POZNAN UNIVERSITY OF TECHNOLOGY

Data Integration Problems and Architectures

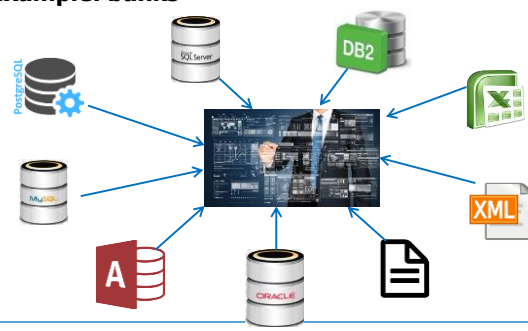
Robert Wrembel
Poznan University of Technology
Institute of Computing Science
Poznań, Poland
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel

Outline

- Data integration problems
  - system heterogeneity
  - data heterogeneity
- Approaches
  - virtual data integration
  - application integration
  - physical data integration

Need for data integration

- The cause of systems heterogeneity (large companies → multiple information systems):
  - adding new functionalities
  - changing technologies
  - contracting different software companies
  - purchasing smaller companies that have their own IS
- typical example: banks



Need for data integration

- Integrating external (own by others) data sources
  - airplane ticket purchasing systems
    - Momondo, SkyScanner
  - hotel booking systems
    - booking.com, hotels.com, trivago
  - tour operator systems (holiday packages)
    - Blue Sky Travel, Tui, Neckerman, DerTour, Meiers
- Online purchasing
  - Amazon, Google Shopping
- Accessing all important data gathered within the lifecycle of a company
  - transactional systems
  - analytical systems

Data integration problems

- **Data source features**
  - **geographically distributed**
  - **autonomous**
    - **managed independently**
      - separate users
      - turned off/of at any time
    - **may evolve independently**
      - structure (schema)
      - new software
  - **heterogeneous**

Heterogeneity of data sources

- **Different software producers**
  - **e.g., IBM, Oracle, Microsoft, Sybase, Teradata, ...**
- **Different implementation technologies**
  - **.Net, C++, C#, Java, PHP, Scala, ...**
- **Different functionality**
  - **databases / pseudo-databases / no-databases**
  - **SQL dialects**
  - **data access drivers**
  - **data access and processing techniques**

Heterogeneity of data sources

- **Different data models**
  - **hierarchical, network**
  - **relational**
  - **object**
  - **object-relational**
  - **multidimensional**
  - **semistructured (XML, JSON)**
  - **NoSQL**
  - **graph**
- **Different data types**
  - **smallint, int, biging, decimal (SQLServer)**
  - **smallint, int, bigint, float, real, double (DB2)**
  - **number, binary_integer (Oracle)**
  - **constant and variable length string data types**

Heterogeneity of data sources

- **Different data structures representing the same information**
  - **Car dealer A**
    - **table Vehicles {VId, make, model, engineType, engineCap, horsepower, vType, maxPersons, maxLoad}**
    - **stores trucks, SUVs, ...**
  - **Car dealer B**
    - **table Trucks {engineNo, make, model, engineType, maxLoad}**
    - **table SUVs {engineNo, make, model, engineType, engineCap, horsepower}**

Heterogeneity of data sources

- Missing and wrong data
- Different measurement units
  - price (EUR, GBP, USD, ...}
  - weight {pounds, kilograms, ...}
- Different abbreviations and symbolic values
  - GB, Great Britain, G.Britain, ...
  - prof. ⇨ professor, prof. ⇨ professional
- Mixed text and numeric values
  - typical for integrating data scrapped from web pages
  - 1K, 1 th., 1 thousand, one th., 1000, 1 000, 1,000.00
- Names in an original language transcripted into another language

Heterogeneity of data sources

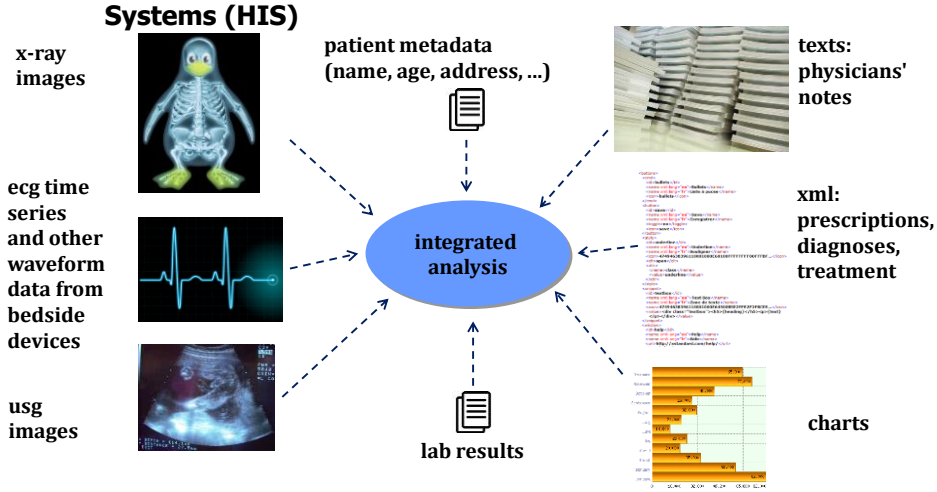
- Homonymes
  - Supplier.code ⇨ postal code
  - Product.code ⇨ bar code
- Synonymes
  - Patient.SSN
  - Patient.Id ⇨ storing the value of SSN
- Object names in different languages

Other problems

- Object names without semantics
  - e.g., columns Col1, Col2, ..., Col10
- Schemas composed of 10 000 - 20 000 tables
  - CRM, ERP, e.g., SAP
- Outdated or not existing documentation (typical)
- Duplicated data (typical)
- Missing or wrong data governance
  - no dictionary of common data
  - no central repository of master data → no master data management

Heterogeneity of source systems

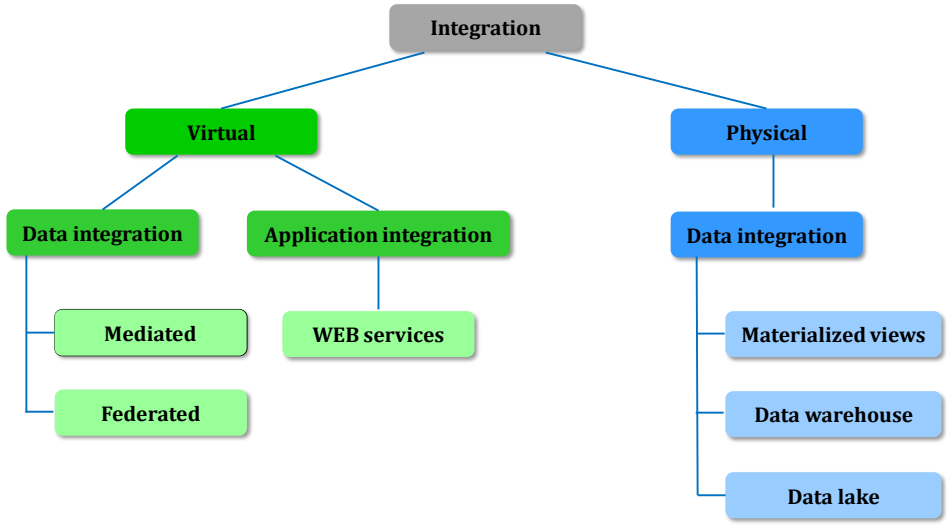
- Complex data ⇨ typical problem in Health Information Systems (HIS)



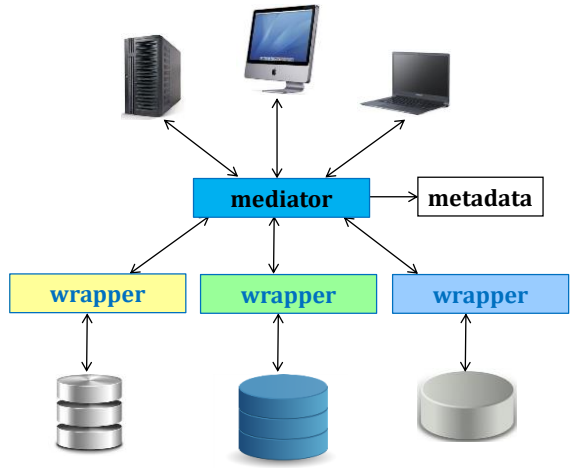
HIS

- Picture Archiving and Communication System (PACS)
  - architecture (technologies) for sending and storing images of different types
- Common format for storing and sending images → Digital Imaging and Communications in Medicine (DICOM)
- Image types
  - ultrasound (US)
  - magnetic resonance (MR)
  - positron emission tomography (PET)
  - computed tomography (CT)
  - endoscopy (ES)
  - mammograms (MG)
  - digital radiography (DR)
  - ...

Taxonomy of integration architectures (view 1)



Mediated



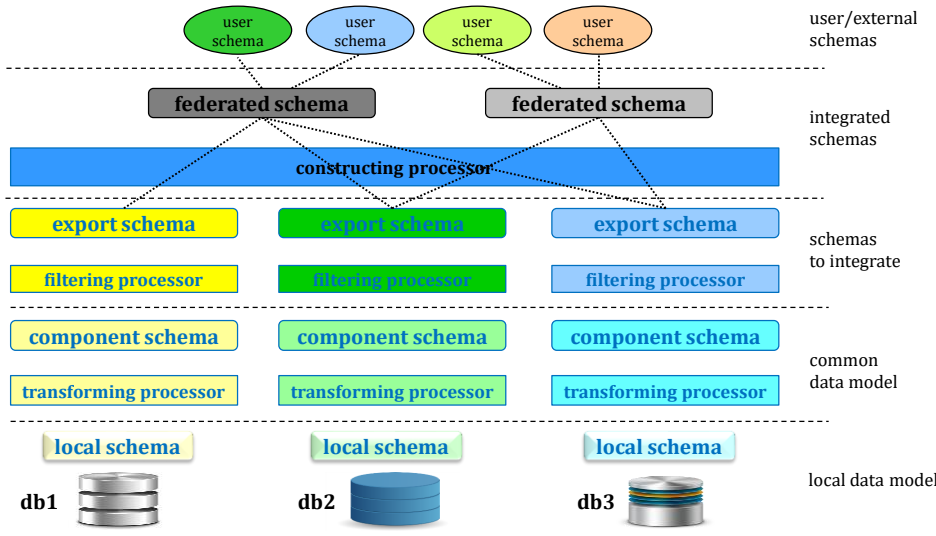
Mediated - querying

- Global query (GQ) sent to a mediator
- Mediator
  - decomposing QG into local queries (LQ)
  - query optimization (static or dynamic like in RDB)
  - sending LQ to data sources (wrappers)
- Wrapper
  - receiving LQ
  - translating LQ into a program/query executable in DS
  - sending LQ into DS for execution
- DS
  - executing LQ
  - preparing a result data set

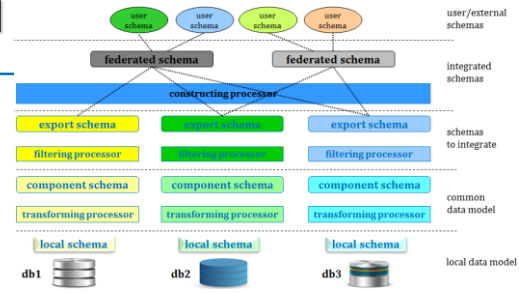
Mediated - querying

- DS
  - sending a result data set to a wrapper
- Wrapper
  - converting a data set to a common data model
  - sending a data set to a mediator
- Mediator
  - receiving data sets from wrappers
  - integrating data sets
  - sending an integrated data set to the user

Federated

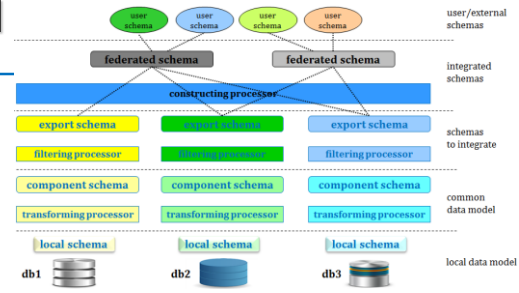


Federated



- **Transforming processor**
  - **maintaining mappings between local and component schema elements**
  - **translation of commands from a federated query language to a query language of a component database**
  - **translation of data from a local to common data format**
- **Filtering processor**
  - **controls the set of operations that are issued for a component schema using the information about data visibility and access control specified in an export schema**

Federated

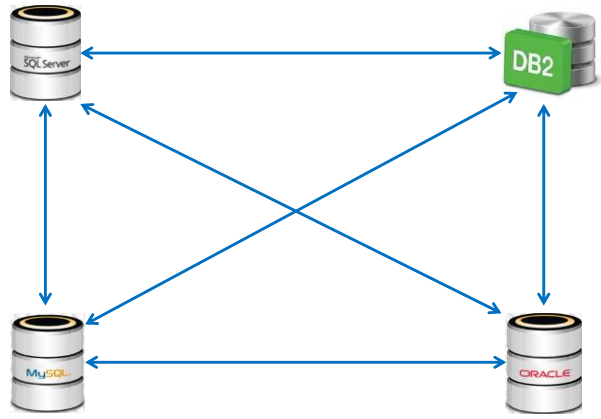


- **Constructing processor**
  - **integrating different data sources by resolving inconsistencies and conflicts between them**
  - **determining the set of data sources capable to answer a given query that was issued and formulated in terms of a federated schema**
  - **decomposing, optimising, and transforming the query into local queries, that is queries for each of the data sources**
  - **sending each local query to appropriate data source**
  - **receiving query results from data sources, translating, filtering, and merging these results to form a global result**

Federated vs. Mediated

- Mediated
  - DSs are more autonomous
  - applied to integrating an existing architecture
  - applied to integrating not only databases
  - one integrated schema
- Federated
  - DSs are less autonomous
  - applied to building an integration architecture from scratch
  - applied to integrating databases
  - typically all databases use the same data model (relational)
  - multiple federated schemas
  - multiple user schemas

Special case: P2P



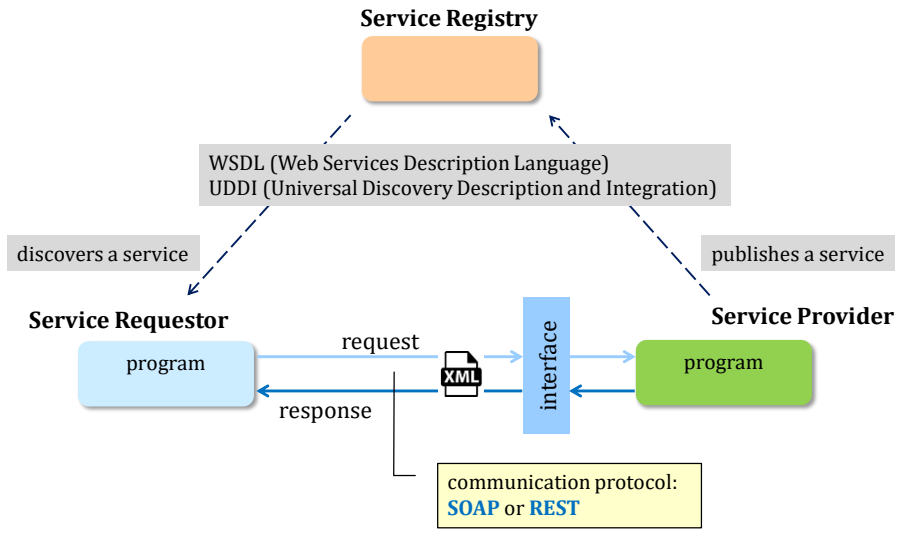
P2P in practice: Blockchain

- https://blockgeeks.com/guides/what-is-blockchain-technology/
- Any participant of a transaction has a copy of a ledger
- Any additions made to the ledger are sent to all participants

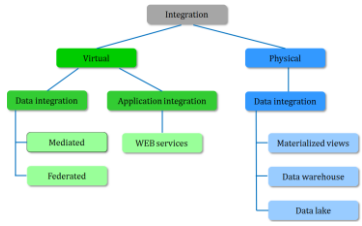
Web services

- Application integration technology
  - allows programs written in different languages on different platforms to communicate with each other in a standard-based way
- Exchange data between different applications and different platforms
- Program-to-program communication model, built on HTTP, XML, SOAP, WSDL, and UDDI
- Exchanged data format: XML or JSON

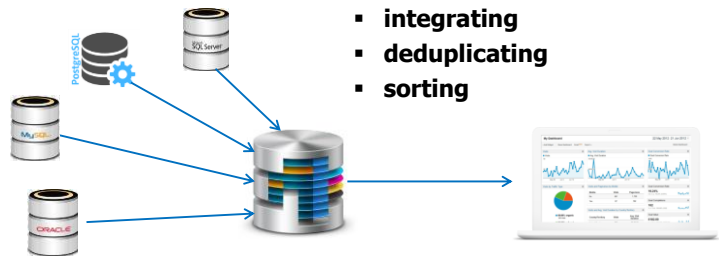
Web services



Physical data integration



- **Data retrieved from DSs are stored (materialized) in a local DB**
- **Steps**
  - **retrieving data**
  - **homogenizing**
  - **cleaning**
  - **integrating**
  - **deduplicating**
  - **sorting**



Virtual vs. physical integration

- Virtual → disadvantages
  - the results of a query may arrive with a long delay caused by a slow network, or a low response time of data sources
  - decomposition and translation of a query as well as merging the results of a query incur additional time overhead
  - queries coming from a federated / mediated system may interfere with queries executed locally in component databases, as a consequence, federated queries may slow down the execution of the local queries
  - some of the component data sources may be temporarily unavailable, thus making the query results incomplete or unavailable

Virtual vs. physical integration

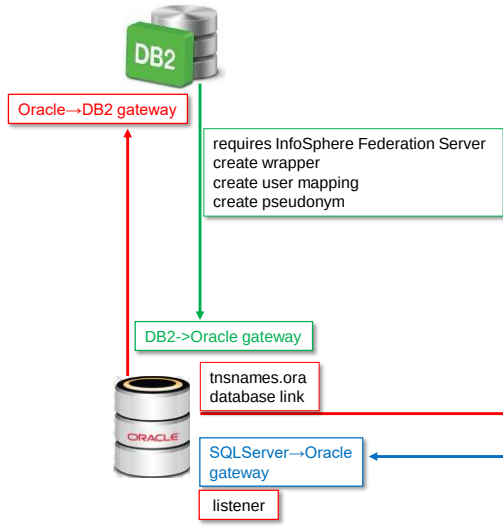
- Virtual → advantages
  - no data redundancy
  - access to up to date data
  - user can query any data that are available (unless limited by a global schema)

Taxonomy of integration architectures (view 2)

- **Federated** ----------> Mariposa, TSIMMIS, Multibase
  - multiple homogeneous storage (one data model)
  - one access interface (query language)
- **Polyglot** ----------> Spark
  - multiple homogeneous storage (one data model)
  - multiple access interfaces (SQL-like, procedural)
- **Multistore** ----------> HadoopDB, Polybase
  - multiple heterogeneous storage (multiple data models)
  - one access interface (query language)
- **Polystore** ----------> BigDAWG, Polypheny
  - multiple heterogeneous storage (multiple data models)
  - multiple access interfaces (SQL-like, procedural)

R.Tan, R. Chirkova, V. Gadepally, T.G. Mattson: Enabling Query Processing across Heterogeneous Data Models: A Survey. IEEE International Conference on Big Data, 2017

Accessing heterogeneous data sources



- **Wrappers/gateways**
  - translating SQL dialects
  - translating data types
  - transactional processing

Software

- SAP
  - BusinessObjects Data Federator
    - acces to relational and non-relational sources (Oracle, IBM, Microsoft, SAP NetWeaver Business Warehouse, SAS, Teradata, Web service, XML)
  - BusinessObjects Data Integrator - ETL
- IBM
  - InfoSphere Information Server
    - InfoSphere Federation Server
    - InfoSphere DataStage
    - InfoSphere Change Data Capture
    - InfoSphere Quality Stage

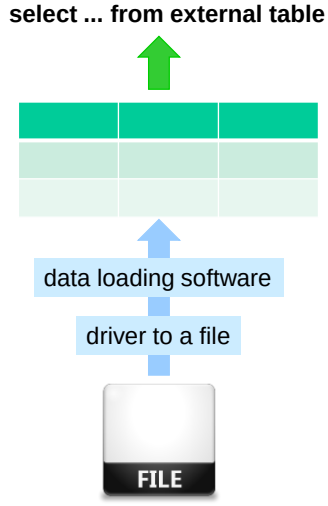
Software

- Oracle
  - Transparent Gateways
    - access to IBM DB2 and Informix, Sybase Adaptive Server Enterprise, MS SQL Server, Teradata
  - Warehouse Builder
  - Data Integrator
- Microsoft
  - SQL Server Integration Services
    - access to Oracle, XML, ODBC, OLE DB data sources

Accessing heterogeneous data sources

- ODBC
  - standardized access methods (API) for multiple data sources (databases, text files, dbf files, ...)
  - ODBC/JDBC driver ➔ API
  - OS: MS Windows, Unix, Linux, OS/2, OS/400, IBM i5/OS, Mac OS X
- OLE DB (Object Linking and Embedding DataBase)
  - API for accessing multiple data sources under Windows from COM-based programs (e.g., VB)
- Dedicated drivers for flat and i XML files
- JDBC
  - counterpart of ODBC for Java applications
- unixODBC
  - drivers available for most of the commercial and open source DBs

External table



- Does not have its own data
- Reads data from an OS file
- Indexing not allowed
- I, U, D not allowed
- Available in
  - Oracle
  - SQL Server
  - DB2 (implemented as a function that returns a table), part of InfoSphere Federation Server

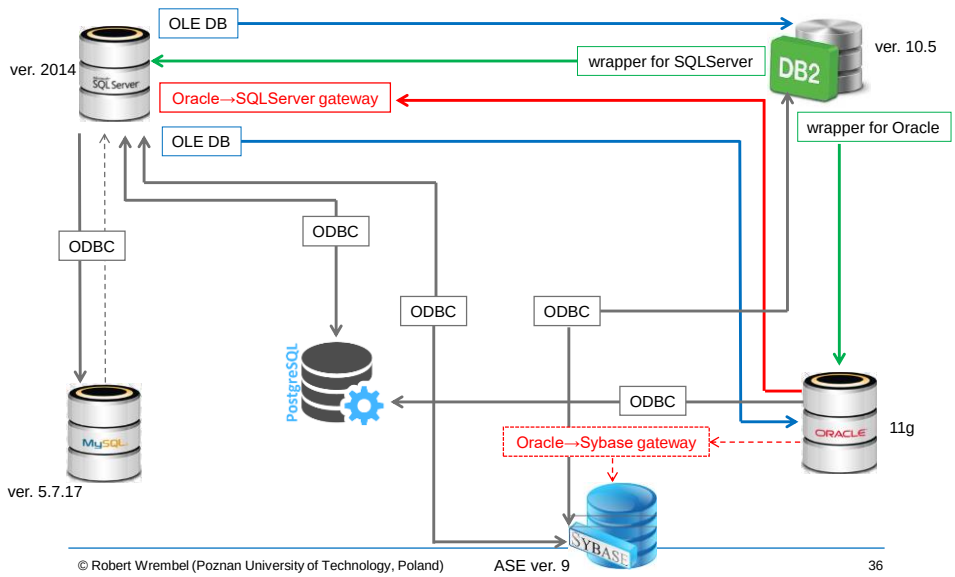
External table

```
create table ext_sales
 (ID number(6),
  Week_No number(2),
  ProdID number(6),
  Amount number(4))
organization external
 (type oracle_loader
  default directory exttab_sources
  access parameters
  ( records delimited by newline
    badfile 'bad-records.log'
    logfile 'log.log'
    fields terminated by ';'
    ( no char(4),
      week char(4),
      product char(4),
      items char(4)))
   location ('input_data.txt')
  )
parallel 2
reject limit 100;
```

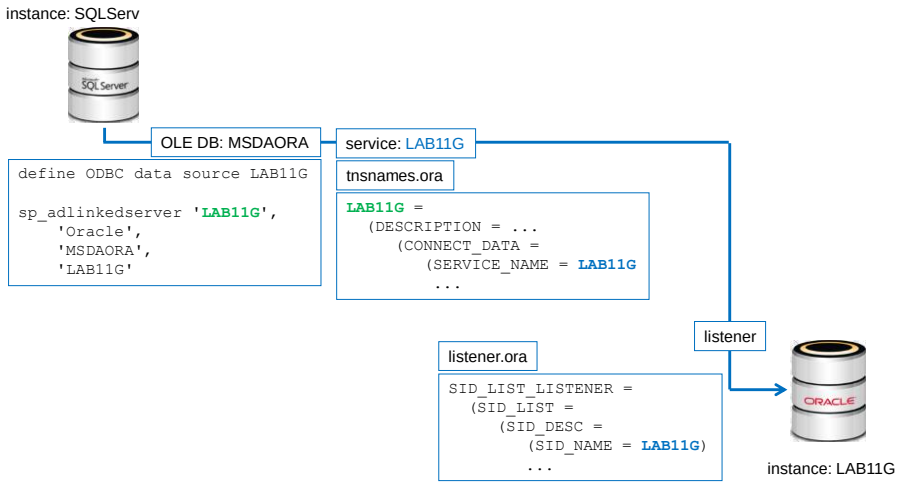
```
create or replace directory
    exttab_sources as 'c:\extTab1';
```

- **LOAD WHEN column=value**
- **SKIP n**
- **accepting or rejecting records with nulls**

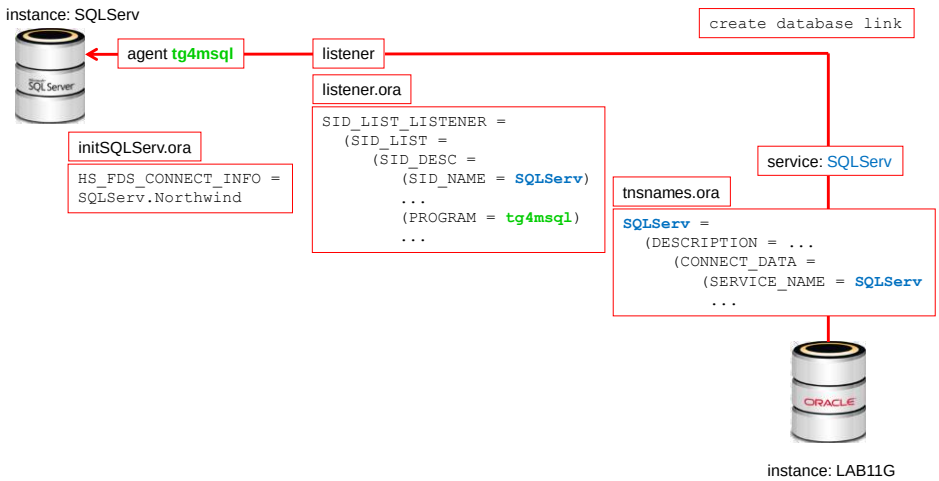
Appendix: Example test architecture



Appendix: Example test architecture



Appendix: Example test architecture



Appendix: Possible problems

- SQL SERVER → ORACLE
  - insert into a table in Oracle must contain values for all the attributes in the table, even if some attributes may have NULL values
- ORACLE → SQL SERVER
  - inserts from Oracle into an SQL Server table lock the whole table → in SQL SERVER even data reads are impossible
- ASA → SQL SERVER
  - translation error (in queries) into type decimal in ASA
  - solution: replace decimal with real
- SQL SERVER → DB2
  - insert and update from DB2 may result in error

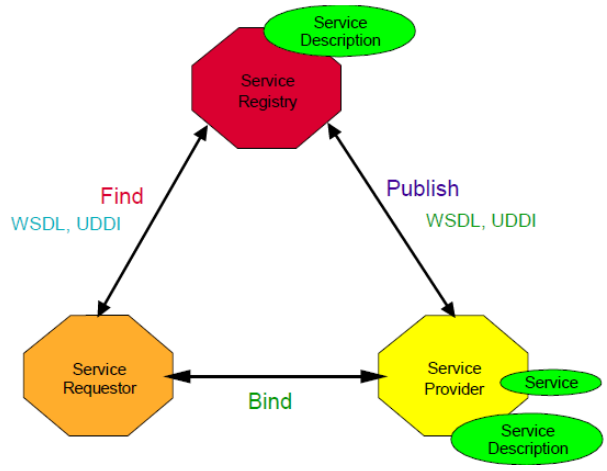
Appendix: Possible problems

- General
  - names of DB objects and attributes are case sensitive
  - only particular gateway / driver versions work with particular DBMS versions
  - MySQL do not support accessing external heterogeneous DBs
    - MySQL offers Federated Storage Engine for building federations of MySQLs
    - extensions for Oracle, DB2, and SQL Server → do not work, not supported for recent versions of MySQL

Appendix: IBM InfoSphere Federation Server

- **Wrapper**
  - **set of predefined wrappers (implemented as libraries)**
  - **user-implemented wrappers (C++, Java)**
- **Server**
  - **defines external data source (connection parameters, generating query execution plan)**
- **User mapping**
  - **required for external DBMSs**
  - **maps a local to an external user**
- **Nickname**
  - **local name for a remote object**

Appendix: Web services



Web Services Conceptual Architecture (IBM)

Appendix: Web services

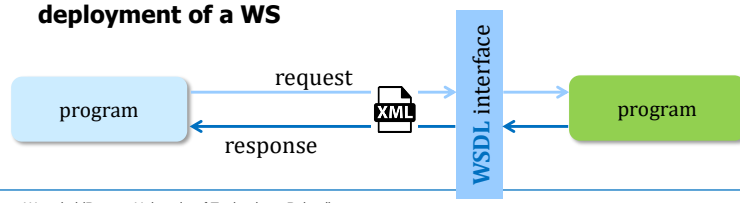
- **Service provider**
  - **makes available a software (service)**
  - **provides the service description** (data types, operations, binding information and network location)
  - **publishes the description in a service registry (XML format)**
- **Service requestor**
  - **finds and retrieves a service description**
  - **uses the service description to: bind with the service provider and to invoke or interact with the Web service implementation**
- **Service registry**
  - **searchable registry of service descriptions where service providers publish their service descriptions**
  - **service requestors find services and obtain binding information for services during development**

Appendix: Web services

- **Communication (transport) protocol: Simple Object Access Protocol (SOAP)**
  - **XML-based**
  - **used to exchange data over HTTP**
  - **defines**
    - **an XML format for messages → how to represent data**
    - **the format of an HTTP message that contains a SOAP message**
  - **independent of application programming languages**
  - **in practice, SOAP messages are created and parsed by various toolkits that translate function calls from a programming language to a SOAP message:**
    - **Microsoft SOAP Toolkit translates COM function calls to SOAP**
    - **Apache Toolkit translates JAVA function calls to SOAP**

Appendix: Web services

- **Interface description of a service: Web Services Description Language (WSDL)**
  - **to allow an application to communicate with the service**
  - **typically an XML document that describes a set of SOAP messages**
  - **the interface includes among others**
    - **WSDL:portType → defines method signatures of a Web service (what XML messages can appear in the input and output data flows)**
    - **WSDL:message → specifies XML data types for various parts of a message (e.g., input and output parameters of an operation)**
  - **in practice generated and parsed by software during the deployment of a WS**



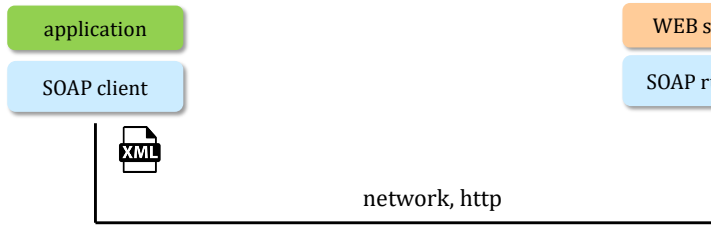
Appendix: Web services

- **Directory of Web services: Universal Discovery Description and Integration (UDDI)**
  - **allows to discover a service in the Internet**
  - **a registry of all web service's metadata, including WSDL descriptions of the services**
  - **a set of WSDL definitions for manipulating and searching the registry**
  - **can be implemented as and XML file**
  - **also includes**
    - **info about a company offering the service (e.g., name, address, contacts)**
    - **industrial categories based on standard taxonomies**

Appendix: Web services interaction

- Application creates a SOAP message (XML) - a request that invokes the Web service operation at the service provider
- SOAP client runtime sends the message to the provided network address using a network protocol (e.g., HTTP)

- The service processes the message and produces response that is encoded into SOAP message (XML)
- The message is sent to the requestor



- The message is received, converted into native application objects (in a target programming language)
- The converted message (data) is used in the application

- The message is delivered to the service provider's SOAP runtime (server)
- The runtime converts the message into programming language-specific objects and routes the request message to the service provider's Web service

Appendix: Using Web service

- **WSs deployment environment is needed**
- **Implement a web service (e.g., a java class)**
- **Compile and deploy the class in a WSs runtime server →**
  - **WSDL file is generated**
  - **the service is available at** http://address:port/service_name
- **Implement a WS client (e.g., java program)**
  - **create a WebServiceLookup object**
  - **the object is used to create a Web Service proxy (by invoking the lookup method)**
  - **the lookup method requires**
    - **a reference to a WSDL file (service address)**
    - **a class that will reference the proxy instance**
  - **the lookup method returns the proxy that is used to invoke the Web Service**