Prof. Dr.-Ing. Stefan Deßloch
AG Heterogene Informationssysteme
Geb. 36, Raum 329
Tel. 0631/205 3275
dessloch@informatik.uni-kl.de

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

# Chapter 17
# Web Services – Additional Topics



Prof. Dr.-Ing. Stefan Deßloch
AG Heterogene Informationssysteme
Geb. 36, Raum 329
Tel. 0631/205 3275
dessloch@informatik.uni-kl.de

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

# Security

Middleware for Heterogenous and Distributed Information Systems - WS05/06

# 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/TLS
  - Web Service Security
    - Digital Signature, Encryption, …

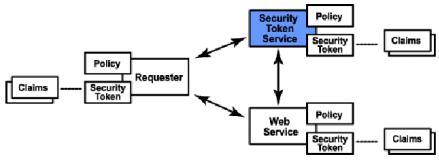


# Web Services and SSL/TLS

- SSL/TLS can be used for transmitting SOAP messages
  - SOAP/HTTPS
- Problems with SSL/TLS for SOAP messaging
  - SSL assumes that communication occurs directly between two parties
    - SOAP messaging may include third-party intermediaries
  - SSL encrypts the whole message
    - not possible to encrypt only parts of a SOAP message (e.g., the body)
  - SSL does not support digital signing of (parts of) the SOAP message

# Web Services Security Model

- End-to-end security
- General Model
  - WS can, as part of its *policy*, require proof of a *set of claims* from a requester
    - name, key, permission, capability
  - A requestor can provide proof of claims with a message by attaching a *security token*
    - e.g., X.509 certificate, Kerberos ticket, …
  - Requestor may try to obtain required claims from *security token services*






# Web Service Security

- Initially industry proposal, standardization by OASIS
- WS-Security
  - SOAP extensions (headers)
  - focus on WS integrity and confidentiality
    - pass security tokens, sign and encrypt messages
  - mechanisms to be used with other extensions, higher-level protocols for complete security solution
  - Leverages XML Encryption, XML Digital Signature, …
- WS-Security does not attempt to address interoperability across different security infrastructures and trust domains
  - how to make sure that partners understand and support each others security policies (e.g., which kind of security tokens are used, …)
  - this is left for other specifications to solve

# SOAP Signature Details

- 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

- 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

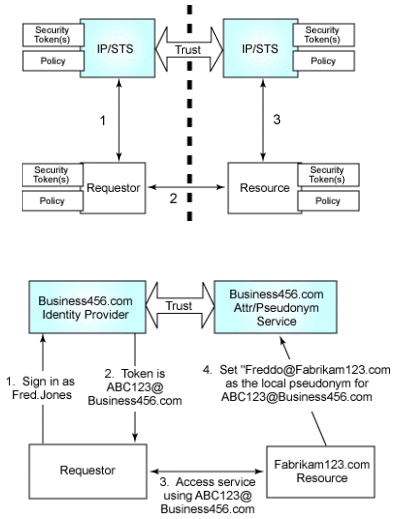
# Policies

- Interoperability, step 1
  - ability to express how you implement security, what you expect from a service partner
- WS-Policy
  - express capabilities, characteristics of entities in a WS-based system
    - authentication scheme
    - transport protocol
    - privacy policy
    - Quality-of-Service characteristics
  - policy assertions, expressions, statements
  - allows senders, receivers to specify their security requirements and capabilities
- WS-PolicyAttachment
  - associate policy expressions with subjects
    - reference policies from WSDL definitions
    - associate policies with UDDI entities



# Trust

- Interoperability, step 2
  - ability of a service partner to request from a recognized authority that a particular security token is exchanged for another
  - establish chain of trust
- WS-Trust
  - security token service (STS)
  - request/obtain security token
  - manage trusts, establish and assess trust relationships
    - build a chain of trust from recipient's trust authority to the sender authority
- WS-Federation
  - extends the WS-Trust model to allow attributes and pseudonyms to be integrated into the token issuance mechanism
    - provide federated identity mapping mechanisms
    - facilitate single sign-on



# Additional Efforts

- WS-SecureConversation
  - describes how to manage and authenticate message exchanges between parties including security context exchange and establishing and deriving session keys
- Still to come as part of the web services security stack
  - WS-Privacy: will describe a model for how Web services and requesters state privacy preferences and organizational privacy practice statements
  - WS-Authorization: will describe how to manage authorization data and authorization policies
- 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 communicating 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

Prof. Dr.-Ing. Stefan Deßloch
AG Heterogene Informationssysteme
Geb. 36, Raum 329
Tel. 0631/205 3275
dessloch@informatik.uni-kl.de

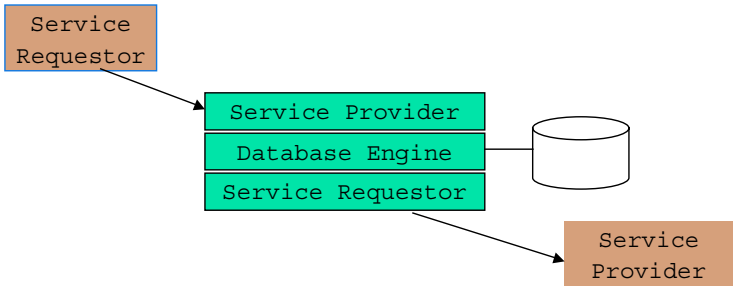
TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

# 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)



# Databases and Web Services

- DBMS as a web service provider
  - offer DB operations as web service
    - query, update, invoke a routine, …
  - "speak" XML
    - natively
    - translated
- DBMS as a web service consumer
  - invoke a WS through query/DML statement or as a side-effect of updates
  - process and analyze WS results inside query engine
  - provide integration services



# SQL/XML

- Goal: standardization of interaction/integration of SQL and XML
  - how to represent SQL data (tables, results, ...) in XML (and vice versa)
  - how to map SQL metadata (information schema) to XML schema (and vice versa)
- Potential areas of use
  - "present" SQL data as XML
  - integration of XML data into SQL data bases
  - use XML for SQL data interchange
  - XML views over relational
    - possible foundation for XQuery
- Example
  - SQL table "EMPLOYEE"
  - XML document:

```
<EMPLOYEE>
  <row>
    <EMPNO>000010</EMPNO>
    <FIRSTNME>CHRISTINE</FIRSTNME>
    <LASTNAME>HAAS</LASTNAME>
    <BIRTHDATE>1933-08-
  24</BIRTHDATE>
    <SALARY>52750.00</SALARY>
  </row>
  <row>
    <EMPNO>000020</EMPNO>
    <FIRSTNME>MICHAEL</FIRSTNME>

  <LASTNAME>THOMPSON</LASTNAME>
    <BIRTHDATE>1948-02-
  02</BIRTHDATE>
    <SALARY>41250.00</SALARY>
  </row>
  ...
</EMPLOYEE>
```

# DBMS as a Web Service Provider

- Mapping for tables, schemas, catalogs to XML
  - no default mapping of arbitrary SQL query results in SQL standard
- No standard way of publishing queries, routine invocations, etc. as a web service
  - left to tooling provided by DBMS vendors
  - 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
- No standard set of web services for interacting with SQL or XML databases at the general API level
  - see ongoing work in data grid area



# 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

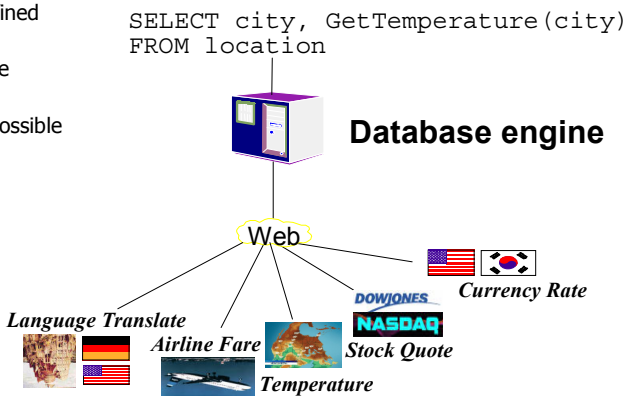
# DBMS as a Web Service Consumer

- Use SQL MED
  - web service result as one or more SQL tables
    - alternative: foreign routine
  - foreign data wrapper
    - invokes web service
    - maps (parts of) result from XML to SQL tables
    - challenge: support complex input parameters for WS
- Use SQL user-defined routines
  - web service as stored procedure
    - SP paradigm may not be adequate for further result processing
  - web service as user-defined (scalar or table) function
    - result is limited to a single value (chunk of XML) or a single table



# Database – Web Service UDFs

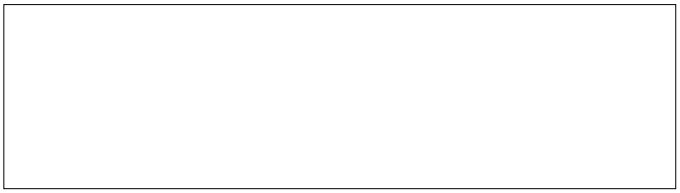
- Web service invocation in engine
- Web Service UDF
  - SOAP User-defined Function
  - Scalar vs. Table Functions
  - Tool support possible



Prof. Dr.-Ing. Stefan Deßloch
AG Heterogene Informationssysteme
Geb. 36, Raum 329
Tel. 0631/205 3275
dessloch@informatik.uni-kl.de

Technische Universität Kaiserslautern

# Grid Computing





## Grid Computing

- Primary goal
  - computing as a utility
    - provide shared computing resources
    - hide details of components
      - location, management, …
    - virtualization of services
- Web Services
  - can be used in a Grid architecture to provide grid services
- Grid Computing and Databases
  - increased focus on data-intensive applications
    - significant processing on verly large amounts of data
      - collaboration
      - scalability
  - Grid for data access and integration

# Global Grid Forum

- Open forum for standardizing grid interfaces
  - founded in 1998
  - produce technical specs that become grid recommendations
- Organized into topic areas, working/research groups
  - for example:
    - Architecture
      - Open Grid Services Architecture WG (OGSA)
      - Open Grid Services Infrastructure WG (OGSI)
    - Data
      - Data Access and Integration WG (DAIS)
      - OGSA Replication Services WG (OREP)
      - Data Format and Description Language WG (DFDL)
      - GridFTP WG (GridFTP)
      - Grid File System WG (GFS)



# OGSA

- Identifies
  - the components that make up the infrastructure of a grid computing environment
    - described as services
  - the basic mechanisms which must be supported by grid components
    - expressed as web services
    - defined by OGSI
- Platform interfaces for
  - service groups and discovery, service domains, security, policy, messaging and queuing, events, distributed logging, metering and accounting, administration, transactions, orchestration
  - data management
    - access, replication, caching, metadata, schema transformation, storage

# OGSI

- Grid service must expose web service interfaces conforming to OGSI spec (e.g., factory)
- Grid services have state
  - long-term information to be maintained across client requests
- Conventions for performing service-related activities
  - handle: refers to an instance of a service
  - referring to collections of instances as a whole
  - factory: starting up service
  - service data: accessing a service state
  - state change notification
  - service lifetime management
  - inheritance support for grid services

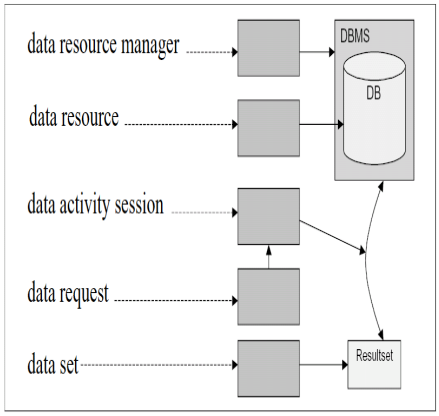


# DAIS – Data Access and Integration Services

- Service-based interface for accessing and integrating data on the grid
  - relational databases
  - XML databases
- Some features
  - naming results for subsequent use
  - multiple result formats
  - chunking large quantities of data
  - asynchronous result delivery
  - result delivery to third party
- Work in progress

# DAIS – Main Constructs

- Services
  - Data Resource Manager
    - DBMS
  - Data Resource
    - database (tables or collections of XML)
  - Data Access Session
    - relationship between client and data resource
- Data Formats
  - Data Request
    - SQL, XPath, XQuery
  - Data Set
    - output result format






# DAIS Topics

- DAIS model
  - see main constructs
- Transformations
  - transformation of results
- Stored Procedures
  - how parameters and result sets are handled
- Security
  - how database and grid security interact
- Transaction
  - transaction support in a grid environment
- Metadata
  - DBMS characteristics, database metadata, …