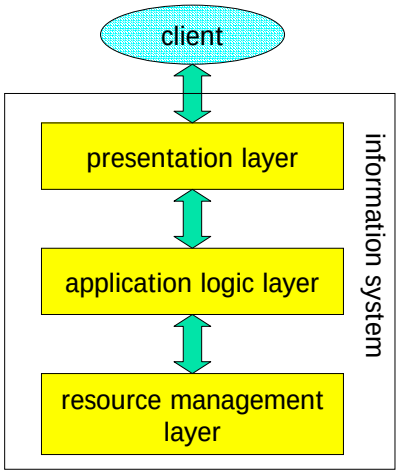
Prof. Dr.-Ing. Stefan Deßloch
AG Heterogene Informationssysteme
Geb. 36, Raum 329
Tel. 0631/205 3275
dessloch@informatik.uni-kl.de

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Chapter 1 – Motivation

Layers of an Information System

- Separation of functionality into three conceptual layers
  - presentation
  - application logic
  - resource (e.g., data) management
- Architecture of an IS
  - layers can be combined and distributed in different ways
  - 1-tier, 2-tier, 3-tier, n-tier



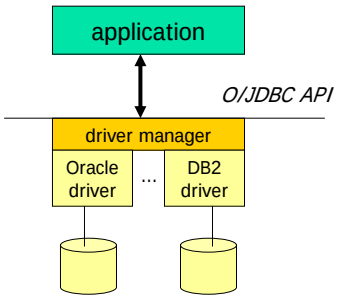
Middleware

- Middleware
  - supports the development, deployment, and execution of complex information systems
  - facilitates interaction between and integration of applications

  across multiple distributed, heterogeneous platforms and data sources
- Two major aspects
  - middleware as a programming abstraction
  - middleware as infrastructure
- Principles
  - Make distribution transparent
  - Support standardized APIs/languages/data formats to overcome platform heterogeneity
  - Transform data and/or operations/requests to bridge structural and semantic heterogeneity
  - Application logic independent from infrastructure code
  - Powerful programming abstractions

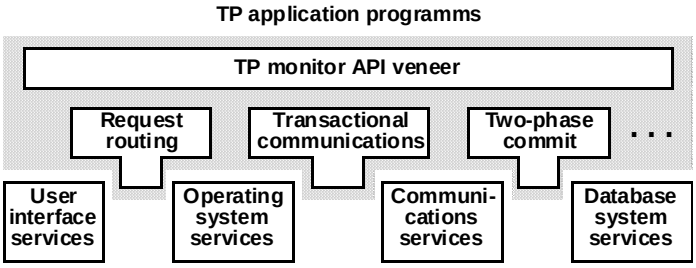
Database Gateways

- Uniform Database Access
  - Query Language (SQL)
  - Meta data
  - Programming Interface
- Dynamic, late binding to specific DB/DBS
  - call level interface (CLI)
    - no vendor-specific pre-compiler
  - dynamic binding of run-time libraries
  - late query compilation
- Simultaneous access to multiple DB/DBMS
  - architecture supports use of (multiple) DBMS-specific drivers
  - coordinated by a driver manager
- Support for vendor-specific extensions

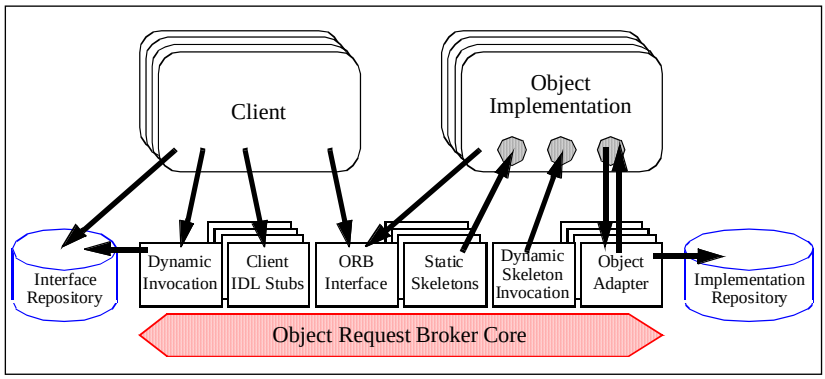


TP-Monitor Tasks

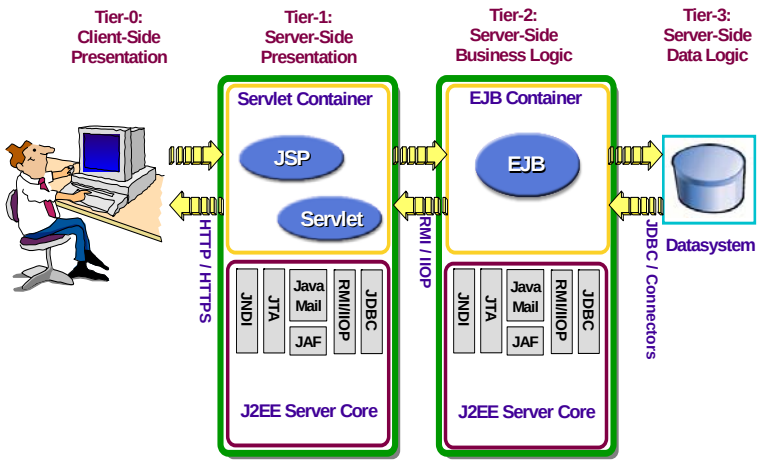
- Bridging heterogeneity
- Controlling communication
- Terminal management
- Presentation services
- Context management
- Start/Restart
- Program management/execution
- Configuration management
- Load balancing
- Authorization
- Providing administration capabilities



CORBA – Distributed Object Computing

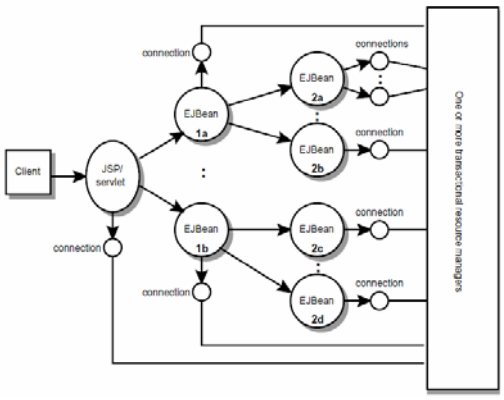


4-Tier Distributed Computing in J2EE



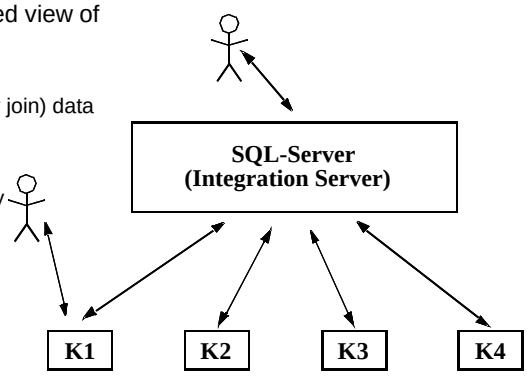
Important Services for Distributed IS

- Transactions
  - explicit
  - implicit/declarative
- Data Access
  - persistence
  - relationships
  - query
- Security

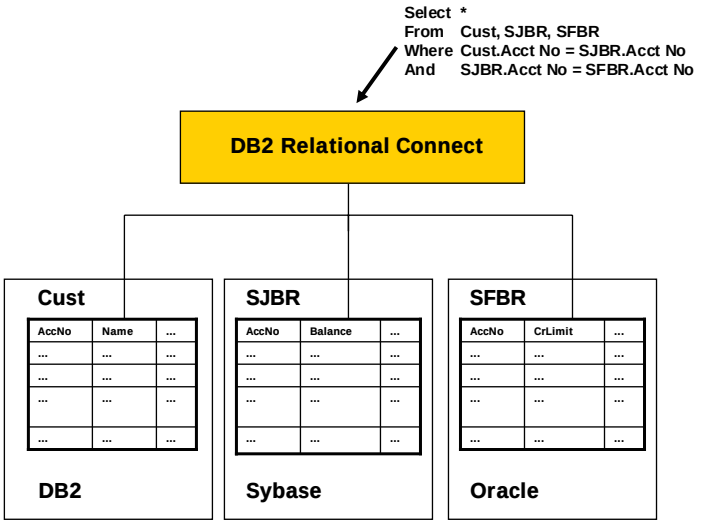


Data Federation/Integration

- Goal: homogeneous, integrated view of data from multiple sources
  - a single (logical) database
  - a single query may collect (or join) data from multiple sources
- Data Federation requires
  - Wrapper/mediator technology
  - Data and schema integration mechanisms

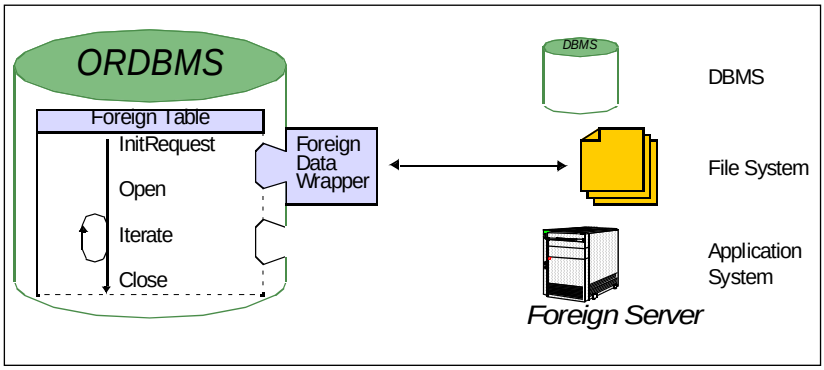


Example - DB2 Relational Connect



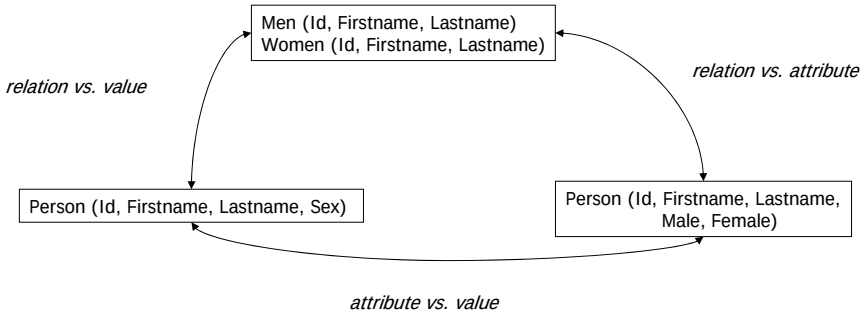
Standard – SQL/MED

- 'Foreign Data Wrapper' in 'SQL/MED'



Schematic Heterogeneity

- Potential conflicts in terms of
  - data types, integrity constraints, primary/foreign keys, etc.
  - schema (relation vs. attribute vs. value, ...)
  - structure (grouping as tables)



Semantic Heterogeneity

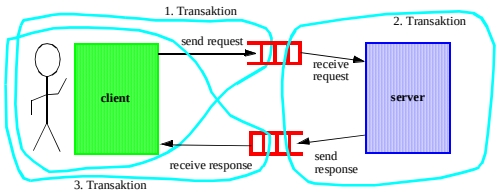
- Differences in the meaning, interpretation, or usage of a concept, feature, or value
- Naming conflicts
  - definition of a concept
  - synonyms and homonyms (surname vs. last name; title)
  - measures (cm vs. inch)
- Identity
  - object identification, recognizing duplicates, ...
- Data conflicts
  - duplicates have conflicting attribute values

Integration Process

- Schema Matching
  - Find inter-schema correspondences
- Schema Mapping
  - Based on correspondences
  - Define how to "translate" one schema into another
    - implies data transformation
- Schema Integration
  - Based on correspondences (and mapping)
  - Define an integrated, global/federated schema

Message-Oriented Middleware (MOM)

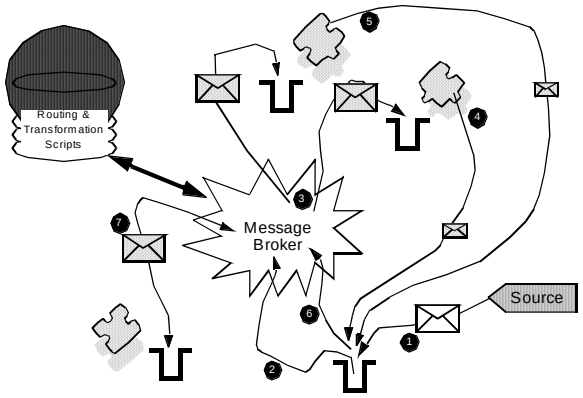
- Message-oriented interoperability
  - programming model: asynchronous message exchange
- Support for persistent, transactional message queues
  - asynchronous transactions
  - reliable messaging
- Optimizing throughput, not response time
- Loosely-coupled application components
  - “client” not blocked during request processing
  - “server” may chose request processing time more flexibly
    - may not even be available at request enqueue time



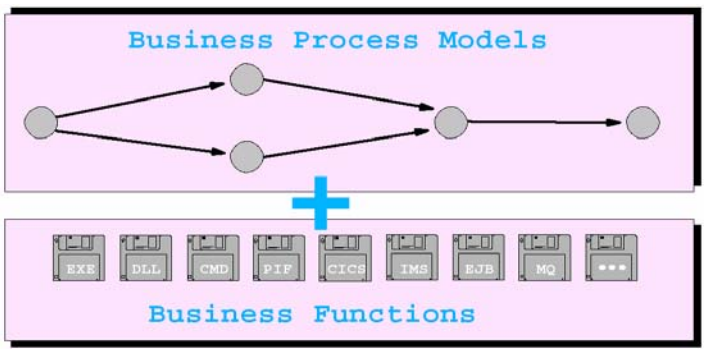
Enterprise Application Integration

- Focus on application integration within an enterprise (vs. development of new application)
  - integration across different middleware platforms
  - major shift towards asynchronous interactions
- Message Brokers
  - based on MOM
  - hub-and-spoke (instead of point-to-point)
  - publish and subscribe model to link applications together
- Business Process Modeling and Workflow Management Systems
  - make integration logic explicit, easy to modify/extend
  - "programming in the large"

Message Brokering – Processing Model



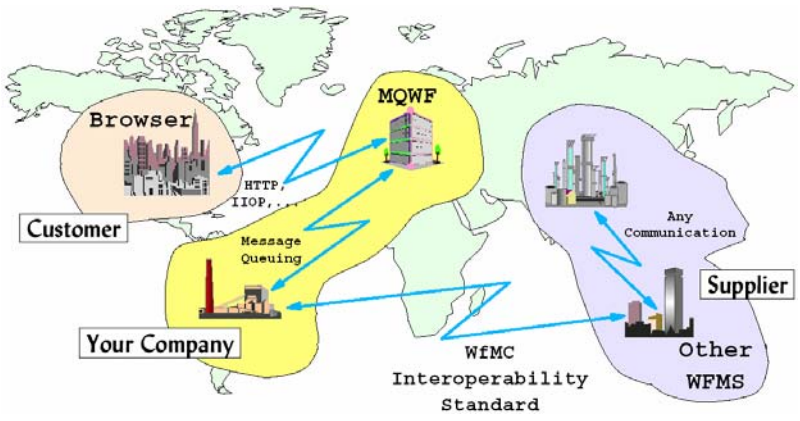
Workflow-Based Applications: Structure



Workflows And External Communications

- Customers invoke company's applications to perform certain steps of the business process
  - E.g. place on order, inquire status,...
  - Company's applications must get a browser-based front-end for that purpose ("web-up")
- Workflow activities may directly communicate with the outside
  - Send e-mail, faxes, messages,...
- Workflow activities may trigger actions in another company
  - Simple invokation of program or start of another workflow ("subprocess" from invokers point-of-view)
  - Such "business-to-business" scenarios are the base for realizing sophisticated "supply chains"

Virtual Enterprise: Scenario



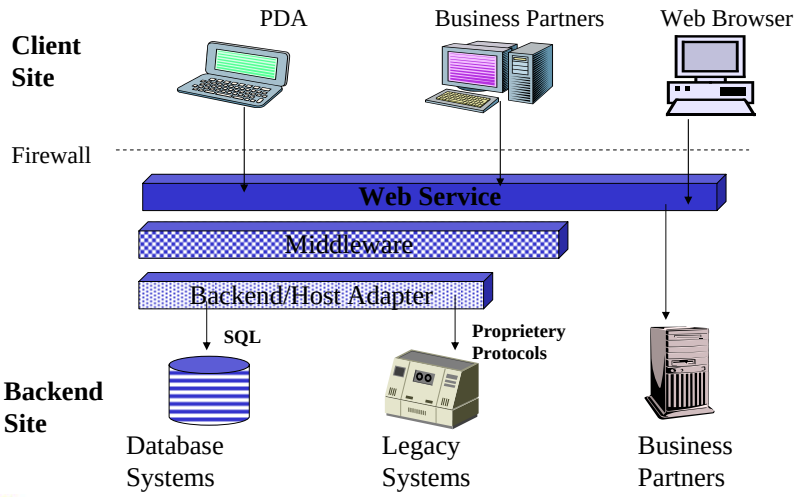
Business-To-Businesss (B2B) Integration

- Goal: facilitate interaction among trading partners, across companies
  - Establish relation between processes of different enterprises
  - Predominant are relation to suppliers, and customer relations to other enterprises like industrial consumers, retailers, banks
- Traditional B2B has focused on well-defined, standard message formats and protocols (e.g., RosettaNet, cXML)
  - Ad hoc B2B occurs today via XML over HTTP
- How to publish business functions to customers, partners and suppliers?
  - E.g. access to reservation systems, quote systems
  - Programmatic access to a service, independent of underlying implementation and client software
- Technologies such as Corba, DCOM, EJBs, etc. barely present in this context

Web Services

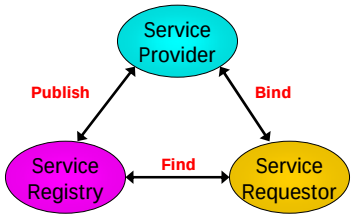
- New distributed computing platform built on existing infrastructure including XML & HTTP
  - Web services are for B2B what browsers are for B2C
- Self-contained, self describing, modular service that can be published, located and invoked across the web
  - Refer to open standards and specifications:
    - component model (WSDL)
    - inter-component model communication (SOAP)
    - discovery (UDDI)
  - Platform- and implementation-independent access
  - Described, searched, and executed based on XML
- Enable component-oriented applications
  - Loose coupling from client to service
  - Enable to integrate legacy systems into the web
  - Useful for other distributed computing frameworks such as Corba, DCOM, EJBs

Web Service System Architecture



Service-Oriented Architecture (SOA)

- Service Requestor
  - Finds required services via Service Broker
  - Binds to services via Service Provider
- Service Provider
  - Provides e-business services
  - Publishes availability of these services through a registry
- Service Registry
  - Provides support for publishing and locating services
  - Like telephone yellow pages

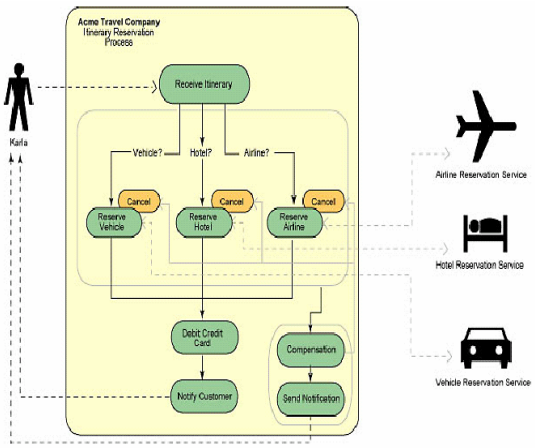


Standards

- UDDI
  - Universal Description, Discovery and Integration
  - Registry of and search for web services
- SOAP
  - Simple Object Access Protocol
  - Communication protocol
- WSDL
  - Web Services Description Language
  - Description of a service's functionality
- XML
  - eXtensible Markup Language
  - Underlying basic representation approach

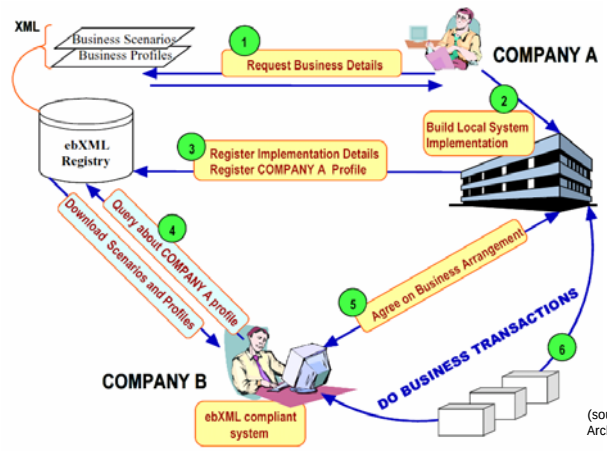
Web Services & Business Processes

- Business process making use of web services
- Business process externalized as a web service
- Long-running transactions
- Compensation
- Correlation
- Dynamic Binding of business partners and web services



e-Business Collaboration

- Example: ebXML



(source: ebXML Technical Architecture Specification)