The Good, The Bad and The Ugly

WS-* Specifications For SOA

Peter Tröger
Web Services And SOA

- If a SOA is the solution, what was the original problem?
  - Interoperate and integrate legacy systems
  - Inappropriate mapping between business processes and IT processes
- Service-Oriented Computing (M.P. Papazoglou and D. Georgakopoulos)
Terminology

- Web Service (W3C Web Service Architecture Group)
- Machine-to-machine network interaction
- Machine-processable WSDL interface description
- Interaction using SOAP messages
- Typically conveyed using HTTP (but not a must)
- XML serialization
- Endpoint: Reference of target for Web service messages
Service Interaction

Consumer

Provider
Service Interaction

Registry

Find

Publish

Consumer

Interact

Provider
Interoperability On Different Layers

Diagram showing the interaction between application layers and network layers through TCP.
Interoperability on Different Layers
Interoperability On Different Layers

Application

Messaging

Transport

Network

SOAP

HTTP

TCP
Interoperability on Different Layers

Application

Messaging

Transport

Network

TCP

SOAP

MQ?

Intermediary?
Interoperability On Different Layers

Application

Features ?

Messaging

Transport

Network

Application

Features ?

Messaging

Transport

Network

SOAP

MQ ? Intermediary ?

TCP
Interoperability on Different Layers

- Application
- Messaging
- Transport
- Network

Features?

WS-*

SOAP

MQ? Intermediary?

TCP
WS-* Standards

• Everything on-top-of SOAP, WSDL, and UDDI
• Huge number of documents, topics, and versions
• Different organizations with different influence and power
• Concurrent specifications, competing implementations
• Microsoft with / against IBM against the rest
WS-Why (Don Box)

• WS-DesertIsland –
  a must have for core XML Web services
• WS-IslandResort -
  the next layer of important specs
• WS-NewZealand -
  specs you'd probably need once in a lifetime
• WS-IslandOfDoctorMoreau -
  the ugly step children of the WS-* spec family
• WS-FantasyIsland - specs Don would love to see
OASIS Standardization Body

- Organization for the Advancement of Structures Information
- Founded 1993 as SGML Open, till 1998
- Focus on high-level Web services
- Royalties and Patents
- Standardization track: TC draft, TC specification, OASIS standard
W3C
Standardization Body

- Founded 1994 by Tim Berners-Lee
- Internet standards (HTTP, HTML, XML)
- All ratified standards must be royalty-free
- Standardization track:
  - Working group note and working draft
  - Candidate recommendation
  - Proposed / W3C recommendation
- Member submission
WS-I
Standardization Body

• Web Services Interoperability Organization
• Founded in 2002 by Microsoft, IBM and others
• Clarifies ambiguities and restricts WS specifications
• Profiles for basic specs (SOAP, WSDL, UDDI) and Security
• Conformance test tool chain (Java, C#)
Web Service
Specification Landscape
Web Service Specification Landscape

- Messaging
- Metadata
- Security

- XML, Schema

- Transport (HTTP, MQ, TCP, IIOP, ...)

11
Web Service Specification Landscape

Service Composition / Business Process

Transactions
Resources
Management
Agreement
Reliability

Messaging
Metadata
Security

XML, Schema

Transport (HTTP, MQ, TCP, IIOP, ...)

11
Web Service Specification Landscape

Service Composition / Business Process

Interoperability
Transactions
Resources
Management
Agreement
Reliability

Messaging
Metadata
Security

XML, Schema

Transport (HTTP, MQ, TCP, IIOP, ...)

11
Implementations

- SOAP, WSDL, UDDI - Endless number of implementations
- WS-* specifications
  - Apache projects (Axis, WSFX, Muse, ...)
  - IBM Emerging Technologies Toolkit (ETTK)
  - Microsoft Web Services Enhancements toolkit (WSE) / Windows Communication Foundation (WCF)
  - Sun Java Java Web Services Developer Pack (WSDP)
  - Verisign Trust Service Integration Kit (TSIK)
**WS-CheckList**

- Participating Companies?
  - Either agreed by IBM and MS, or we have two concurrent specifications
- Status in standardization body?
- Maturity of the document
- Implementations?
  - More than one implementation is an indicator for real-world adoption
- Moving target !!!
**WS-CheckList**

- Participating Companies?
  - Either agreed by IBM and MS, or we have two concurrent specifications
- Status in standardization body?
- Maturity of the document
- Implementations?
  - More than one implementation is an indicator for real-world adoption
- Moving target !!!
Web Service
Specification Landscape

- Service Composition / Business Process
- Interoperability
- Transactions
- Resources
- Management
- Agreement
- Reliability
- Messaging
- Metadata
- Security
- XML, Schema
- Transport (HTTP, MQ, TCP, IIOP, ...)

SOAP

- Current version 1.2 (W3C), version 1.1 widely used (Don Box)
- Envelope structure for defining messages
- Processing model (roles, relays)
- Set of encoding rules for application specific data types
- Fault information
- Convention for doing RPC
SOAP in 30 Seconds

Client

1. <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Add xmlns="Web Service Namespace">
      <a>5</a>
      <b>8</b>
    </Add>
  </soap:Body>
</soap:Envelope>

Server

2. <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AddResponse xmlns="Web Service Namespace">
      <AddResult>13</AddResult>
    </AddResponse>
  </soap:Body>
</soap:Envelope>
Academic SOAP Usage Scenarios

- Fire-and-forget to single/multiple receiver (notifications)
- RPC, Request/response asynchronous communication
- (Multiple) Third party intermediary
- Request with acknowledgment
- Request with encrypted payload
- Multiple asynchronous responses
- Caching
- Routing
WS-Addressing

- Scenario: We need a transport-neutral session identifier in our SOAP requests
  - Altering (HTTP) URL does not help
  - Standardized extension of SOAP header needed

- Scenario: Transport-neutral transmission through processing nodes
  - Asynchronous / event-based processing
  - Endpoint reference (EPR)
  - Message Information (MI) header
WS-Addressing
Endpoint Reference

- Convey the information for endpoint identification and referencing in an XML document
  - WSDL is for the service, EPR is for the endpoint
- Inclusion of parameters in endpoint description
  - Provided by reference issuer, opaque to client
- Inclusion of metadata
  - WSDL information can be part of the EPR
- Mapping to SOAP 1.1 / 1.2 and WSDL 1.1 / 2.0
<wsa:EndpointReference
    xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns:shop="mailto:admin@example.org">
    <wsa:Address>
        http://exampleshopping.net/hello/index.asmx
    </wsa:Address>
    <wsa:ReferenceParameters>
        <shop:UserProfile>123456</shop:UserProfile>
        <shop:SessionID>987654</shop:SessionID>
    </wsa:ReferenceParameters>
    <wsa:Metadata>
        <wsdl:definitions
            targetNamespace="http://example.org/shop"
            xmlns:wsdl="http://schemas.xmlsoap.org/wsd1/">
            ...
        </wsdl:definitions>
    </wsa:Metadata>
</wsa:EndpointReference>
WS-A Message
Addressing Headers

• Transport-neutral end-to-end message characteristics as additional SOAP header entries
• Communication of information that is distributed over SOAP and transport layer
  • Source and destination endpoint
  • Reply and fault endpoint (reliable messaging)
  • Message identifier
  • Policy
• Basis for many on-top WS specifications
Message Addressing Properties

<wsa:To>
  xs:anyURI</wsa:To>
<wsa:From>
  wsa:EndpointReferenceType</wsa:From>
<wsa:ReplyTo>
  wsa:EndpointReferenceType</wsa:ReplyTo>
<wsa:FaultTo>
  wsa:EndpointReferenceType</wsa:FaultTo>
<wsa:Action>
  xs:anyURI</wsa:Action>
<wsa:MessageID>
  xs:anyURI</wsa:MessageID>
<wsa:RelatesTo>
  RelationshipType="xs:anyURI">
  xs:anyURI</wsa:RelatesTo>
<wsa:ReferenceParameters>
  xs:any*/</wsa:ReferenceParameters>
Message Addressing Properties

- Destination (mandatory)
  - IRI of the intended receiver, independent from transport destination information
  - Equal to HTTP request URL in usual SOAP case
- Source endpoint (optional)
  - Endpoint where the message was originated
- Predefined address values
  - http://www.w3.org/2005/08/addressing/anonymous
  - http://www.w3.org/2005/08/addressing/none
Message Addressing Properties

- Reply endpoint (optional)
  - Intended receiver for replies to this message
  - Source endpoint MAY be used when absent
  - Requires also message ID

- Fault endpoint (optional)
  - Intended receiver for faults related to this message
  - Reply or source endpoint MAY be used when absent
  - Requires also message ID
Message Addressing Properties

- Relationship value (optional)
  - Relationship type as attribute
  - Only reply type is predefined by the spec
- Action specifier (optional)
  - Similar to SOAP action specifier
- Message identifier (optional)
  - Identifies the message uniquely in time and space
  - Must be set when a reply is expected
Example SOAP Request

```xml
<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:MessageID>http://example.com/4711</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://example.com/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:To>mailto:fabrikam@example.com</wsa:To>
    <wsa:Action>http://example.com/mail/Delete</wsa:Action>
  </S:Header>
  <S:Body>
    <f:Delete xmlns:f="http://example.com/">
      <maxCount>42</maxCount>
    </f:Delete>
  </S:Body>
</S:Envelope>
```
Example SOAP Response

```xml
<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:MessageID>http://example.com/1147</wsa:MessageID>
    <wsa:RelatesTo>http://example.com/4711</wsa:RelatesTo>
    <wsa:To>http://example.com/client1</wsa:To>
    <wsa:Action>http://example.com/mail/DeleteAck</wsa:Action>
  </S:Header>
  <S:Body>
    <f:DeleteAck xmlns:f="http://example.com/"/>
  </S:Body>
</S:Envelope>
```
WS-Addressing

Conclusion

- All relevant information for processing and routing inside of the SOAP envelope
- Less problems with intermediaries or non-HTTP transport protocols
- Client is unaware of additional information needed by the provider
  - Supports factory / registry patterns
- Support for asynchronous messaging
- Standardized way to pass around WS references
Message Transmission Optimization Mechanism

• Optimization of the transmission and/or wire format of SOAP messages
• Backward compatible to SOAP with Attachments
• Based on XOP (XML-binary optimized packaging)
  • Repackaging & transmitting base64Binary data in native format, keep link in original Infoset
  • Base64 data comes in separate MIME parts
  • MIME / Multipart serialization of SOAP messages
• Hop-by-hop contract
Asynchronous Messaging

- WS-Notification 1.2 (IBM)
- WS-BaseNotification
- WS-BrokeredNotification
- WS-Eventing (IBM, MS)
Push Notification With Web Services

Event Source

Subscribe Event

Renew Event

Notify Event

GetStatus Event

Unsubscribe Event

End Event

Event Sink
WS-Notification

- Web Service Base Notification 1.3
  - Interfaces for consumers and producers
- Web Services Topics 1.3 (WS-Topics)
  - Mechanisms to organize and categorize items of interest for subscription
- Web Services Brokered Notification 1.3
  - Interface for Notification Broker
**WS-baseNotification**

- Different roles with their endpoints - producer, consumer, subscription manager, subscriber
- Two ways of notification (no response expected)
  - Producer sends raw application-specific content
  - Producer send special Notify message
    - Subscription reference
    - Topic and topic dialect
    - Producer reference
Notify Message Infoset

...
<wsnt:Notify>
  <wsnt:NotificationMessage>
    <wsnt:SubscriptionReference>
      wsa:EndpointReferenceType
    </wsnt:SubscriptionReference>
  <wsnt:Topic Dialect="xsd:anyURI">
    {any} 
  </wsnt:Topic>?
  <wsnt:ProducerReference>
    wsa:EndpointReferenceType
  </wsnt:ProducerReference>
  ?
  <wsnt:Message>
    {any}
  </wsnt:Message>
  +
  {any} *
</wsnt:Notify>
SOAP Notify Example

<s:Envelope><s:Header>
  <wsa:Action>
   http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify
  </wsa:Action>
  ...
</s:Header><s:Body>
  <wsnt:Notify><wsnt:NotificationMessage>
    <wsnt:SubscriptionReference>
      <wsa:Address>http://www.example.org/SubManager</wsa:Address>
    </wsnt:SubscriptionReference>
    <wsnt:Topic Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
      npex:SomeTopic
    </wsnt:Topic>
    <wsnt:ProducerReference>
      <wsa:Address>http://www.example.org/NotiProd</wsa:Address>
    </wsnt:ProducerReference>
    <wsnt:Message>
      <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>
    </wsnt:Message>
  </wsnt:NotificationMessage></wsnt:Notify>
</s:Body></s:Envelope>
WS-Notification Subscription

- Several options for subscription
- Notification consumer EPR
- Boolean filter expressions
  - Order and timing for tests defined by producer
  - Topic, or XPath expression to check message content
- Initial termination time
- Subscription policy (e.g. # of messages)
- Indicator for raw subscription
- Subscription response with EPR and termination time
Subscription Example

<s:Envelope><s:Header>
  <wsa:Action>
    http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeRequest
  </wsa:Action>
  ...
</s:Header><s:Body>
  <wsnt:Subscribe>
    <wsnt:ConsumerReference>
      <wsa:Address>http://www.example.org/NotificationConsumer</wsa:Address>
    </wsnt:ConsumerReference>
    <wsnt:Filter>
      <wsnt:TopicExpression>
        Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
        npex:SomeTopic
      </wsnt:TopicExpression>
      <wsnt:MessageContent>
        Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
        boolean(ncex:Producer="15")
      </wsnt:MessageContent>
    </wsnt:Filter>
    <wsnt:InitialTerminationTime>
      2005-12-25T00:00:00.00000Z
    </wsnt:InitialTerminationTime>
  </wsnt:Subscribe>
</s:Body></s:Envelope>
WS-Notification vs. WS-Eventing

• WS-Notification features
  • Support for small devices  
    (restricted set of mandatory features) 
  • Support for direct and brokered notification 
  • Transformation and aggregation of Topics 
  • Runtime metadata (e.g. available subscription types) 
  • Broker federations 
  • Based on WS-ResourceProperties and WS-ResourceLifetime (from WSRF)
Web Service Specification Landscape

Service Composition / Business Process

- Interoperability
- Transactions
- Resources
- Management
- Agreement
- Reliability

- Messaging
- Metadata
- Security

XML, Schema

Transport (HTTP, MQ, TCP, IIOP, ...)

40
• Web Service Description Language (WSDL) 1.1 / 2.0
• Universal Description, Discovery and Integration (UDDI)
• WS-MetadataExchange
• WS-Policy (WS-PolicyAssertions, WS-PolicyAttachment)
• WS-Discovery (Bea, Canon, Intel, Microsoft)
• WS-Agreement
• WS-Inspection
UDDI

• Universal Description Discovery and Integration
• Repository broker service for discovery of services, offered as Web service
• Data model for service info and metadata
• UDDI Business Registry
• Part of WS-I Basic Profile
• Unbeloved child of the WS community
WS-MetadataExchange

- Web Service consumers need metadata to interact
  - Messages, protocols, endpoint addresses
  - Policies
- Request for specific metadata type (dialect)
- Simple interaction to get service description (communication bootstrapping)
- Intended for retrieval of WSDL and WS-Policy

C
S
I

SAP, IBM, Microsoft, Bea, CA, ...
Industry Proposal
Microsoft, IBM, ...
<s12:Envelope
    xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
    xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
    <s12:Header>
        <wsa:Action>
        </wsa:Action>
        <wsa:MessageID>4f6d5c6027f4</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://example.com/MyEndpoint</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://server.example.org/YourEndpoint</wsa:To>
    </s12:Header>
    <s12:Body>
        <wsx:GetMetadata>
            <wsx:Dialect>http://schemas.xmlsoap.org/wsdl/</wsx:Dialect>
        </s12:Body>
    </s12:Envelope>
<s12:Envelope
    xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
    xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
  <s12:Header>
    <wsa:Action>
    </wsa:Action>
    <wsa:RelatesTo>4f6d5c6027f4</wsa:RelatesTo>
    <wsa:To>http://client.example.com/MyEndpoint</wsa:To>
  </s12:Header>
  <s12:Body>
    <wsx:Metadata>
      <wsx:MetadataSection
        Dialect='http://schemas.xmlsoap.org/wSDL/'>
        <wsdl:definitions ... </wsdl:definitions>
      </wsx:MetadataSection>
    </wsx:Metadata>
  </s12:Body>
</s12:Envelope>
WS-Policy

• Describe and communicate the policies of a web service
  • Policy: possibly empty collection of policy alternatives
  • Policy alternative: possibly empty collection of policy assertions
  • Policy assertion: Domain-specific requirement or capability, or other property of behaviour
  • ‘Wire’ properties (authentication, transport)
  • Extended properties (e.g. QoS characteristics)
WS-Policy Data Model

Policy

Policy Alternative

Policy Alternative is an unordered collection of Policy Alternatives

Policy Assertion

Policy Alternative is an unordered collection of Policy Assertions

Policy Assertion is identified by a QName and may contain Parameters and a Nested Policy

Policy Assertion Parameters are the opaque payload of a Policy Assertion

QName

Nested Policy

Policy Assertion Parameters

WS-Policy Data Model
WS-Policy Example

<wsp:Policy>
  <wsp:ExactlyOne>
    <wsp:All>
      <wsse:SecurityToken>
        <wsse:TokenType>wsse:Kerberosv5TGT</wsse:TokenType>
      </wsse:SecurityToken>
    </wsp:All>
    <wsp:All>
      <wsse:SecurityToken>
        <wsse:TokenType>wsse:X509v3</wsse:TokenType>
      </wsse:SecurityToken>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
WS-Policy Extensions

- WS-PolicyAttachment
  - Associating policies with their subjects
  - Reference policies from WSDL definitions
  - Associate policies with deployed WS endpoints
  - Associate policies with UDDI entities
- WS-SecurityPolicy
  - WS-Policy assertions that apply to WS-Security
- WS-PolicyAssertions
  - Common message policy assertions
Attaching Policy Expressions To WSDL

```xml
<Policy wsu:Id="common2">
  <mtom:OptimizedMimeSerialization wsp:Optional="true"/>
  <wsap:UsingAddressing/>
</Policy>

<Policy wsu:Id="secure2">
  <ExactlyOne>
    <sp:TransportBinding>…</sp:TransportBinding>
    <sp:AsymmetricBinding>…</sp:AsymmetricBinding>
  </ExactlyOne>
</Policy>

<wSDL:binding name="SecureBinding" type="tns:RealTimeDataInterface">
  <PolicyReference URI="#secure2"/>
  <wsdl:operation name="GetRealQuote">…</wsdl:operation>
  ...
</wSDL:binding>

<wSDL:service name="RealTimeDataService">
  <wsdl:port name="RealTimeDataPort" binding="tns:SecureBinding">
    <PolicyReference URI="#common2"/>
    ...
  </wsdl:port>
</wSDL:service>
```
Web Service Specification Landscape

- Service Composition / Business Process
- Interoperability
- Transactions
- Resources
- Management
- Agreement
- Reliability
- Messaging
- Metadata
- Security
- XML, Schema
- Transport (HTTP, MQ, TCP, IIOP, ...)

51
## Security

- **WS-Security**
- **SOAP message security**
- **Several on-top-of specifications**
- **WS-Trust**
  - Dissemination of credentials between different trust domains
- **Security token service**
- **WS-SecureConversation**
  - Establish and share security context for connections

<table>
<thead>
<tr>
<th>C</th>
<th>IBM, Microsoft, Sun, Verisign</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>v1.x: OASIS Standard</td>
</tr>
<tr>
<td>I</td>
<td>WSE, ETTK, ...</td>
</tr>
</tbody>
</table>
Security Triad - CIA

- Confidentiality
  - Keeping secrets a secret
  - Encryption and access control
  - Ensure that parties are really who they claim to be
- Integrity
  - No unintended modification
  - Signing and hashing
  - Might include non-repudiation
- Availability (DDOS)
WS-Security

• Generic mechanism to associate security tokens with messages
• How to encode binary security tokens (X.509, Kerberos)
• Based on collection of different specifications
  • XML Signature and XML Encryption to secure XML elements in a standardized manner
• Different profile specifications for authentication mechanisms -> unsigned / signed security tokens
  • Username / X.509 / SAML / Kerberos Token Profile
Token Examples

<wsse:UsernameToken>
  <wsse:Username>Sven</wsse:Username>
  <wsse:Password Type="wsse:PasswordText">Kennwort</wsse:Password>
</wsse:UsernameToken>

<wsse:UsernameToken>
  <wsse:Username>Sven</wsse:Username>
  <wsse:Password Type="wsse:PasswordDigest">
    KE6QugOpkPyT3Eo0SEgT30W4Keg=
  </wsse:Password>
  <wsse:Nonce>5uW4ABku/m6/S5rnE+L7vg==</wsse:Nonce>
  <wsu:Created>2002-08-19T00:44:02Z</wsu:Created>
</wsse:UsernameToken>

<wsse:BinarySecurityToken
  ValueType="wsse:X509v3"
  EncodingType="wsse:Base64Binary">
  MIiHdjCCB...
</wsse:BinarySecurityToken>
X.509 Scenario

Web Service Request
- Body signed with Private Key of CA Cert
- Body encrypted with Public Key of SP Cert
- with request including CA X.509 Cert

Consuming Application (CA)

Service Provider (SP)

Web Service Response
- Body Signed with Private Key of SP Cert
- Body Encrypted with Public Key of CA Cert
- with response including SP X.509 Cert

1) Consuming Application's Certificate
2) Root Certificate of SP’s Certificate Authority

1) Service Provider's Certificate
2) Root Certificate of CA’s Certificate Authority
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
    <S:Header>
        <wsse:Security>
            <wsse:BinarySecurityToken ValueType="wsse:X509v3" EncodingType="wsse:Base64Binary"
                Id="X509Token">
                MIIEZzCCA9CgAwIBAgIQEmtJZc0rqrKh5i...
            </wsse:BinarySecurityToken>
            <ds:Signature>
                <ds:SignedInfo>...</ds:SignedInfo>
                <ds:SignatureValue>BL8jdfToEb1l/vXcMZNjPOV...</ds:SignatureValue>
                <ds:KeyInfo>
                    <wsse:SecurityTokenReference>
                        <wsse:Reference URI="#X509Token"/>
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
        </wsse:Security>
    </S:Header>
    <S:Body>
        <tru:StockSymbol xmlns:tru="http://fabrikam123.com/payloads">
            QQQ
        </tru:StockSymbol>
    </S:Body>
</S:Envelope>
Encryption Example

<S11:Envelope xmlns:S11="..." xmlns:ds="..." xmlns:wsse="..." xmlns:xenc="...">
  <S11:Header>
    <wsse:Security>
      <xenc:EncryptedKey>
        <xenc:EncryptionMethod Algorithm="..."/>
        <ds:KeyInfo>
          <wsse:SecurityTokenReference>
            <ds:X509Data>
              <ds:X509IssuerSerial>
                <ds:X509IssuerName>DC=ACMECorp, DC=com</ds:X509IssuerName>
                <ds:X509SerialNumber>12345678</ds:X509SerialNumber>
              </ds:X509IssuerSerial>
            </ds:X509Data>
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </wsse:SecurityTokenReference>
    </wsse:Security>
  </S11:Header>
  <S11:Body>
    <xenc:EncryptedData Id="encrypted" Type="...">
      <xenc:CipherData><xenc:CipherValue>...</xenc:CipherValue></xenc:CipherData>
    </xenc:EncryptedData>
  </S11:Body>
</S11:Envelope>
WS-Security Usage

![Diagram of WS-Security Usage]

1) Consuming Application’s Certificate
2) Root Certificate of SP’s Certificate Authority

1) Service Provider’s Certificate
2) Root Certificate of CA’s Certificate Authority
Web Service Specification Landscape

- Service Composition / Business Process
- Interoperability
- Transactions
- Resources
- Management
- Agreement
- Reliability
- Messaging
- Metadata
- Security
- XML, Schema
- Transport (HTTP, MQ, TCP, IIOP, ...)

60
Reliability

- Reliable point-to-point transmission of messages (message acknowledgment, retransmission)

- WS-ReliableMessaging

- WS-Reliability

| C | IBM, MS, Bea, TIBCO, ...
| S | Industry proposal
| I | Apache, ETTK, WSE, ...

| C | Sun, Oracle, Fujitsu, ...
| S | v1.1: OASIS standard
| I | Apache, WSE, ...
Delivery Assurance

- Messages might be lost, duplicated, reordered, or in an unknown state
- Well-known concepts, realized with SOAP-based message protocol
  - AtMostOnce delivery
  - AtLeastOnce delivery
  - ExactlyOnce delivery
  - InOrder delivery
WS-Reliab* Concept

Producer
Submit
Notify
Sending RMP

Consumer
Deliver
Respond
Receiving RMP

(Reliable Message)
( RM-Reply) (Respond Payload)
Web Service Specification Landscape

Service Composition / Business Process

Interoperability
Transactions
Resources
Management
Agreement
Reliability

Messaging
Metadata
Security

XML, Schema

Transport (HTTP, MQ, TCP, IIOP, ...)

64
Transactions

- Define mechanisms for transactional interoperability in WS applications
- Huge set of specifications
WS-Coordination

• Framework for the coordination of a distributed application
• When application needs to reach consistent agreement between parts
• Integrates existing legacy transaction processing or workflow systems
• Definition of a context
  • Context propagation between services
WS-AtomicTransaction

- Based on WS-Coordination
- For short-lived, all-or-nothing activities
- Defines three specific agreement protocols
  - Volatile two-phase commit
  - Durable two-phase commit
  - Completion

IBM, MS, Bea, Iona, ...
Industry proposal
ETTK, WCF, ...
Web Service Specification Landscape

Service Composition / Business Process

Interoperability
Transactions
Messaging
Resources
Metadata
Management
Agreement
Reliability
Security
XML, Schema
Transport (HTTP, MQ, TCP, IIOP, ...)

68
Resources

- Web Services Resource Framework (WSRF)
  - WS-BaseFaults
  - WS-ServiceGroup
  - WS-ResourceProperties
  - WS-ResourceLifetime
- WS-Transfer

| C | IBM, Oracle, HP, CA, ...
| S | OASIS standard
| I | Apache, ETTK, Globus, ...

69
Web Services Resource Framework (WSRF)

- Covers representation of stateful resources as Web service (‘WS-Resource’)
- Updated version of GGF OGSI approach
  - WS-ResourceProperties: Querying and changing state information
  - WS-ResourceLifetime: Free WS-Resources
  - WS-BaseFaults, WS-ServiceGroup, WS-Notification
Web Service Specification Landscape

Service Composition / Business Process

- Interoperability
- Transactions
- Resources
- Management
- Agreement
- Reliability
- Messaging
- Metadata
- Security
- XML, Schema
- Transport (HTTP, MQ, TCP, IIOP, ...)

72
Management

- Management using Web Services (MUWS)
- Management of Web Services (MOWS)
- Web Services Management Framework (WSMF) - HP
- WS-Management
- WS-Events
WSDM-MUWS

- WSDM - Web Service Distributed Management
- Concept of manageability capabilities
- Management (of IT resources) using Web services
Manageability Capabilities

- Uniquely defined, with specific semantics
- Extensible (DMTF, CIM)
- Composable
- Some predefined values (like identity)
WSDM-MOWS

- Management of Web service endpoints
- Through Web services - relies on the MUWS concepts
- Management operation separately, or as part of the functional endpoint
- Includes support for notification
**Manageability Capabilities**

- Obtaining management interface
- Identity of managed resource
- Endpoint metrics
  - Number of requests, failed requests and successful requests
- Max / Last response time
- Operational state (WSLC)
- Service time
Web Service Specification Landscape

Service Composition / Business Process

- Interoperability
- Messaging
- Resources
- Management
- Agreement
- Reliability
- Security

XML, Schema

Transport (HTTP, MQ, TCP, IIOP, ...)

Transactions
Web Services Business Process Execution Language (BPEL4WS / WS-BPEL) 1.1
- Describes flow from single endpoint
WS-Choreography 1.0
- Rules of interaction with multiple business process endpoints
BPEL4WS

- XML language for describing business processes
- Describes conversation of WSDL message exchanges - Web service composition language
- Process offers coarse-grained Web service for aggregated fine-grained Web services
  - Each step as activity
  - Partner links (services used by business process)
  - Variables for long-running interactions
  - Control flow, correlation, scopes

C

BEA, IBM, Microsoft

S

OASIS Standard

I

Axis
**WS-Choreography**

- Define sequence and conditions for SOAP message exchange
- Information model for Choreography Definition Language

<table>
<thead>
<tr>
<th></th>
<th>Abstract</th>
<th>Portable</th>
<th>Concrete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of Messages</td>
<td>Identified</td>
<td>Identified</td>
<td>Identified</td>
</tr>
<tr>
<td>Message Structure</td>
<td>Not Defined</td>
<td>Defined</td>
<td>Defined</td>
</tr>
<tr>
<td>Conditions</td>
<td>Identified</td>
<td>Identified</td>
<td>Identified</td>
</tr>
<tr>
<td>Condition evaluation rules</td>
<td>Not defined</td>
<td>Defined as far as possible</td>
<td>Defined as far as possible</td>
</tr>
<tr>
<td>Technology used</td>
<td>Not defined</td>
<td>Defined</td>
<td>Defined</td>
</tr>
<tr>
<td>Message Endpoint Data</td>
<td>Not defined</td>
<td>Not Defined</td>
<td>Defined</td>
</tr>
</tbody>
</table>
Web Service Specification Landscape

Service Composition / Business Process

Interoperability
Transactions
Resources
Management
Agreement
Reliability

Messaging
Metadata
Security

XML, Schema

Transport (HTTP, MQ, TCP, IIOP, ...)

82
Interoperability

• WS-I profiles
  • Basic Profile Working Group
  • Basic Security Profile Working Group
  • Requirements gathering, sample scenarios, testing tools, XML schema issues
• Clarifications: Missing details, interop problems, attachments, SOAP binding, security token, ...
• Sun: “Shadow government for standards”
The Good, The Bad, And The Ugly

• The Good
  • IBM + MS + somebody else
  • W3C recommendation / OASIS standard, maybe also covered by WS-I

• The Bad
  • Superseded specifications
  • Specs without participation from the big vendors

• The Ugly
  • Proposed standards or company proposals
WS Standards In ASG
WS Standards in ASG

- WS-Security
- SOAP
- WSRF
- WS-BPEL
Conclusions & Remarks

- Web services are no promise for interoperability
- Web services are no promise for service orientation
- Web services are maybe a ‘bad version’ of CORBA (researchers still fighting)
- Web services are (no longer) bound to the Web
  - But: Google Maps is named a Web service
  - So please: Start talking protocol names (SOAP, HTTP)
Conclusions & Remarks

- Don’t fear to read the standard
- Respect standard compliance
- The process is your friend
- Interoperability ALWAYS requires YOUR participation and testing
Conclusions & Remarks

- Don’t fear to read the standard
- Respect standard compliance
- The process is your friend
- Interoperability ALWAYS requires YOUR participation and testing
Some Sources

- soaprpc.com
- Scott Mitchell. An Extensive Examination of Web Services: Part 1. 4guysfromrolla.com
- Jordi Albornoz. Finding your way through Web service standards, Part 1. IBM DeveloperWorks
- David Booth et.al. Web Services Architecture. W3C Web Services Architecture Working Group Note