integration



Application Integration



http://irep.oracle.com

Application Integration

ORACLE Integration Repository

View Product Family

- Advanced Planning
- Applications Technology
- Contracts Suite
- Discrete Manufacturing
- Exchange Suite
- Financial Globalizations Suite
- Financial Payables Suite
- Financial Receivables Suite
- Financials
  - Assets
    - Asset
    - Capital Budget
    - **Payables Invoice**
  - Cash Management
  - Collections
  - Credit Management
  - Federal Financials
  - General Ledger
  - Internet Expenses
  - Latin America Localizations
  - Lease Management
  - Payables
  - Property Manager
  - Public Sector Applications
  - Public Sector Financials International
  - Receivables
  - Trading Community
  - Treasury
  - iPayment
- Human Resources Suite

Payables Invoice

Search

Export — Previous 1-10 Next 10

Name	Internal Name	Product	Type	Status	Description
Create Credit Card Issuer Invoice	APXCCINV	Internet Expenses	Open Interface	Active	The Create Credit Card Issuer Invoice program creates invoices for your credit card issuers in the Payables Open Interface tables.
Export Lease Payments to Payables	FALPEXPT	Assets	Open Interface	Active	The Export Lease Payments to Payables process exports the lease payment data to the Oracle Payables interface tables.
IN: Invoice (810/INVOIC)	AP:INI	Payables	EDI	Active	This is the source file to support the inbound Invoice transaction.
IN: Invoice (810/INVOIC)	ECEINI	Payables	Open Interface	Active	Extracts the Invoices in the EDI flat file and populates it into the Payables Open Interface tables and submits the Payables Invoice Import concurrent program.
Invoice Notification XML Message	CLN:NTFYINVCI	Supply Chain Trading Connector	Web Service	Active	This RosettaNet Invoice Notification XML Message is used to consume the invoice notification information received from the trading partner, within Oracle Accounts Payable.
Invoice Notification XML Message	CLN:NTFYINVCI	Supply Chain Trading Connector	XML Gateway Map	Active	This RosettaNet Invoice Notification XML Message is used to consume the invoice notification information received from the trading partner, within Oracle Accounts Payable.
Invoice Reject Notification XML Message	CLN:INV_REJECT_NOTIF_OUT	Supply Chain Trading Connector	XML Gateway Map	Active	This RosettaNet Invoice Reject Notification XML Message is used to publish information on a rejected invoice from the buy side instance, or Oracle Accounts Payable.
OUT: Application Advice (824/APERAK)	ECEADVO	e-Commerce Gateway	Open Interface	Active	Extracts the Application Advices to communicate the status of inbound documents back to the Trading Partner.
Pay Invoices Transfer to AP Invoice Interface	OKL_PAY_INVOICES_TRANS	Lease Management	Open Interface	Active	Executes the open interface for transferring Lease Management payable transactions and their distributions for the Payables Import Program to process.
Payables Open Interface Import	APXIIMPT	Payables	Open Interface	Active	This process executes the Open Interface for Accounts Payables Invoices.

Export — Previous 1-10 Next 10

http://irep.oracle.com

About Innowave

- Technology Consulting Firm founded in 2005
- Provide unique, innovative solutions using Oracle Applications and Technology
- Integration Experts: Fusion Middleware, SOA Suite, BPEL & Web services
- Implement, upgrade, and enhance Oracle E-Business Suite, JD Edwards, PeopleSoft, and Siebel Applications and related technologies
- Extensive experience and expertise managing global and local implementations across various industry verticals
- Provide clients with proven methodologies, tools, and templates specifically tailored to their requirements

Q & A

Basheer Khan
bk@innowavetech.com