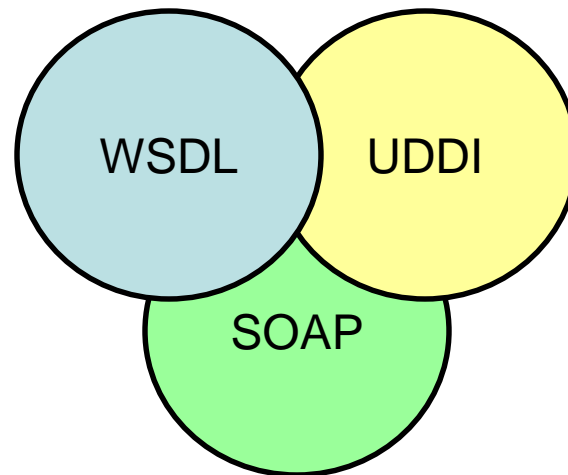


The evolution of Web Services

Magnus Larsson and Mats Ekhammar

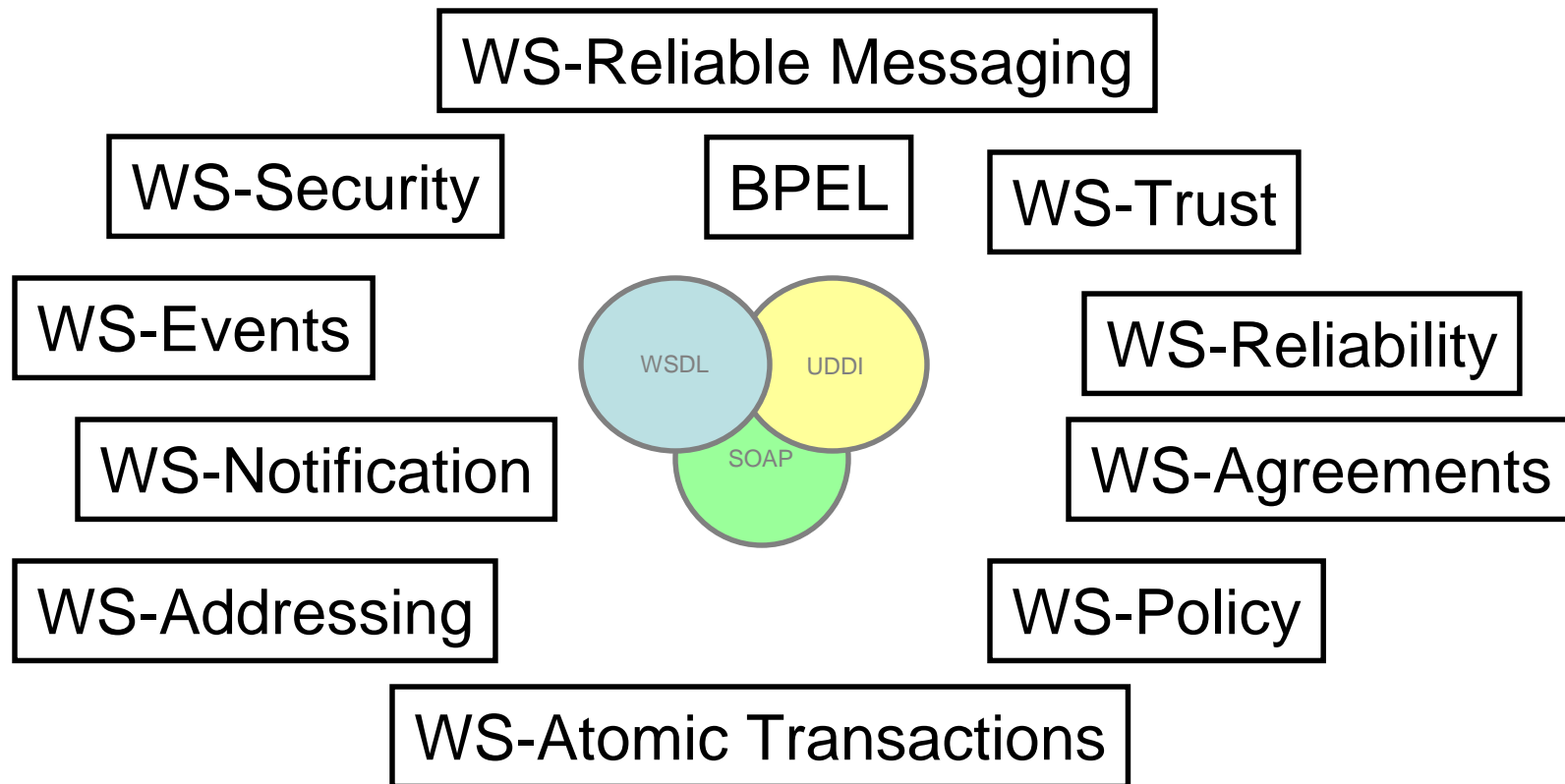
Overview

- In the beginning it was so simple...



Overview

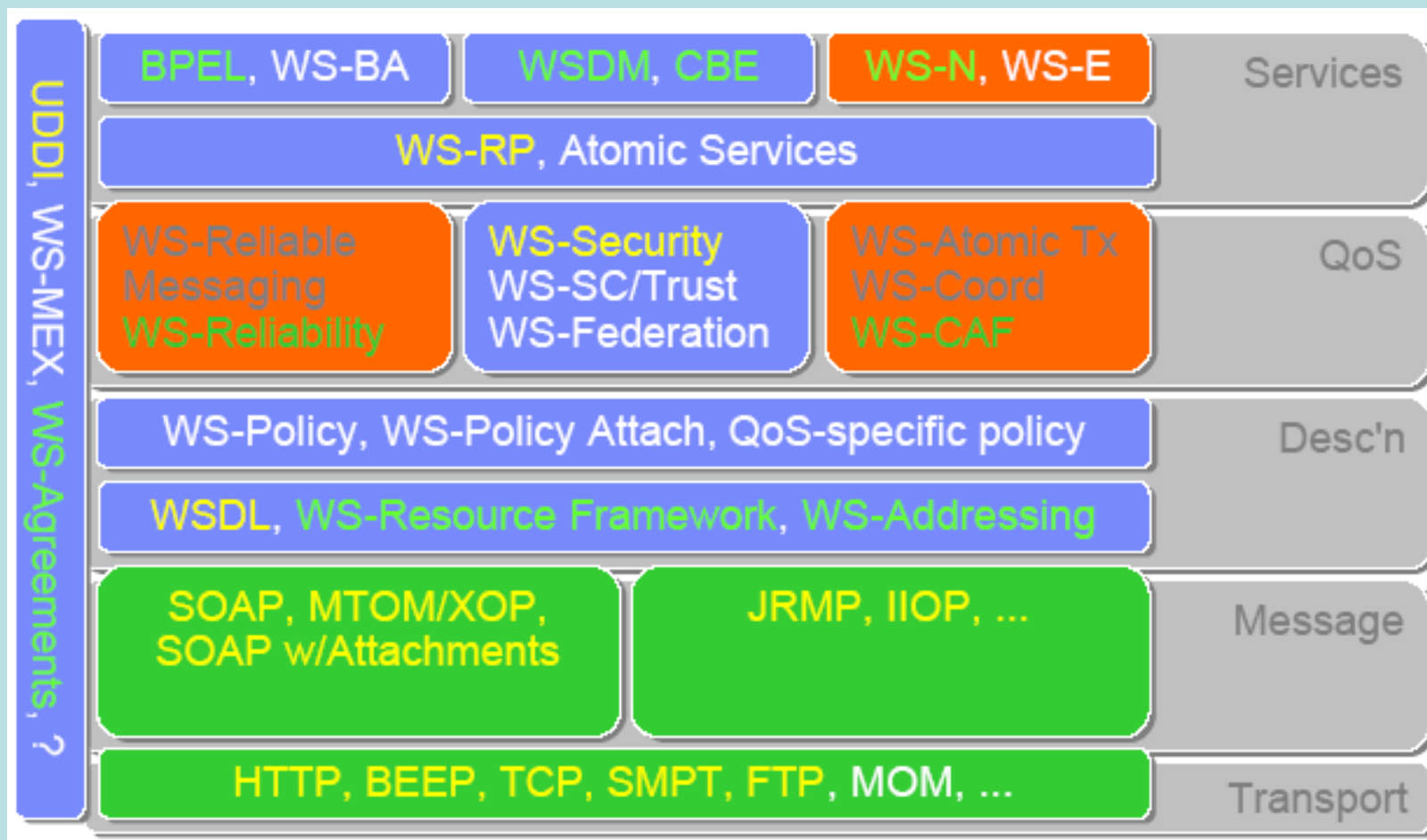
- ...then something happened



- We need a map that can bring some order!!!

The WS Stack with Specs...

WS - I



Overview

- Let's look closer on a few of the most interesting...
 - Interoperability (WS-I)
 - They better talk each other...
 - Security (WS-Security)
 - ...in a secure way...
 - Business Process Orchestration (BPEL)
 - This should really add some value...
 - ...and not just more plumbing

Interoperability

- Current problems
 - Implementation "A" can not talk to implementation "B" over Web Services...
 - Existing WS specifications leaves many open options...
 - Examples of known problems (if not only sending single strings back and forth)
 - "Non string" primitive types: Integer, Float and Date
 - Complex types: DTO, Arrays of DTO, Graphs of DTO
 - Null values
 - Binding and Use
 - Document vs. RPC
 - Encoded vs. Literal (with and w/o Wrapped)
 - Exceptions
 - Attachments

Interoperability

- WS-I (Web Services Interoperability) to the rescue...
 - www.ws-i.org
 - Overview
 - Details usage of HTTP, SOAP, WSDL, XML Schema, UDDI
 - E.g. Binding: Document/Literal
 - Usage of datatypes as defined by “[XML Schema Part 2: Datatypes](#)”
 - Current Version
 - Basic Profile v1.0 (BP 1.0)
 - HTTP 1.1
 - SOAP 1.1
 - WSDL 1.1
 - XML 1.0
 - XML Schema Part 1 “Structures” and Part 2 “Datatypes”
 - UDDI v2.0

Interoperability

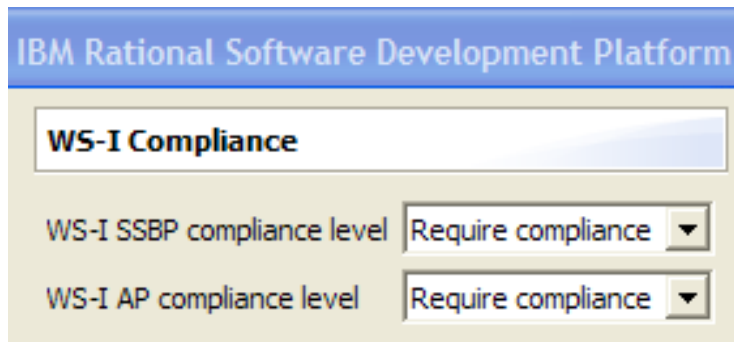
- WS-I to the rescue...
 - Next Version:
 - Basic Profile v1.1 (BP 1.1)
 - Include "Binding Profile 1.0" and "Attachments Profile 1.0"
 - "Security Profile" on its way...
- Vendor support (examples)
 - BP 1.0
 - Microsoft .NET 2.0
 - J2EE v1.4 servers (e.g. IBM WebSphere 6.0 and JBoss 4.0)
 - Apache Axis v1.2 (Currently only in RC2 - stage)
 - BP 1.1
 - Sun JWSDP 1.4

Interoperability

- **WS-I in Microsoft .NET 2.0**

```
[WebServiceBinding(  
    ConformanceClaims=WsiClaims.BP10,  
    EmitConformanceClaims = true  
)]  
public class Service : System.Web.Services.WebService {
```

- **WS-I in IBM Rational SDP and WebSphere 6.0**



- **WS-I in Sun JWSDP 1.4**

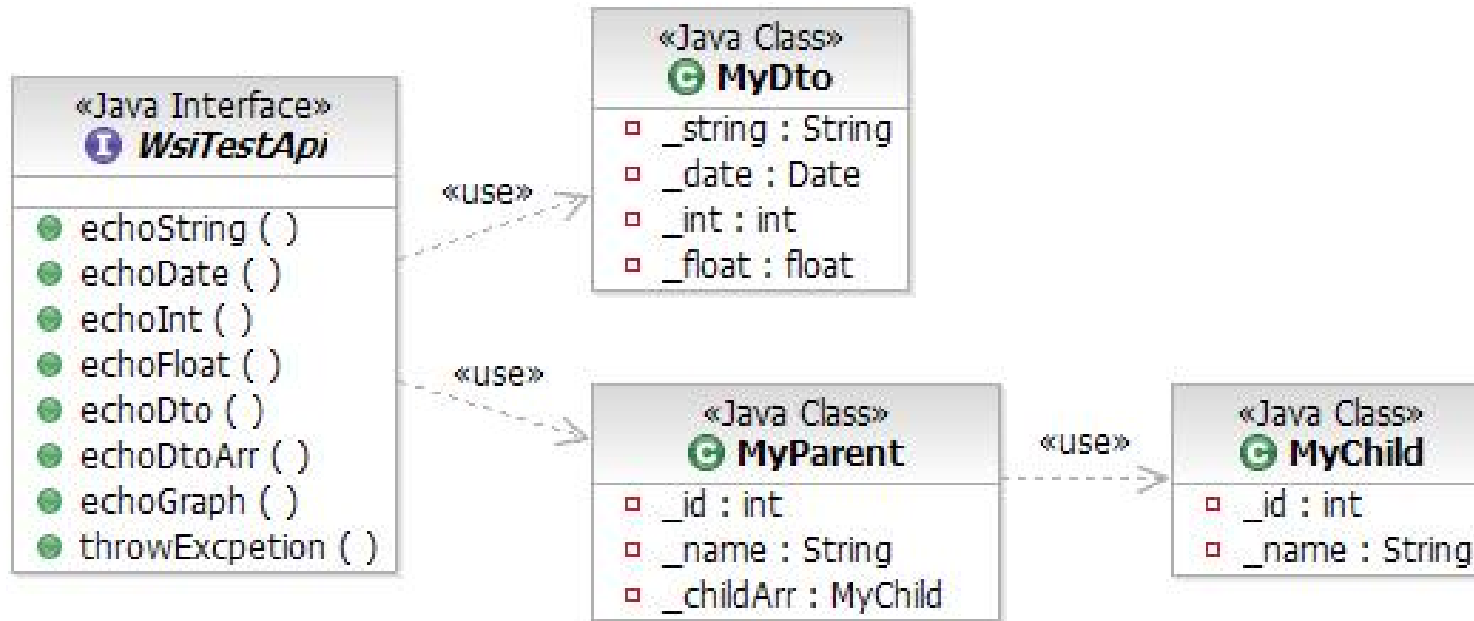
```
wscompile -f:wsi ...
```

Interoperability - Demo

- WS-I BP 1.0
 - Server
 - J2EE 1.4 server – WebSphere 6.0
 - IBM Rational Software Development Platform 6.0
 - Client
 - Microsoft .NET 2.0
 - Visual Studio 2005 Team System

Interoperability - Demo

- Web Services Operations



Interoperability - Summary

- WS-I Basic Profile 1.0 is a major step forward!
 - The list of known problems addressed by BP1.0!
 - Except for attachments, see below...
- What is still missing in WS-I BP 1.0 and 1.1?
 - Attachments...
 - WS-I BP 1.1 contain "Attachments Profile 1.0"
 - Based on "SOAP Messages with attachments"
 - Supported by Java but not by Microsoft .NET...
 - » Microsoft .NET supports "WS Attachments"
 - Work is ongoing with new standards: "MTOM" and "XOP"
 - Only old "Base64 encoding" will do as of today...
- Recommendation
 - Use WS-I BP Compliant tools now whenever possible
 - Use "Base64 encoding" for attachments for now...

Security

- A Web Service must be able to **handle**
 1. **Identification**
 2. **Integrity**
 3. **Confidentiality**
- A Web Service must be able to **support**
(I.e. supplying the service implementation with authentication tokens)
 1. **Authentication**
 2. **Authorization**
 3. **Auditing**
 4. **Non-repudiation**

Security

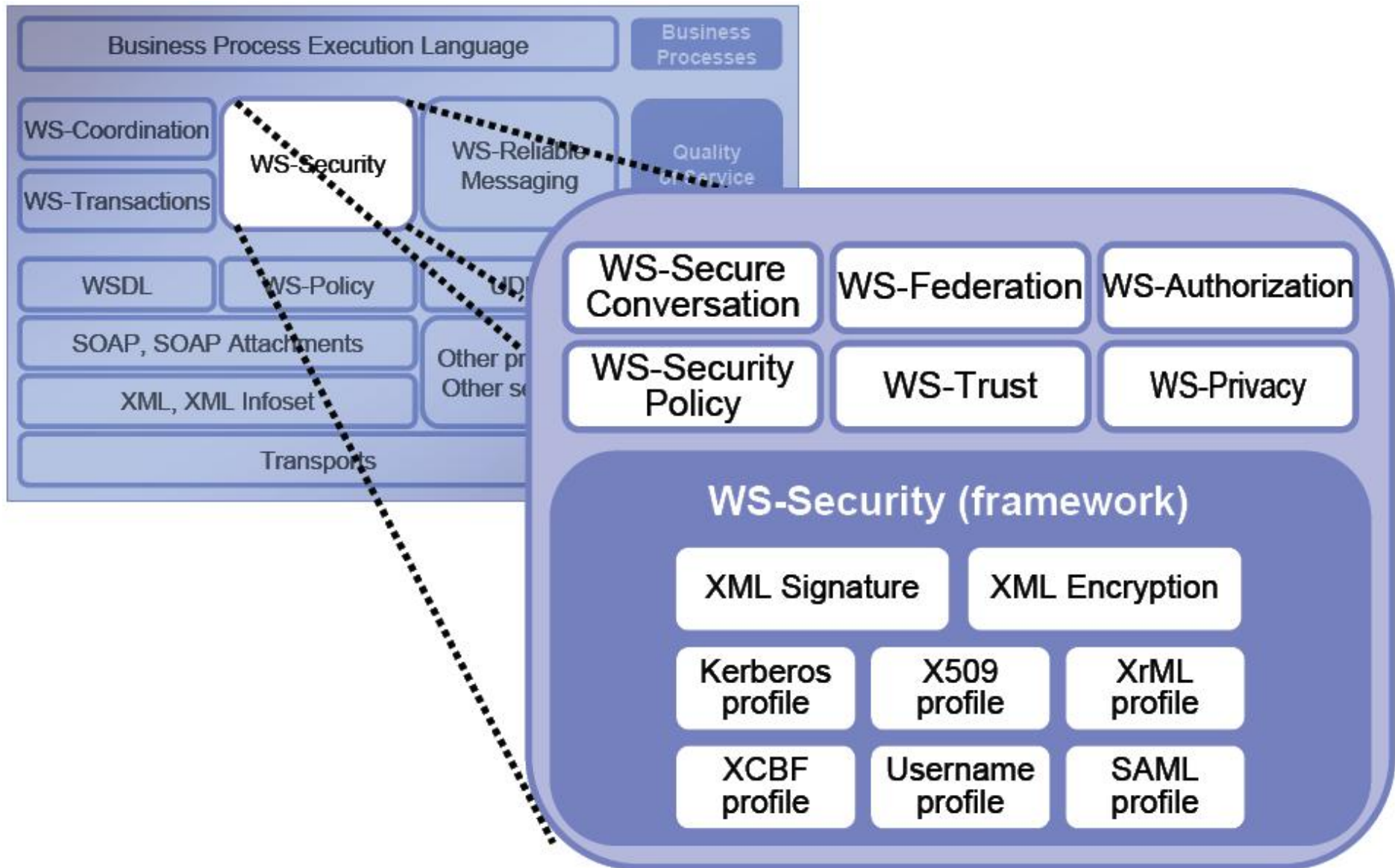
- Current problems
 - Today we secure our Web Services with SSL/HTTPS
 - SSL/HTTP only gives security on the protocol level...
 - We need security on application level
 - Multiple hops
 - Middle servers might not be allowed to read the content
 - HTTP is maybe not the protocol in all hops
 - Encryption for Confidentiality cost performance
 - In most cases we only need to encrypt parts of a message
 - Authentication, Authorization, Auditing and Non-repudiation need access to the authentication tokens
 - Different parts of a message might need different Digital Signatures

Security

- **WS-Security to the rescue...**
 - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
 - Overview
 - **Identification through Authentication Tokens**
 - Username/Password and X509 Certificates
 - **Integrity through XML Digital Signing**
 - **Confidentiality through XML Encryption**
 - **Infrastructure**
 - Asymmetric Private/Public Keys + CA
 - Current Version
 - WS-Security 1.0 released by OASIS in April 2004
 - Coming versions
 - WS-I “Security Profile 1.0” is on its way...

Security

WS – Security is not a single specification...



Security

- Vendor support (examples)
 - Microsoft .NET 1.1 + WSE 2.0
 - Sun JWSDP 1.4
 - IBM WebSphere 6.0
- Limited Java vendor support
 - J2EE does currently not support WS-Security
 - Apache WSS4J (Security for Axis) is a new project with as of today no deliverables

Security

- XML Digital Signature
 - Stored in a **<Signature>** - element in the SOAP - Header
 - It has three main parts
 - **<SignedInfo>** – Information about what is signed
 - **<SignatureValue>** – The value of the digital signature itself
 - **<KeyInfo>** – The public key used to verify the signature

Security

XML Digital Signature Example

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
```

```
<SignedInfo>
```

What is signed?

```
<CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
```

```
<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
```

```
<Reference URI="#wssecurity_body_id_2601212934311668096_1040651106378">
```

```
<Transforms>
```

```
<Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
```

```
</Transforms>
```

```
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
```

```
<DigestValue>AWQKpmksMpzzT4PxcizO980gVHw=</DigestValue>
```

```
</Reference>
```

```
</SignedInfo>
```

Signature Value

```
<SignatureValue>bNhT+DsNN9PR [binary data has been truncated]</SignatureValue>
```

```
<KeyInfo>
```

Public Key (optional)

```
<wsse:SecurityTokenReference>
```

```
<wsse:Reference URI="#wssecurity_binary_security_token_id_1603091_4272645" />
```

```
</wsse:SecurityTokenReference>
```

```
</KeyInfo>
```

```
</Signature>
```

Signature Block

Security

- XML Encryption
 - Replace the encrypted information with a **<EncryptedData>** - element
 - **<EncryptionMethod>**
 - The algorithm used to encrypt the data
 - **<KeyInfo>**
 - Information about the key used to encrypt the data
 - **<CipherData>**
 - **<CipherValue>**
 - » Contains the actual encrypted data

Security

- Encryption Example: Only a part of the message encrypted

```
<PayBalanceDue xmlns='http://example.org/paymentv2' >
  <Name>John Smith</Name>
  <CreditCard Limit='5,000' Currency='USD' >
    <Number>4019 2445 0277 5567</Number>
    <Issuer>Bank of the Internet</Issuer>
    <Expiration>04/02</Expiration>
  </CreditCard>
</PayBalanceDue >
```

Red text is data to be encrypted
Green text is left unencrypted

```
<PayBalanceDue xmlns='http://example.org/paymentv2' >
  <Name>John Smith</Name>
  <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
    Type='http://www.isi.edu/in-notes/iana/assignments/media-types/text/xml' >
    <CipherData><CipherValue>A23B4C6</CipherValue></CipherData>
  </EncryptedData>
</PayBalanceDue >
```

<CreditCard> group was replaced
by <EncryptedData> element

Security - Demo

- With WS-Security
 - Environment
 - Server: Apache Tomcat 5.0 + Sun JWSDP 1.4
 - Client: Microsoft .NET 1.1 + WSE 2.0
 - Tests
 - SOAP Message without security
 - SOAP Message with XML Signature
 - SOAP Message with XML Encryption
 - Security is declared in WSE 2.0 using WS Policy

Security - Summary

- How far does WS-Security take us today?
 - It works, but...
 - It takes a lot of detailed knowledge to get .NET and Java to interoperate over WS-Security
 - Encryption algorithms
 - Certificate protocols
 - Certificate handling
 - We need a WS-I “Security Profile”!
 - No J2EE-spec support today
 - Limited J2EE vendor support
- Recommendation
 - If HTTPS/SSL doesn't meet your security requirements look into WS-Security before you start to develop your own digital signing and encryption...

BPEL

BPEL stands for **B**usiness **P**rocess
Execution **L**anguage.

- It is used for composing multiple web services into process flows (orchestration).



- BPEL is a cornerstone of SOA (**S**ervice **O**riented **A**rchitecture)

BPEL

- A BPEL process is called from a client as a normal Web Service.
- The process can be synchronous or asynchronous.
- The process flow is described with XML
 - The resulting flow file will be a BPEL WSDL file.
 - BPEL tools provide graphical flow programming.
 - Based on WSDL, XML Schema, XPath and WS-Addressing.

BPEL

Some BPEL operations:

- <invoke> calls another web service
- <receive> asynchronous callbacks
- <assign> set/get parameter values
- <flow> parallel execution
- <faultHandlers> catches and manages exceptions
- ...and many more





BPEL

Anatomy of a Web Service call

- Define a **partnerlink**
- Add an **invoke**
 - Creates **variables**
- Add two **assigns**
- Define a **scope**

BPEL - Partnerlink

Standard WSDL elements

```
<wsdl:message name="HelloCADECSoapIn">
  <wsdl:part name="parameters" element="tns:HelloCADEC" />
</wsdl:message>
<wsdl:message name="HelloCADECSoapOut">
  <wsdl:part name="parameters"
    element="tns:HelloCADECResponse" />
</wsdl:message>

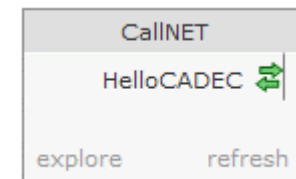
<wsdl:portType name="Service1Soap">
  <wsdl:operation name="HelloCADEC">
    <wsdl:input message="tns:HelloCADECSoapIn" />
    <wsdl:output message="tns:HelloCADECSoapOut" />
  </wsdl:operation>
</wsdl:portType>
```

BPEL – Partnerlink cont.

BPEL WSDL elements

```
<plnk:partnerLinkType
  name="Service1SoapLink">
  <plnk:role name="Service1SoapProvider">
    <plnk:portType name="tns:Service1Soap" />
  </plnk:role>
</plnk:partnerLinkType>
```

```
<partnerLink
  partnerLinkType="ns0:Service1SoapLink"
  partnerRole="Service1SoapProvider"
  name="CallNET"
/>
```



BPEL - Invoke

Invoke will make a call to a Web Service

```
<invoke  
  portType="ns0:Service1 Soap"  
  name="TestInvoke"  
  partnerLink="CallNET"  
  operation="HelloCADEC"  
  inputVariable="inCallNet"  
  outputVariable="outCallNet"  
>
```



BPEL - Assign

Assign will copy data to the request message and from the response message

```
<assign xmlns="http://s..."
  name="assignBefore">
  <copy>
    <from variable="input" part="payload"
      query="/tns:AsynkTestRequest/tns:input" />
    <to variable="inCallNet" part="parameters"
      query="/ns0:HelloCADEC/ns0:name" />
  </copy>
</assign>
```



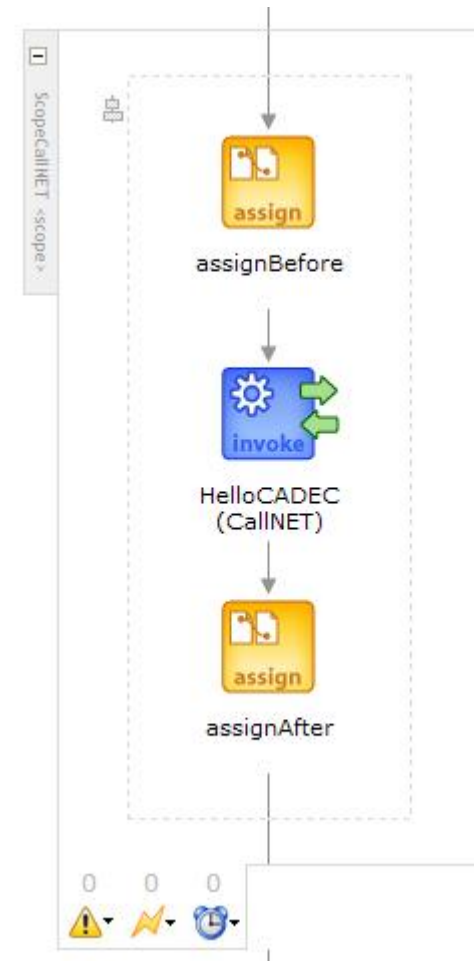
We will often use variables defined in our BPEL process

```
<variable
  messageType="ns0:HelloCADECSoapIn"
  name="inCallNet"
/>
```

BPEL - Scope

Inside a scope we can define local variables, fault handlers, compensation handlers, ...

```
<scope name="ScopeCallNET" >  
  ...  
</scope>
```



BPEL – Demo I

First demo shows how to call a webservice deployed in .Net

We will create a new Web Service in .NET and then create a BPEL process that uses this process.

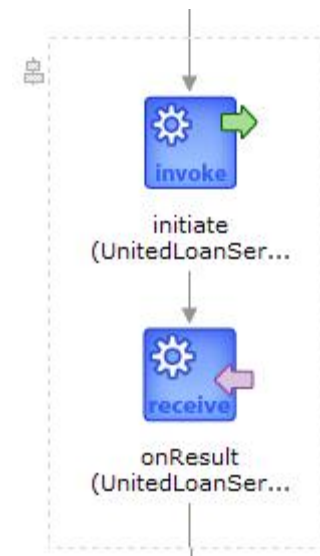
Demodiving!

BPEL

An asynchronous call consists of one invoke and one receive activity.

Invoke will call the Web Service as before

Receive will get the callback from the service called in invoke.





BPEL

Other Language constructs

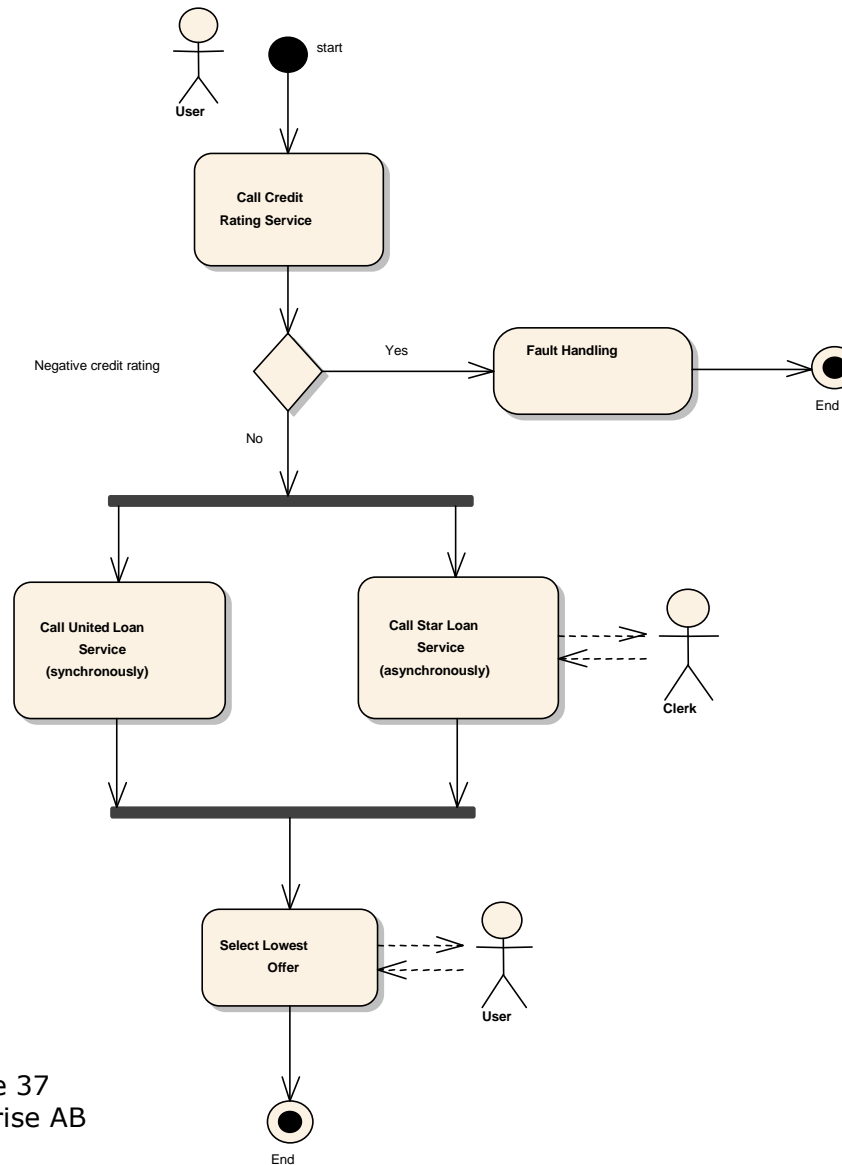
- Correlation Sets
- Fault Handlers
- Compensation Handlers
- Sequence, flow
- Switch, while, pick
- Link
- ...

BPEL – demo II

Second demo will show a more complex process flow.

This demo requires some manual interaction before the flow can complete.

BPEL - Demo



BPEL - Links

- Specification

- <http://www-128.ibm.com/developerworks/webservices/library/ws-bpel/index.html>

- Oracle

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

- IBM

- <http://www-130.ibm.com/developerworks/webservices>

- Microsoft

- <http://www.microsoft.com/biztalk>

BPEL - Summary

- Try it !
- It is very easy to use from the beginning.
- BUT ! Everything depends of the services that you wish to call.
- If you dont have a consistent service architecture, BPEL won't save your day!

Questions?

