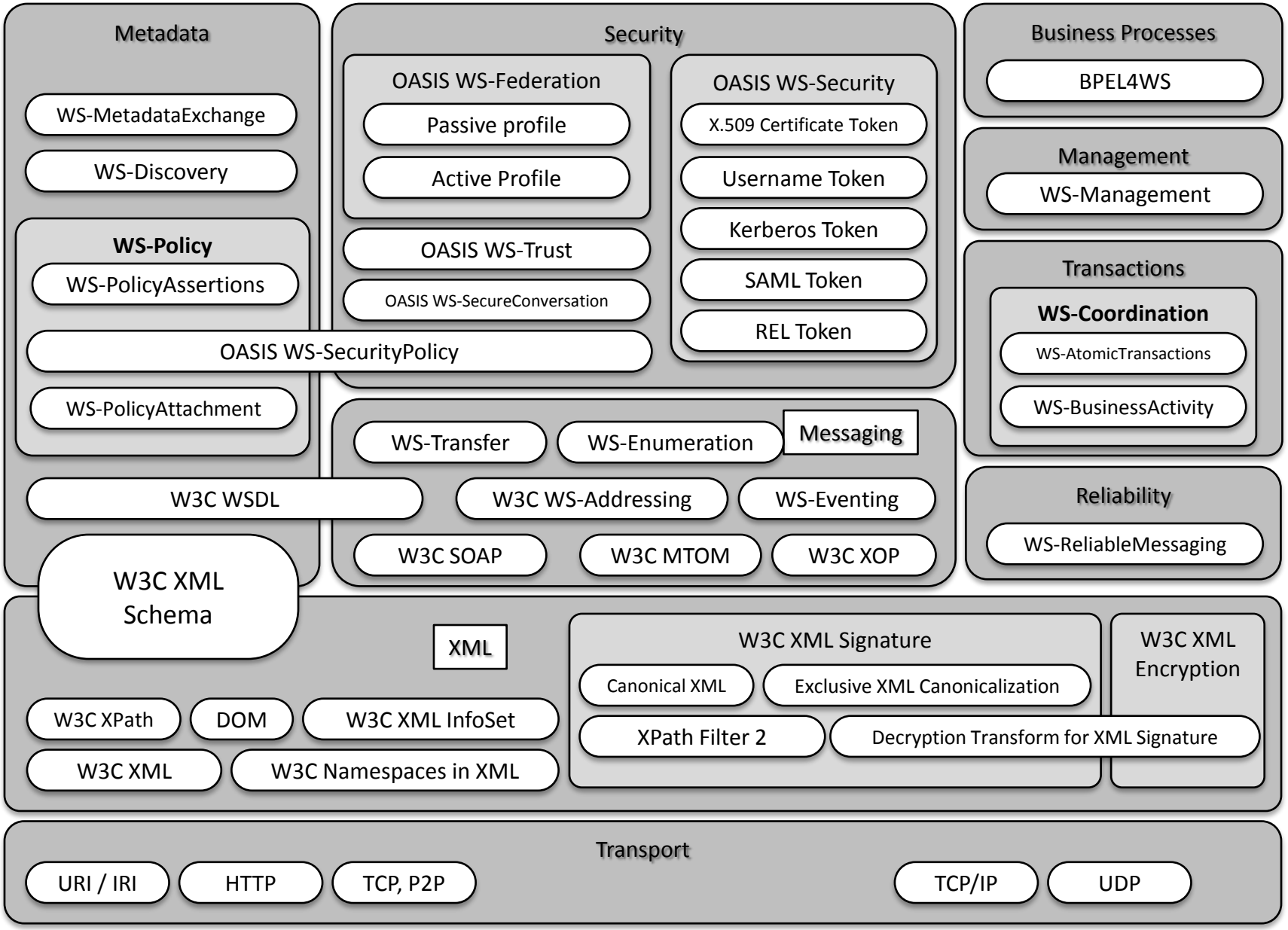
SOA-Tag Koblenz – 28. September 2007

WS-Security Tutorial

Dr.-Ing. Christian Geuer-Pollmann
European Microsoft Innovation Center
Aachen, Germany

WS-FooBar – Buchstabensuppe

Specifications and Standards



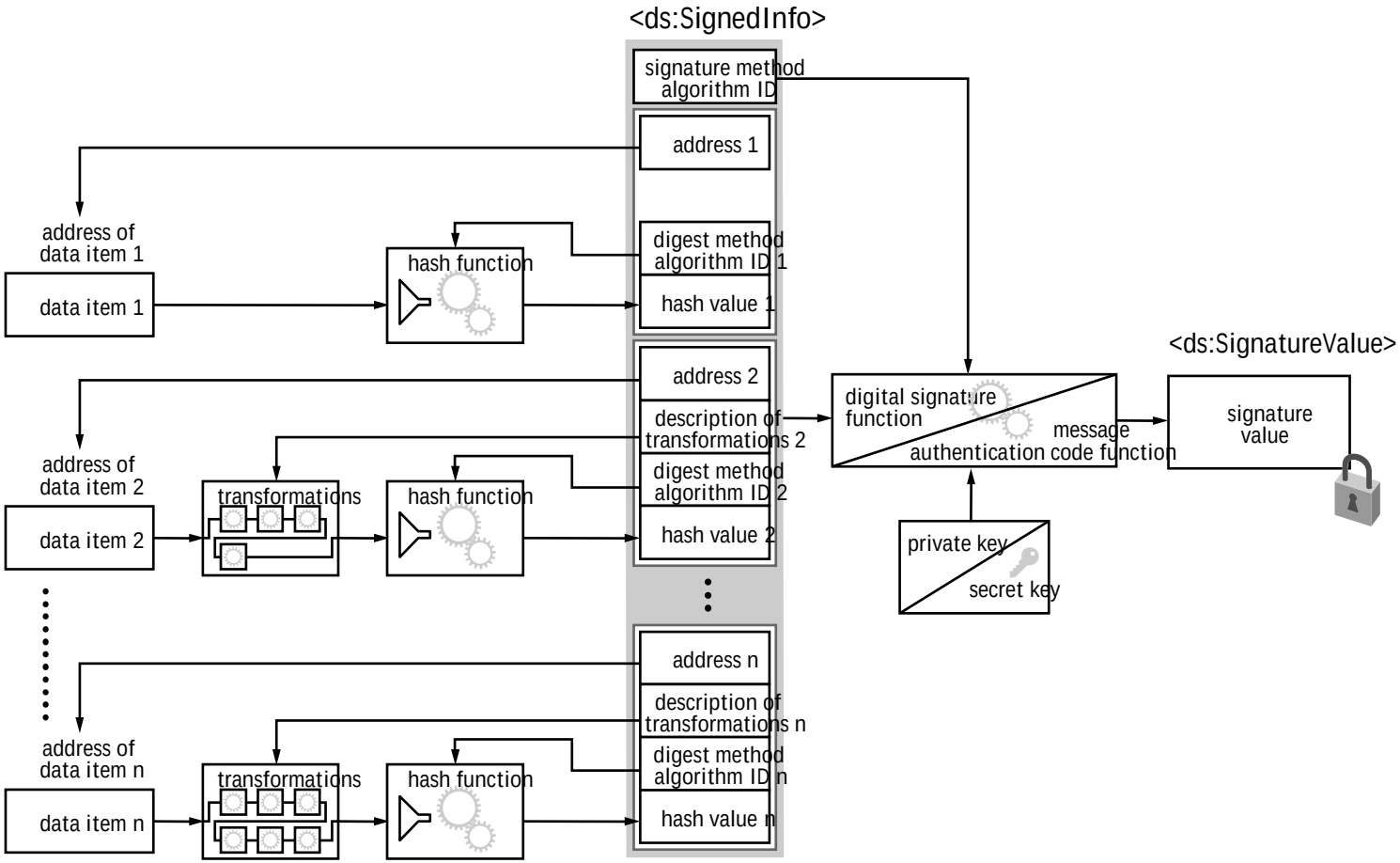
XML Core

- XML – eXtensible Markup Language
- Namespaces – Binding elements and attributes to a URI
- DOM – Document Object Model
- XPath – Addressing parts of an XML document
- XML Information Set (Info Set) – XML Data model
- XML Schema – Describes schemas for XML documents

XML Security

- XML Signature – Signature expressed in XML
  - Signature can cover both XML and non-XML
- Canonicalization (c14n) turns XML into octets (for digest)
- Exclusive C14n fixes problems in C14n
- XML Signature introduces “transforms”
  - C14n is a transform
  - XPath and XSLT are others
- XML Encryption
  - Encrypting XML
  - Encrypting non-XML (but key is XML)

XML Signature



SOAP Messaging

- SOAP – Simple Object Access Protocol
- SOA – Service-oriented Architecture
- WS-Addressing – SOAP headers for addressing messages and services
- MTOM – Message Transmission Optimization Mechanism (Attachments for SOAP)
- XOP – XML-binary optimized packaging (reduce base64-bloat)
- WS-Eventing – Pub/sub model
- WS-Transfer – Accessing XML representations of resources (GET, PUT and DELETE for WS-*)
- WS-Enumeration – Accessing large collections

A SOAP message

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" >
 <soap:Header>
  <wsa:To>http://www.contoso.com/Service/Simulate.ashx</wsa:To>
  <wsa:ReplyTo>
   <wsa:Address>
     http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
  </wsa:ReplyTo>
  <wsa:Action>http://contoso.com/Simulate</wsa:Action>
  <wsa:MessageID>urn:uuid:c1f3be56…</wsa:MessageID>
 </soap:Header>
 <soap:Body>
   ... Here is the XML for the service
 </soap:Body>
</soap:Envelope>
```

Metadata

- WSDL – Web Services Description Language (Endpoints, Methods and Parameters)
- WS-Policy – Just a grouping construct (AND and OR) for ‘policy assertions’
- WS-PolicyAttachment – Bind WS-Policy to WSDL and UDDI
- WS-MetadataExchange (MEX) – Bootstrap to download metadata (such as WS-Policy and WSDL)
- WS-SecurityPolicy – Communications security requirements for a service
- WS-PolicyAssertions – Language settings etc.
- WS-Discovery – Multicast discovery protocol for LAN

Reliable messaging & Transactions

- WS-ReliableMessaging – The TCP of SOAP
  - In-order delivery
  - No lost messages
  - Recipient acknowledgements
- WS-Coordination
  - Defines message patterns for multi-party cooperation
- WS-AtomicTransaction
  - Builds on WS-Coordination
  - Defines message patterns for ACID transactions
- WS-BusinessActivity
  - Builds on WS-Coordination
  - Supports long running activities

Security

- Transport level vs. message security – SSL
- WS-Security – Sign and encrypt SOAP messages
  - Both header and body
  - How? See WS-SecPol
- WS-SecurityPolicy
  - Retrieve what a service expects
- WS-SecureConversation (“SSL for SOAP”)
  - Negotiate and manage a session key
- WS-Security Tokens
  - X.509 certificates
  - Username / password
  - Kerberos (for Intranet)
  - SAML (cross-organizationally)
  - SecPAL authorization
- WS-Trust
  - A security token service (STS) issues, validates, renews and cancels security tokens

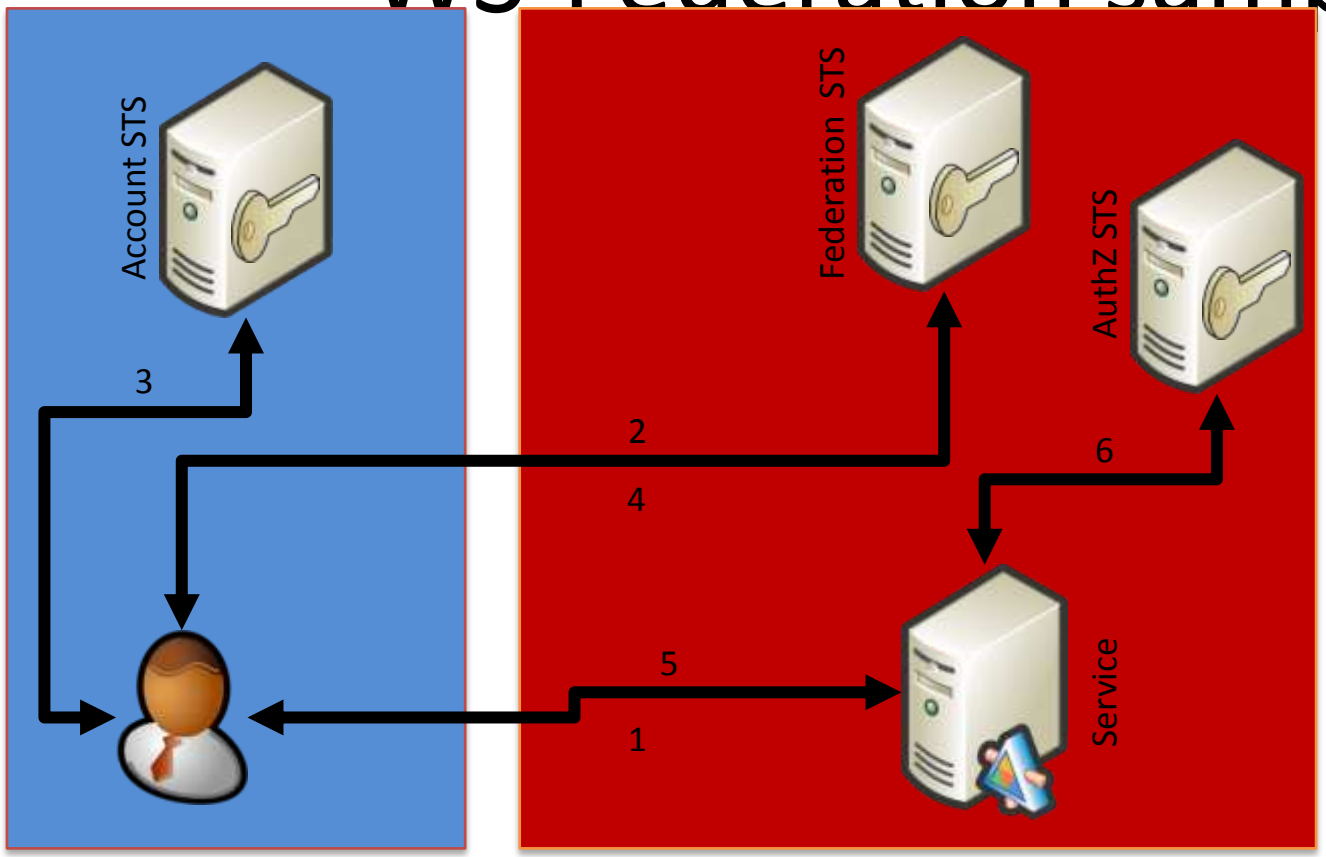
A protected message

```
<soap:Envelope>
 <soap:Header>
  <wsa:To>http://localhost/STSClient/STSX509.ashx</wsa:To>
  <wsa:Action>http://schemas.xmlsoap.org/ws/2005/02/trust/RST/Issue</wsa:Action>
  <wsa:MessageID>urn:uuid:6a1d3e0c-c665-486b-943c-2c994597ff13</wsa:MessageID>
  <wsa:ReplyTo>
<wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
  </wsa:ReplyTo>
  <wsse:Security soap:mustUnderstand="1">
   <wsu:Timestamp wsu:Id="ID-Timestamp">
    <wsu:Created>2007-09-27T21:11:07Z</wsu:Created>
    <wsu:Expires>2007-09-27T21:16:07Z</wsu:Expires>
   </wsu:Timestamp>
   <wsse:BinarySecurityToken wsu:Id="ID-X509Certificate"
       ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
       EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
        MIIB2DCCAUWgAwIBAgIBBDAJBgUrD....8=
   </wsse:BinarySecurityToken>
   <xenc:EncryptedKey Id="SecurityToken-461f8c64-ecfd-4c19-a728-062811136238">
    <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
     <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    </xenc:EncryptionMethod>
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
     <wsse:SecurityTokenReference>
      <X509Data>
       <X509IssuerSerial>
        <X509IssuerName>CN=EMIC SAFe CA</X509IssuerName>
        <X509SerialNumber>2</X509SerialNumber>
       </X509IssuerSerial>
      </X509Data>
     </wsse:SecurityTokenReference>
    </KeyInfo>
    <xenc:CipherData>
     <xenc:CipherValue>
      M5L/7....
     </xenc:CipherValue>
    </xenc:CipherData>
    <xenc:ReferenceList>
     <xenc:DataReference URI="#ID-EncryptedDataInBody" />
    </xenc:ReferenceList>
   </xenc:EncryptedKey>
```

```
 <ds:Signature>
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
/>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
          <ds:Reference URI="#ID-SoapBody">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>HSqkTgHz1DmZQQt/KVxzWVMmpjA=</DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>     HGrSh0SUQgp....     </ds:SignatureValue>
        <ds:KeyInfo>
          <wsse:SecurityTokenReference>
            <wsse:Reference
                URI="#ID-X509Certificate"
                ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-
token-profile-1.0#X509v3" />
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </ds:Signature>
    </wsse:Security>
  </soap:Header>


  <soap:Body wsu:Id="ID-SoapBody">
    <xenc:EncryptedData Id="ID-EncryptedDataInBody"
Type="http://www.w3.org/2001/04/xmlenc#Content">
      <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc" />
      <xenc:CipherData>
        <xenc:CipherValue>     qT9APGutRN1fBaXDflaHTpbQMT...   </xenc:CipherValue>
      </xenc:CipherData>
    </xenc:EncryptedData>
  </soap:Body>
</soap:Envelope>
```

WS-Federation sample



1. Client fetches service policy
2. Client fetches FedSTS policy
3. Client requests ID token
4. Client requests Fed token
5. Client invokes service
6. (Service may ask for authz decision)

- Fetch policy using WS-MetadataExchange
- Request security tokens using WS-Trust
- Protect messages using WS-Security

Vielen Dank für die Aufmerksamkeit

christian.geuer-pollmann@microsoft.com