Web Services Security
and Interoperability

Bjarne Rasmussen
Manager, Messaging Technology
Novell, Inc.

Vijay Rajan
Senior Engineer
Novell, Inc.

Novell

The one Net vision

Novell exteNd™ Novell Nsure™ Novell Nterprise™ Novell Ngage℠

one Net: *Information without boundaries…where the right people are connected with the right information at the right time to make the right decisions.*

© April 25, 2005 Novell Inc.

Novell

The one Net vision

Novell exteNd™ | Novell Nsure™ | Novell Nterprise™ | Novell Ngage℠

Novell exteNd™

Novell exteNd solutions provide you with a range of advanced Web application development tools to liberate information and create more effective business solutions for your customers, partners, and employees.

© April 25, 2005 Novell Inc.

Novell.

Agenda

- Brief Introduction to Web Services
- Web Services Security
- WS-I (Web Services Interoperability) Organization
- WS-I Sample Application & Demo
- Summary

© April 25, 2005 Novell Inc.

Novell.

Brief Introduction to
Web Services

Brief Introduction to Web Services

What it is ?

- Basically a transparent and flexible mechanism to perform Remote Procedure Calls and Message Passing

Differentiation from previous technologies

- Internet based (i.e. accessible through standard-based Internet protocols such as HTTP)
- Language & platform independent
- Based on open standards
- Firewall-friendly
- Inherently loosely coupled (suitable for SOA)
- Based on standardized (XML) as opposed to proprietary binary standards

© April 25, 2005 Novell Inc.

Characteristics of Web Service

WSDL (Web Services Description Language) to describe the interface to a service

UDDI (Universal Description, Discovery and Integration of Web Services) to advertise and lookup services

SOAP (Simple Object Access Protocol) used for message passing

HTTP transport to carry the SOAP messages

Key to Web Services is the unification of the industry around these standards

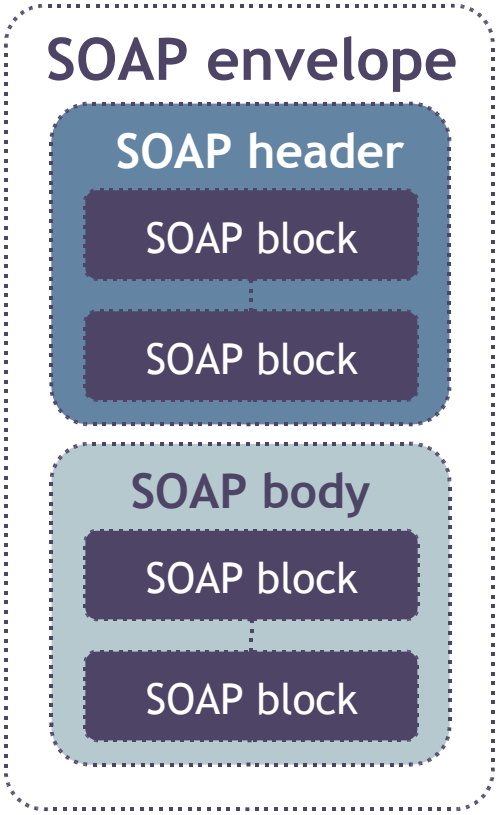
© April 25, 2005 Novell Inc.

Extensible Markup Language (XML)

- Cornerstone of technology to permit interoperable communication
- Ideal for representing data
- Generic and language neutral way to represent structured and typed data
- XML incorporated into several networking protocols in order to provide a standard way to applications to communicate – one such example is SOAP
- WSDL also based on XML
- Simple and extensible mechanism to represent complex documents

© April 25, 2005 Novell Inc.

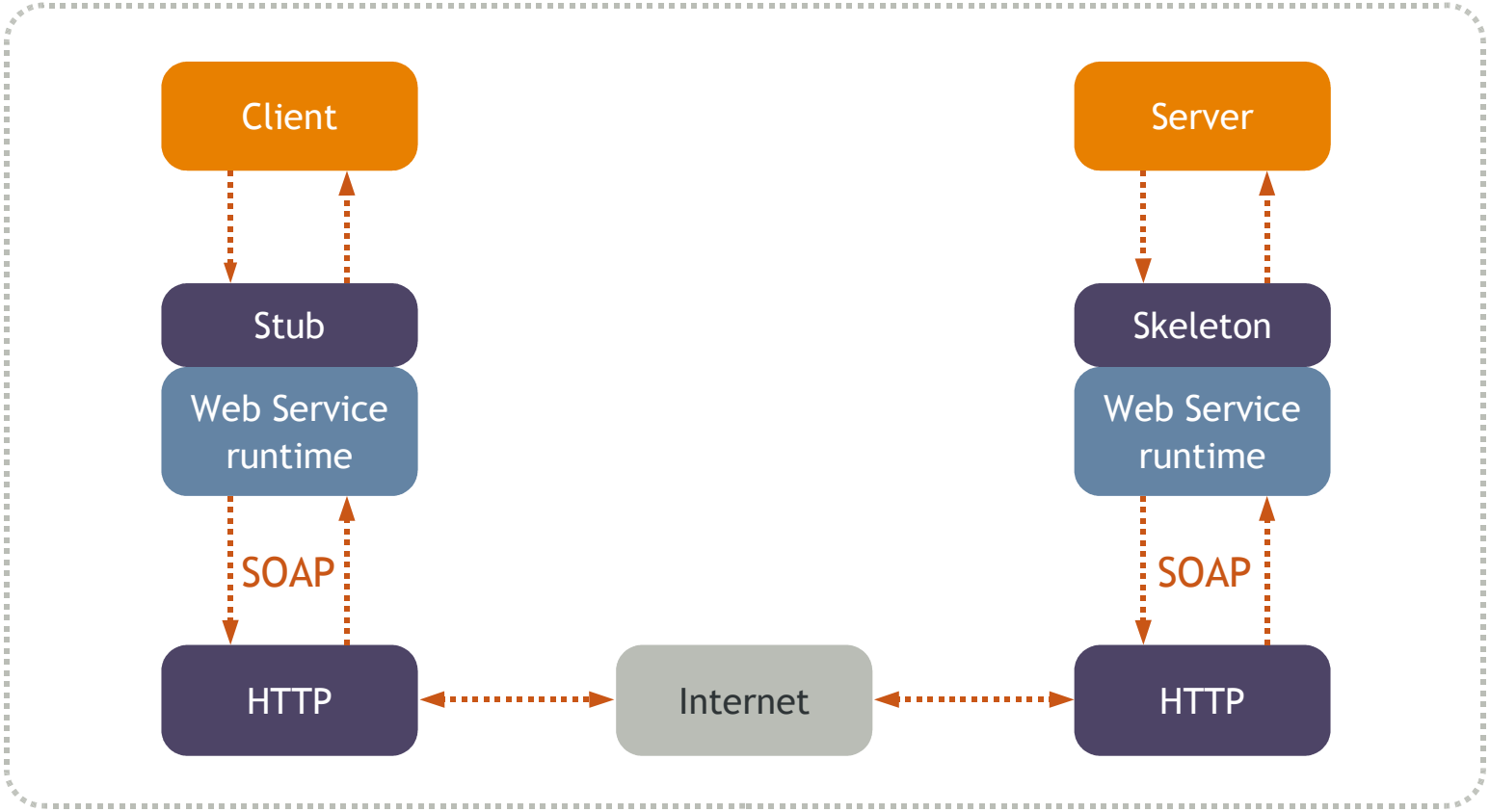
Simple Object Access Protocol (SOAP)




```
<SOAP-ENV:Envelope
xmlns:SOAPENV="http://schemas.xmlsoap.o
rg/soap/envelope/">

  <SOAP-ENV:Header>
    ...
  </SOAP-ENV:Header>

  <SOAP-ENV:Body>
   <ns1:sayHello
xmlns:ns1="http://www.hello"
SOAPENV:encodingStyle="http://schemas.x
mlsoap.org/soap/encoding/"/>
  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```


© April 25, 2005 Novell Inc.

Web Service Stack



© April 25, 2005 Novell Inc.

Novell.

RPC vs Document Style

- RPC mechanism provides a way to represent a remote procedure call and its return value
- In RPC style the SOAP Body contains a single SOAP struct
- The direct child of the Body contains the method name
- The method element contains the parameters in the same order as the method signature
- The parameters are accessible via an accessor (either a tag or an original value for arrays)
- Document style does not contain the method name or parameter accessors

© April 25, 2005 Novell Inc.

Novell.

Encoded vs Literal

- Encoding rules provide a convention for serializing datatypes into a XML representation
- soap:encodingStyle attribute is used to indicate the encoding rules used in the message
- http://schemas.xmlsoap.org/soap/encoding/ refers to rules based on the schema for SOAP 1.1
- In the literal case messages are an instance of the XML schema and look exactly as described in the schema.
- Due to the complexity created by encoding, WS-I Basic Profile prefers the use of literal, non-encoded XML

© April 25, 2005 Novell Inc.

Security

Security Needs

Web Service Infrastructure

- Authentication
- Data Integrity
- Data Confidentiality

Combining Identity and Web Services

- Authentication
- Authorization and Access Control
- Single sign-on
- Identity

© April 25, 2005 Novell Inc.

Authentication

Ensure identity of the user or service

- Various authentication mechanisms
- Mature industry standards exist
- HTTP Authentication
  - Basic
  - Digest
  - NTLM
- Secure Socket Layer (SSL)
- UsernameToken (OASIS WS-Security)

© April 25, 2005 Novell Inc.

Novell.

Data Integrity

Ensure that information has not been modified

Can also be used to identify sender of information

- Secure Socket Layer (SSL)
  - Only point to point
  - Cannot be applied to parts of the information
  - Difficult for workflow applications, SOAP intermediaries etc.
  - No security for persistent data
  - Introduces transport dependency
- XML Digital Signature - W3C
- WS-Security (Web Services Security) – OASIS
- WS-I Basic Security Profile 1.0 (work in progress)

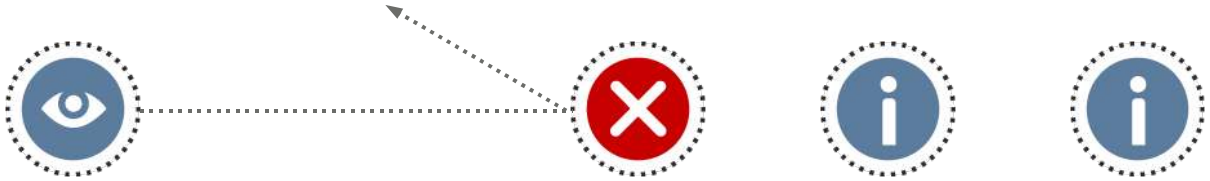
© April 25, 2005 Novell Inc.

Novell.

Data Confidentiality

Protect information from unauthorized viewing

- Secure Socket Layer (SSL)
  - Same issues as in data integrity
- XML Encryption - W3C
- WS-Security (Web Services Security) – OASIS
- Symmetric and Public Key Encryption
- WS-I Basic Security Profile 1.0 (work in progress)



© April 25, 2005 Novell Inc.

WSSDK APIs for Authentication

stub.setProperty(PROPERTY_NAME, value);

PROPERTY_NAME

- javax.xml.rpc.Stub.USERNAME_PROPERTY
- javax.xml.rpc.Stub.PASSWORD_PROPERTY
- com.sssw.jbroker.web.portable.Stub.AUTH_SCHEME
  - BASIC_AUTH_SCHEME
  - DIGEST_AUTH_SCHEME
  - NTLM_AUTH_SCHEME
- com.sssw.jbroker.web.portable.Stub.NTLM_HOST
- com.sssw.jbroker.web.portable.Stub.NTLM_DOMAIN

© April 25, 2005 Novell Inc.

Basic Authentication

Stub properties

```
// set basic authentication properties
stub.setProperty(USERNAME_PROPERTY, "username");
stub.setProperty(PASSWORD_PROPERTY, "password");
```


© April 25, 2005 Novell Inc.

Novell.

Basic Authentication: HTTP Header

```
POST /services/hello HTTP/1.1
Content-Type: text/xml; charset="utf-8"
Content-Length: 297
soapAction: ""
Authorization: Basic dG9tY2F0OnRvbWNhdA==
User-Agent: Jakarta Commons-HttpClient/2.0rc1
Host: localhost:80

<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
   ......
</SOAP-ENV:Envelope>
```


© April 25, 2005 Novell Inc.

Digest Authentication

Stub properties

```
// set digest authentication properties
stub.setProperty(USERNAME_PROPERTY, "username");
stub.setProperty(PASSWORD_PROPERTY, "password");
stub.setProperty(AUTH_SCHEME, DIGEST_AUTH_SCHEME);
```


© April 25, 2005 Novell Inc.

NTLM Authentication

Stub properties

```
// set NTLM authentication properties
stub.setProperty(USERNAME_PROPERTY, "username");
stub.setProperty(PASSWORD_PROPERTY, "password");
stub.setProperty(AUTH_SCHEME, NTLM_AUTH_SCHEME);
stub.setProperty(NTLM_HOST, "host");
stub.setProperty(NTLM_DOMAIN, "host");
```


© April 25, 2005 Novell Inc.

Public Key Cryptography

Encryption/Decryption Keys used in pairs

- One key used to encrypt (transform) the data
- Only the other key can decrypt the data

Public/Private Key pair

- Computationally infeasible to derive one key from the other

Digital Certificates

- Binding between public key and its owner

© April 25, 2005 Novell Inc.

WSSDK SSL Support

SSL support provided through stub

Set properties for various SSL parameters

- Certificates
- Private key to be used for encryption

SSL transport handler takes care of negotiation between client and server

© April 25, 2005 Novell Inc.

Novell.

WSSDK SSL Configuration

Stub properties

```
// set certificate chain
File certFile = new File("clientCertChain.der");
byte[][] certChain = loadChainFromFile(certFile);
stub.setProperty(SSLProperties.WSSDK_SSL_CERT_CHAIN, certChain);

// set private key
File keyFile = new File("clientKey.pkcs8");
byte[] prvKey = loadFromFile(keyFile);
stub.setProperty(SSLProperties.WSSDK_SSL_PRIVATE_KEY, prvKey);

// set pass phrase
stub.setProperty(SSLProperties.WSSDK_SSL_PASS_PHRASE, "password");

// set root CAs
byte[][] rootCAs = getRootCAs();
stub.setProperty(SSLProperties.WSSDK_SSL_ROOT_CA_CERTS, rootCAs);
```


© April 25, 2005 Novell Inc.

Novell.

Digital Signature

First Step is to generate a “hash”

- For message M, generate H(M)

Need to prevent H(M) from being modified

- HMAC (Hashed Message Authentication Code)
- Public Key Cryptography

Send M and {H(M)}private-key (Signature)

- Receiver computes H'(M) and checks if it matches with decrypted Signature

© April 25, 2005 Novell Inc.

WS Security Header

```
<SOAP-ENV:Envelope ...>
 <SOAP-ENV:Header>
    ....
    <wsse:Security SOAP-ENV:mustUnderstand="1"
xmlns:wsse"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
       ....
    </wsse:Security>
    .....
 </SOAP-ENV:Header>
 ......
</SOAP-ENV:Envelope>
```


© April 25, 2005 Novell Inc.

WS Security Continued....

```
<wsse:Security SOAP-ENV:mustUnderstand="1" xmlns:wsse"...">
  <wsse:UsernameToken use:Id="....">
    <wsse:Username>....</wsse:Username>
  </wsse:UsernameToken>

  <wsse:BinarySecurityToken EncodingType=".." ValueType="..">
    ........
  </wsse:BinarySecurityToken>

  <ds:Signature ... xmlns:ds=".....>
   .........
  </ds:Signature>

  <xenc:EncryptedKey ... xmlns:xenc=".....>
   .........
  </xenc:EncryptedKey>
</wsse:Security>
```


© April 25, 2005 Novell Inc.

WS Security Continued....

```
<wsse:BinarySecurityToken xmlns:wsu="..." ValueType="..."
EncodingType="....." wsu:Id="CertId-1454099">
MIICYDCCAcmgAwIBAgIQNHArdwcE/7xw9tt+unwB2zANBgkqhkiG9w0BAQUFADBDM
</wsse:BinarySecurityToken>



<wsse:SecurityTokenReference xmlns:wsu="...">
    <wsse:Reference URI="#CertId-1454099"/>
</wsse:SecurityTokenReference>
```


© April 25, 2005 Novell Inc.

Novell.

WSSDK APIs for WSS Signatures

Stub properties

```
// set this property to sign outgoing messages
stub.setProperty(XMLSignatureProperties.SIGN_OUTGOING,
    Boolean.TRUE);

// set this property to validate incoming messages
stub.setProperty(XMLSignatureProperties.VALIDATE_INCOMING,
    Boolean.TRUE);

// set key store for certificates
KeyStore ks = KeyStore.getInstance("JKS");

// ...

// signature method
stub.setProperty(XMLSignatureProperties.SIGNATURE_METHOD_URI,
    XMLSignatureProperties.ALGO_ID_SIGNATURE_RSA);
```


© April 25, 2005 Novell Inc.

Novell.

APIs for WSS Signatures Cont'd …

Stub properties

```
// digest method
stub.setProperty(XMLSignatureProperties.DIGEST_URI,
    XMLSignatureProperties.ALGO_ID_DIGEST_SHA1);

// transforms
String[] tsfms = new String[] {
    XMLSignatureProperties.TRANSFORM_ENVELOPED_SIGNATURE,
};
stub.setProperty(XMLSignatureProperties.TRANSFORM_URIS, tsfms);

// canonicalization method
stub.setProperty(XMLSignatureProperties.CANONICALIZATION_METHOD_URI,
    XMLSignatureProperties.ALGO_ID_C14N_OMIT_COMMENTS);
```


© April 25, 2005 Novell Inc.

APIs for WSS Signatures Cont'd …

Stub properties

```
// key info certificate
X509Certificate cert = (X509Certificate) ks.getCertificate("test");
stub.setProperty(XMLSignatureProperties.CERTIFICATE,
    cert.getEncoded());

// validation public key
stub.setProperty(XMLSignatureProperties.VALIDATION_PUBLIC_KEY,
    cert.getPublicKey().getEncoded());

// signing key
PrivateKey privateKey = (PrivateKey) ks.getKey("test", secret);
byte[] prvKeyBytes = privateKey.getEncoded();
stub.setProperty(XMLSignatureProperties.PRIVATE_KEY, prvKeyBytes);
```


© April 25, 2005 Novell Inc.

SOAP Message with WSS Signature

```
<SOAP-ENV:Envelope ...>
 <SOAP-ENV:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
     <ds:CanonicalizationMethod Algorithm="http://...."/>
     <ds:SignatureMethod Algorithm="http://...."/>
     <ds:Reference URI="#id-32486590">
       <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/...."/>
       </ds:Transforms>
       <ds:DigestMethod Algorithm="http://www.w3.org/...."/>
       <ds:DigestValue>w22BXO84rYix5VdLJhh7+v67KpU=</ds:DigestValue>
      </ds:Reference>
     </ds:SignedInfo>
```


© April 25, 2005 Novell Inc.

Message with WSS Signature Cont'd ...

```
    <ds:SignatureValue>
J01xJ/X6zTzamy/5UvcqLY+Jcia/D5QDm7o23Shtm3Vr+IaiJKG3QWDrM/WSe
vk2P8YTkbyRr0ZalVKbh+uRqWq1yf1suBdTfEMXjwXsxEbprVP5r7RXt9Rq8GtAPr9
Kgus6dqOXUaa32Je565ftPFgKrnww=
    </ds:SignatureValue>
    <ds:KeyInfo>
     <wsse:SecurityTokenReference ...>
      <wsse:Reference URI="#CertId-1454099"/>
     </wsse:SecurityTokenReference>
    </ds:KeyInfo>
   </ds:Signature>
  </wsse:Security>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body xmlns:wsu="..." wsu:Id="id-32486590">
  ...
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


© April 25, 2005 Novell Inc.

Novell.

Encryption – Public, private & secret

Asymmetric Key - using Public and Private Keys

- Can only encrypt data of limited size

Symmetric Key

- Need to have a common shared Key
- Faster performance, no size limit

Asymmetric encryption to send secret Key

- Generate a random secret Key and encrypt using receiver's public Key.

© April 25, 2005 Novell Inc.

SOAP Message with WSS Encryption

```
<SOAP-ENV:Envelope ...>
 <SOAP-ENV:Header>
  <wsse:Security ... xmlns:wsse="...">
   <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/....">
    <xenc:EncryptionMethod Algorithm="http://....#rsa-1_5"/>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
     <wsse:SecurityTokenReference>
      <wsse:KeyIdentifier ..>lE5aEvMfbq8V62s=</wsse:KeyIdentifier>
     </wsse:SecurityTokenReference>
    </ds:KeyInfo>
    <xenc:CipherData>
     <xenc:CipherValue>LjZiNGjx1EEzGxQ+WIS7G</xenc:CipherValue>
    </xenc:CipherData>
    <xenc:ReferenceList>
      <xenc:DataReference URI="#EncDataId-3089045"/>
    </xenc:ReferenceList>
   </xenc:EncryptedKey>
  </wsse:Security>
```


© April 25, 2005 Novell Inc.

Novell.

Message with WSS Encryption Cont'd ..

```
 </SOAP-ENV:Header>
 <SOAP-ENV:Body xmlns:wsu="...." wsu:Id="id-3089045">
  <xenc:EncryptedData Id="EncDataId-3089045" xmlns:xenc="..."
Type="http://www.w3.org/2001/04/xmlenc#Content">
   <xenc:EncryptionMethod Algorithm="http://...#aes128-cbc"/>
    <xenc:CipherData>
     <xenc:CipherValue>
TpS3kJ+NIMa7c+vMjAQamaASBKxeRCdLB3tGn7yCp1AD8Z7h0tVJ3bAcCRtyg
Bf3dSCLNgAiQ7QdbUCtJMABlqQoGNrjbyR5HTjD1n7MXVxR1fhjuTMfgiZMTc
80wr3GXCyyBF27yte7YCPOn6tDcFVCWeYexL65TGQ86rcx2Do15fTfer0nmr6
KmJG7aP6pbEeUnlh9DKkzkFV1EbvKnwXTAp1QEvkLNi6+kCKpIRX12YCijMG/
5d6IL14hbrpZTr8uxcCOopkOKd6+YOd4WIDu2R+OWv3E/M4VbC4abcVAUDDC8
     </xenc:CipherValue>
    </xenc:CipherData>
   </xenc:EncryptedData>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


© April 25, 2005 Novell Inc.

Web Services Interoperability

Web Services Interoperability (WS-I)
Organization

Open Industry effort to promote Web Services interoperability across platforms, applications, and programming languages

Members include software vendors, system integrators, enterprise customers, and many others

Provide guidance, recommended practices, and support resources for developing interoperable Web Services

Not a standards organization

Guidance, clarity, best-practices based on industry standards, such as XML, SOAP, UDDI, WSDL, etc.

© April 25, 2005 Novell Inc.

WS-I Deliverables

Profiles – Sets of specifications based on Web Services standards to simplify implementation and promote interoperability

Sample Implementations – Application to exercise main features of the profile & demonstrate interoperability

Implementation Guidelines – Recommendations for use of specifications in ways proven to be most interoperable

Sniffer – monitor and log interactions with Web Services

Analyzer – Process sniffer logs and verify that the Web Services comply with the profile

© April 25, 2005 Novell Inc.

WS-I Basic Profile 1.0

Set of non-proprietary Web services specifications, along with clarifications and amendments of those specifications which promote interoperability

- Impossible to completely guarantee interoperability
- Address the most common problems
- Testing achieved in a non-intrusive manner (by examining artifacts - WSDL, SOAP Messages)
- Only addresses inconsistencies and design flaws in referenced specifications that affect interoperability
- Profile's scope is bounded by the specifications referenced by it
- Scope is further refined by extensibility points

© April 25, 2005 Novell Inc.

WS-I Sample Application 1.0

Supply Chain Management Application

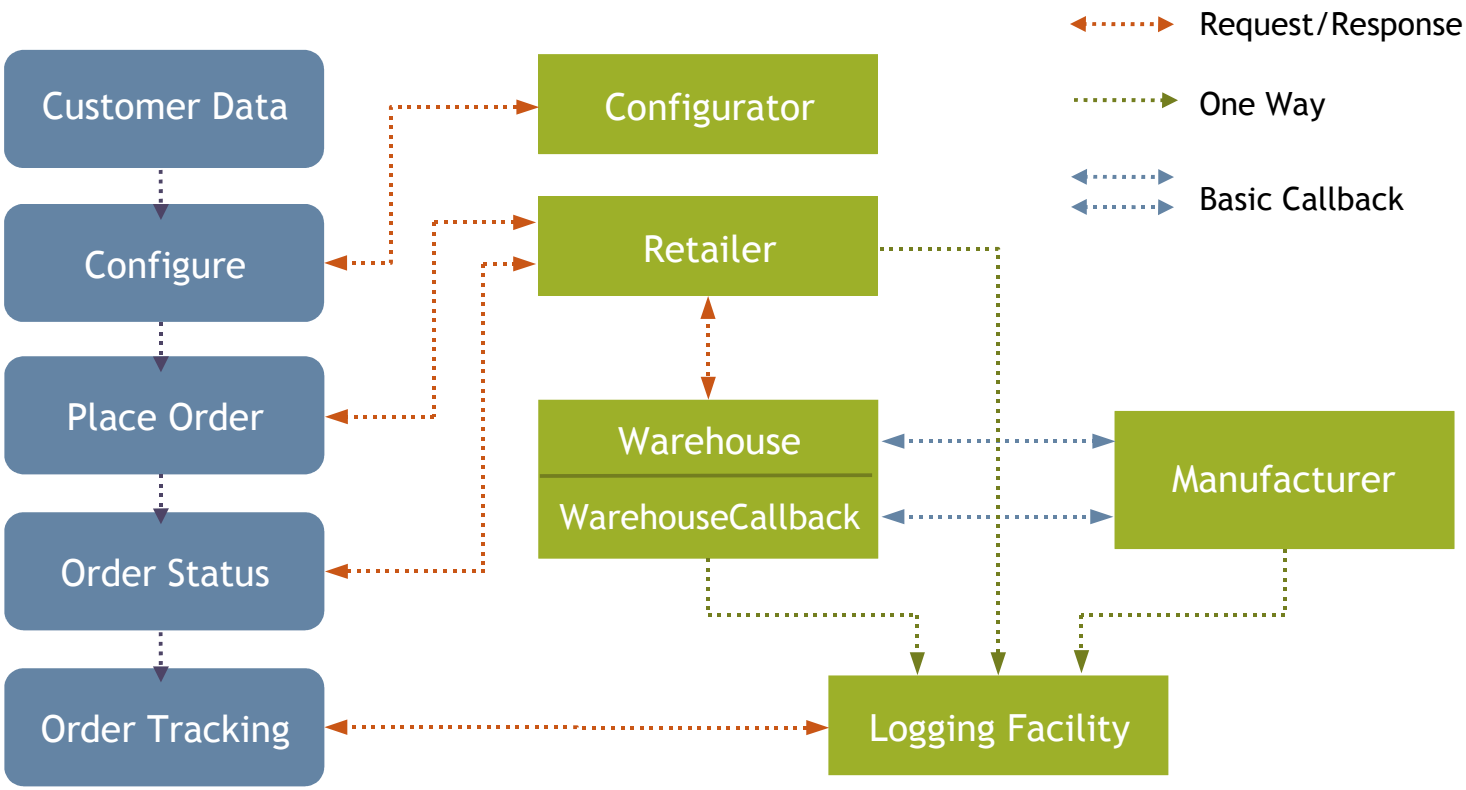
- Model a simplified supply chain management scenario
- Demonstrate features in WS-I Basic Profile 1.0
- Showcase interoperability between all vendor implementations (BEA, Bowstreet, Corillian, IBM, Microsoft, Novell, Oracle, Quovadx, SAP, Sun)

Documents & Implementation

- WS-I SCM Use Cases 1.0
- WS-I SCM Usage Scenarios
- WS-I SCM Technical Architecture 1.0
- Sample Application 1.0 implementations

© April 25, 2005 Novell Inc.

Supply Chain Example



© April 25, 2005 Novell Inc.

Novell

Interoperability Demo

WS-I work in progress
(Continued involvement by Novell)

Attachment Profile 1.0

- Sample Application
- Mid 2004
- WS-1 Testing Tools not yet ready

Basic Security Profile 1.0

- Profile work in progress
- Mid 2005
- WS-1 Testing Tools not yet ready

© April 25, 2005 Novell Inc.

Conclusion

Significant advances in Web Services Interoperability

- WS-I Basic Profile 1.0
- WS-I Attachment Profile 1.0

Evolving standards in Web Services Security

- WS-I Basic Security Profile
- Continued involvement with standards organizations

Novell WSSDK supports all features in WS-I Profile plus features not used in the profile (e.g. rpc/encoded)

- Recommended to follow WS-I profiles when interoperability with other vendors is critical

© April 25, 2005 Novell Inc.

Novell.

Novell®

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Novell, Inc., makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

Novell