Web Services Advanced Topics

Wokflows & Web Services
Kapitel 4

Security

Web Services Security

- Protect resources such that only appropriate "entities" can access them
  - Authorization: decide whether an identity can access a particulare resource
- Ensure the safety of information exchange among trading partners
  - Confidentiality: protection against eavesdroppers
  - Authentication: provide/verify proof of identity
  - Integrity: message was not modified accidentally or deliberately in transit
  - Non-repudiation: sender of message cannot deny he/she sent it
- Cryptography is used to protect the information exchange
  - Transport Security
    - Basic authentication, SSL
  - Web Service Security
    - Digital Signature, Encryption, …

Transport Security

- HTTP Basic Authentication
  - UserID, Password authentication on the web
    - Initial HTTP request results in error "401 Unauthorized"
    - Browser opens dialog to request user, password info, resubmits the request
      - Userid/password are encoded in Base64, NOT encrypted
  - Web server verifies permissions based access control list (ACL)

Transport Security (2)

- Secure Sockets Layer (SSL)
  - Protocol for transmitting data in a secure way
    - Can provide confidentiality, authentication, integrity
  - Located between application layer and transport layer (TCP)
    - Other protocols can be performed over SSL
      - HTTPS is HTTP over SSL
  - Supports server authentication and client authentication
    - The latter is rarely used, requires client to possess a certificate issued by a certificate authority
  - Uses public key cryptography (asymmetrical key cryptography)
    - Public key, private key pairs
    - Sender uses public key of the receiver to encrypt the message
    - Receiver can decrypt the message only using the private key

Web Services Security

- Digital Signatures
  - Needed to prove that the sender actually sent the message (non-repudiation)
  - XML Digital Signature
    - W3C specification
- Encryption
  - Encrypt (parts of a) message in a flexible manner
  - XML Encryption specification
- Web Services Security (WS-Security) specification
  - Initially drafted by Microsoft, IBM, Verisign
  - OASIS as standardization forum
  - Defines a set of standard SOAP extensions for building secure web services
    - Leverages XML Encryption, XML Digital Signature, …

SOAP Signature Details

- How do digital signatures work?
  - A hash function is applied to the data
  - The resulting hash value is encrypted with the private key of the signer, producing the signature
  - To verify the signature, anyone with access to the public key of the signer can
    - Decrypt the signature (original hash) using the public key
    - Apply the hash function to the original data
    - Compare the two hash values to make sure they are identical
- XML Digital Signature
  - Defines a `Signature` element with its descendents to store
    - Information about the hashing and encryption algorithms used
    - Signature itself
    - Public key to verify the signature
      - Or address of PK directory that includes the key
  - XML Canonicalization is used to produce canonical form before signing
- WS-Security specification
  - Defines how to embed the `Signature` element in a SOAP message as a header entry
  - Possible to sign whole message, parts of the message, attachements
    - Multiple signatures in the same SOAP message supported

SOAP Encryption

- Problems with SSL for SOAP messaging
  - SSL assumes that communication occurs directly between to parties
    - SOAP messaging may include third-party intermediaries that need to read the message
  - SSL encrypts the whole message
    - One might want to encrypt only parts of a SOAP message (e.g., the body)
- XML Encryption
  - Defines EncryptedData element to hold
    - Information about the encryption method
    - Key information
      - Name of secret shared key, public key, …
    - Encrypted data
- WS Security
  - Defines Encryption element/header
    - Includes reference to encrypted data
    - Can be directed towards specific intermediary
  - Multiple encryption elements in the same SOAP message supported

Related Efforts

- Decryption Transform for XML Signature
  - Enables signature verification even if both signature and encryption operations are performed on an XML document
- XML Key Management Specification (XKMS)
  - Specifies protocols for distributing and registering public keys
- eXtensible Access Control Language (XACML)
  - Defines an XML Schema for an extensible access control policy Language
- Security Assertion Markup Language (SAML)
  - XML security standard for exchanging authorization and authentication information

Security Assertions

- Security Assertion Markup Language (SAML)
  - XML standard for transporting security information between online commerce systems
    - Implement a single sign-on mechanism
  - Allows web sites and services to share information about a user
    - "entitlement" information
      - Credit limits, gold card profiles, …
    - Registration information
- Various security assertions
  - Authentication, attribute, decision
- Assertions are produced by their respective authorities
  - Example
    - Client sends request including userid and password to authority
    - Authority issues document containing authentication and attribute assertion (e.g., company ranking)
    - Client sends purchase order (request) to web service, attaching the security assertion
    - Service performs authorization, relying on the assertion

Invocation, Description, and Discovery Extensions

Discovering Web Services

- Sometimes you don't want to register (yet) a Web Service in UDDI
  - It may not be of public interest
  - It may not be ready for production
  - ...
- Thus, we need a language to discover Web Services at Web sites
- Web Services Inspection Language (WSIL)
  - Proposed by IBM and Microsoft (11/2001)
  - Supported by toolkits
    - Apache's Axis project
    - ...

WSIL Documents

- A single inspection document (.wsil) may reference multiple service descriptions
- A single service may be described by more than one description
  - Service description is a .wsdl file or a reference to UDDI or plain HTML
    - Even elements from a WSDL file can be referenced
- Thus, inspection document convenient way to aggregate different informations about a Web Service
- Each Web site may store an inspection.wsil file at a common entry point for service descriptions
  - Allows to discover all Web Services supported by this Web site
- A new META tag called serviceInspection may be added to an HTML file
  - Allows to discover all Web Services supported by this Web page
  - Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<META name="serviceInspection"
    content= "http://example.com/inspection.wsil"/>
</head>
    ...
</html>
```


Sample Inspection Document

```
<?xml version="1.0"?>
<inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/">
 <service>
   <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
      location="http://example.com/exampleservice.wsdl" />
</service>
<service>
  <description referencedNamespace="urn:uddi-org:api">
    <wsiluddi:serviceDescription
       location= "http://example.com/uddi/inquiryapi">
       <wsiluddi:serviceKey>
          52946BB0-BC28-11D5-A432-0004AC49CC1E
       </wsiluddi:serviceKey>
    </wsiluddi:serviceDescription>
  </description>
</service>
<link referencedNamespace= "http://schemas.xmlsoap.org/ws/2001/10/inspection/"
  location="http://example.com/tools/toolservices.wsil"/>
</inspection>
```

Reference to WSDL file

Reference to UDDI entry

Reference to WSIL file

Referencing WSDL Elements

```
<?xml version="1.0"?>
<inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/">
<service>
<name xml:lang="en-US">StockQuoteService</name>
<description referencedNamespace="http://schemas.xmlsoap.org/wsdl/">
<wsilwsdl:reference
    endpointPresent="true"
    location="http://localhost:8080/webservices/wsdl/stockquote/sqs.wsdl">
    <wsilwsdl:referencedService
        xmlns:tns="http://www.getquote.com/StockQuoteService">
        tns:StockQuoteService
    </wsilwsdl:referencedService>
    <wsilwsdl:implementedBinding
        xmlns:interface="http://www.getquote.com/StockQuoteService-interface">
        interface:StockQuoteServiceBinding
    </wsilwsdl:implementedBinding>
</wsilwsdl:reference>
</description>
</service> .
</inspection>
```


Web Service Invocation

- How do I easily invoke RPC-based web services in my Java application?
- Java API for XML RPCs (JAX-RPC)
  - APIs for supporting XML based RPC for the Java platform
    - Define web service
    - Use web service
  - Defines
    - WSDL/XML to Java mapping
    - Java to XML/WSDL mapping
    - Core APIs
    - SOAP support (including attachments)
    - Client and Server Programming models involving generated stub classes
- Client side invocation (standard programming model)
  - Application invokes web service through generated stub class
  - JAX-RPC runtime maps the invocation to SOAP, builds the SOAP message, processes the HTTP request
- Server side processing
  - JAX-RPC runtime processes HTTP, SOAP message, maps to RPC and dispatches to target (class implementing the web service)

JAX-RPC

- Mapping WSDL to Java – Example
  - WSDL port type definition

```
<!-- WSDL Extract -->
<message name="getLastTradePrice">
    <part name="tickerSymbol" type="xsd:string"/>
</message>
<message name="getLastTradePriceResponse">
    <part name="result" type="xsd:float"/>
</message>
<portType name="StockQuoteProvider">
    <operation name="getLastTradePrice"
      parameterOrder="tickerSymbol">
      <input message="tns:getLastTradePrice"/>
      <output message="tns:getLastTradePriceResponse"/>
    </operation>
</portType>
```

  - Corresponding Java service endpoint interface:

```
//Java
public interface StockQuoteProvider extends java.rmi.Remote {
    float getLastTradePrice(String tickerSymbol)
      throws java.rmi.RemoteException;
}
```


Web Service Invocation (2)

- Web Services Invocation Framework (WSIF)
  - WSIF provides a unified programming model for services based on WSDL
    - Allow the client to invoke a web service without needing to know the protocol-specific API details.
    - Enable run-time selection ("plugging") of service bindings: supports dynamic discovery and optimization.
  - Example

```
WSIFService sq = ServiceFactory.newInstance().
          getService("http://my.com/svcs/stockquote.wsdl");
MyService mySvcStub = sq.getStub("soap", MyService.class);
mySvcStub.myMethod();
```

  - Initially developed by IBM
    - http://www.alphaWorks.ibm.com/tech/wsif
  - Donated to Apache Software Foundation

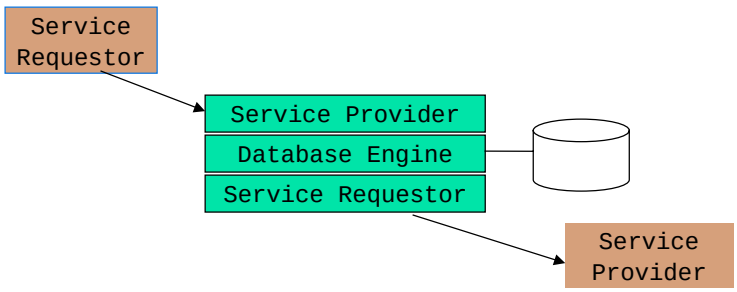
WS Endpoint Description

- WSDL only covers functional description
- Complete description of a service with operational description of service behavior includes:
  - QoS characteristics
  - Sequencing constraints
  - Transactional and conversational semantics
  - Encryption, authentication, security
  - Pre-/Post-conditions
- Web Services Endpoint Language (WSEL)
  - Annotate component descriptions with non-functional characteristics
  - WSEL is an open problem (some of the problems are very hard)

Databases and Web Services

Databases and Web Services

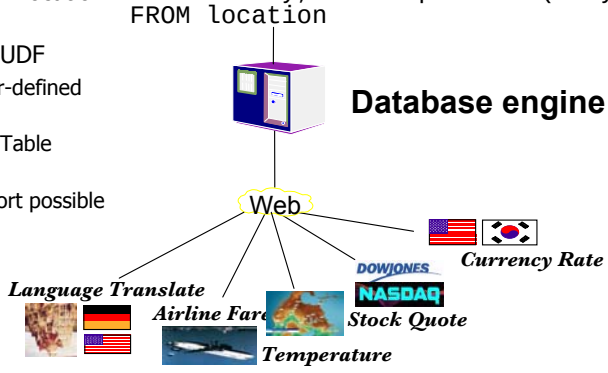
- Information Integration and dissemination
- Database as web service requestor
  - Invoking web services on my data
- Database as web service provider
  - Offering my data as service (making it easy)



Database – Web Service Requestor

- Information integration
  - Enables new applications
- Web service invocation in engine
- Web Service UDF
  - SOAP User-defined Function
  - Scalar vs. Table Functions
  - Tool support possible

```
SELECT city, GetTemperature(city)
FROM location
```



Database – Web Service Provider

- SQL-based database web service
  - ability to send SQL to database and return results with default tagging (includes calls to stored procedures)
  - focus is data in and out of database rather than the format
- XML-based database web service
  - Using DBMS-specific XML plug-ins engine support
  - Compose and decompose XML documents

Example

- DB2 as an SQL-based web service provider

```
<?xml version="1.0" encoding="UTF-8"?>
<DADX xmlns=http://schemas.ibm.com/db2/dxx/dadx>
  <operation name="showemployees">
    <query>
      <SQL_query>SELECT * FROM EMPLOYEE</SQL_query>
    </query>
  </operation>
</DADx>
```

- DADx file (Document Access Definition Extension) contains definition of operations and corresponding data access statements to implement them
  - SQL, including stored procedure invocation
- WS tooling/runtime generates the corresponding web services, performs default tagging of results
- Can invoke DB2 XML extender functionality to perform composition/decomposition in a user-defined manner