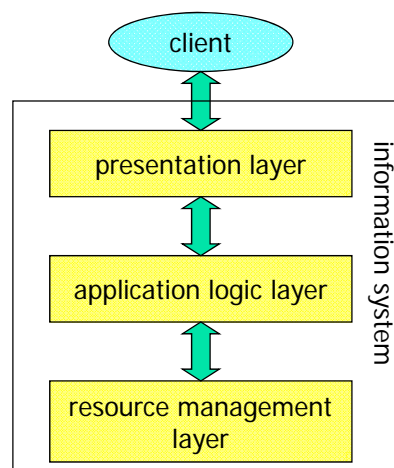


## Chapter 1 – Motivation



### 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
- Challenges
  - distribution
  - autonomy
  - heterogeneity
  - performance & scalability
  - high availability
  - complexity
  - ...



## 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
  - application logic independent from infrastructure code
  - powerful programming abstractions



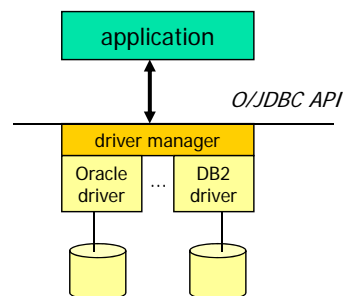
© Prof. Dr.-Ing. Stefan DeBloch

3

Middleware for Information Systems

## 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



© Prof. Dr.-Ing. Stefan DeBloch

4

Middleware for Information Systems

## Role of the WWW for IS

- Initial purpose: sharing information on the internet
  - technologies
    - HTML documents
    - HTTP protocol
  - web browser as client for internet information access
- For Information Systems: connecting remote clients with applications across the internet/intranet
  - "web-enabled" applications
    - extend application reach to the consumer
    - leverage advantages of web technologies
  - web browser as a universal application client
    - "thin client"
    - no application-specific client code has to be installed
  - requirements
    - content is coming from dynamic sources (IS, DBS)
    - request to access a resource has to result in application invocation
    - session state: tracking repeated interactions of the same client with a web server

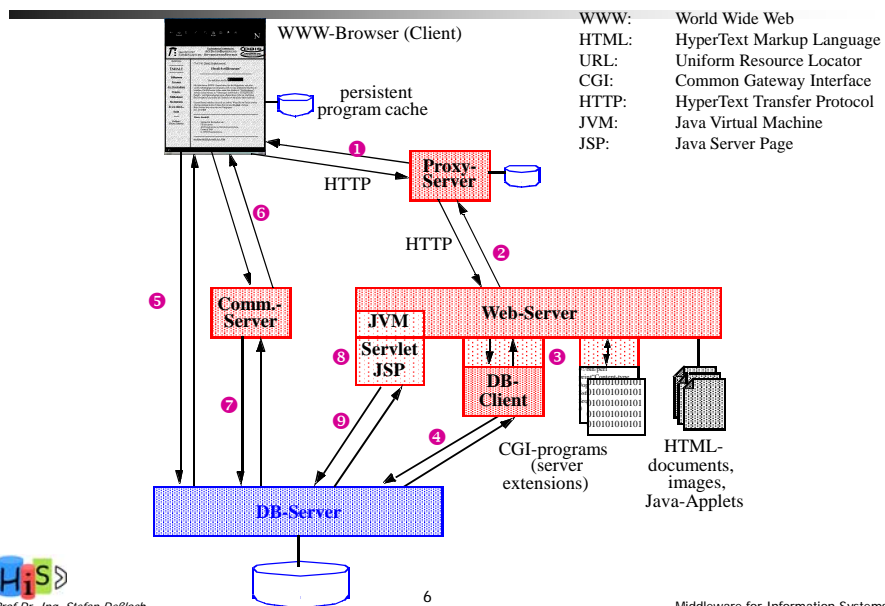


© Prof. Dr.-Ing. Stefan Deßloch

5

Middleware for Information Systems

## Web-based IS - Overview



© Prof. Dr.-Ing. Stefan Deßloch

6

Middleware for Information Systems

## Communication and Distributed Processing

- Distributed (Information) System
  - consists of (possibly autonomous) subsystems
  - jointly working in a coordinated manner
- How do subsystems communicate?
  - **Remote Procedure Calls (RPC)**
    - transparently invoke procedures located on other machines
  - Peer-To-Peer-Messaging
  - Message Queuing
- Transactional Support (ACID properties) for distributed processing
  - Server/system components are Resource Managers
  - (Transactional) Remote Procedure Calls (TRPC)
  - Distributed Transaction Processing

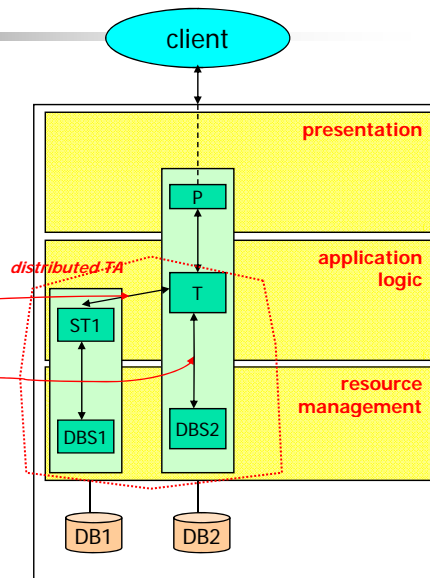


## RPCs and Transactions

- Example scenario for T: debit/credit
  - T invokes debit procedure (ST1), modifying DB1
  - T performs credit operation on DBS2, modifying DB2
- Need transactional guarantees for T
- Program structure of T
 

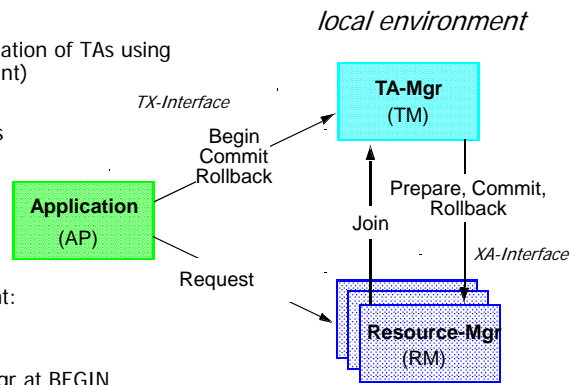
```

BOT
CALL debit(...)
CONNECT (DB2)
UPDATE ACCOUNTS SET ...
DISCONNECT
EOT
            
```
- Requires coordination of distributed transaction
  - based on 2PC



## X/OPEN – Standard for Distributed TA Processing

- Resource Manager
  - recoverable
  - supports external coordination of TAs using 2PC protocol (XA-compliant)
- TA-Mgr
  - coordinates, controls RMs
- Application Program
  - demarcates TA (TA-brackets)
  - invokes RM services
    - e.g., SQL-statements
  - in distributed environment: performs (T)RPCs
- Transactional Context
  - TRID generated by TA-Mgr at BEGIN
  - established at the client
  - passed along (transitively) with RM-requests, RPCs



## Application Middleware – Main Tasks

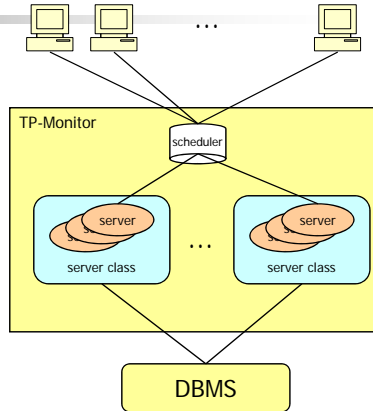
- Distributed computing infrastructure (RPC, RMI)
- Transactional capabilities
  - programming abstractions (demarcation)
  - distributed transaction management
- Security services
  - authentication, authorization, secure transmission, ...
- Unified access to heterogeneous information sources and application systems
- Scalable and efficient application processing
  - large number of client applications or end users
- Reliability, high availability

*Programming model abstractions that allow the developer to focus on application logic (i.e., ignore infrastructure as much as possible)*



## Types of Middleware

- RPC/RMI middleware infrastructure
  - basic development and execution support
  - additional services
- TP monitor
  - transaction management, TRPC
  - process management
  - broad set of capabilities
- Object broker (e.g., CORBA)
  - distributed object computing, RMI
  - additional services
- Object transaction monitor
  - ... = TP monitor + object broker
  - most often: TP monitor extended with object-oriented (object broker) interfaces
- Component Transaction Monitor
  - ... = TP monitor + distributed objects + server-side component model

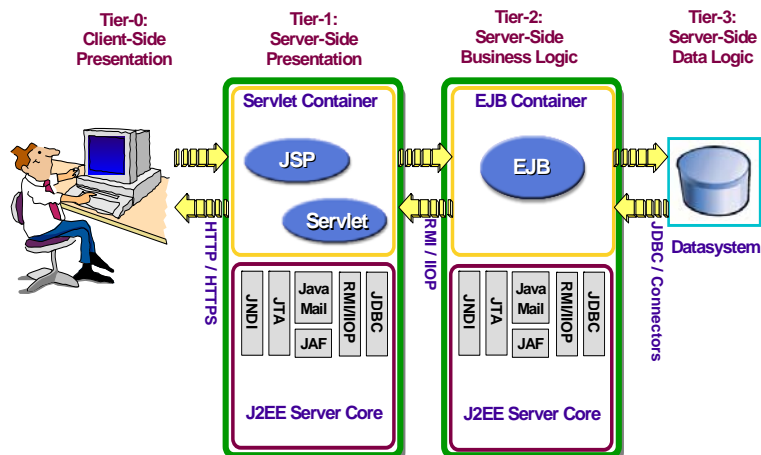


© Prof. Dr.-Ing. Stefan DeBloch

11

Middleware for Information Systems

## 4-Tier Distributed Computing in J2EE



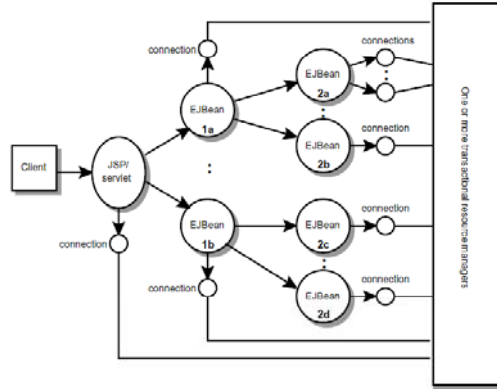
© Prof. Dr.-Ing. Stefan DeBloch

12

Middleware for Information Systems

## Important Services for Distributed IS

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

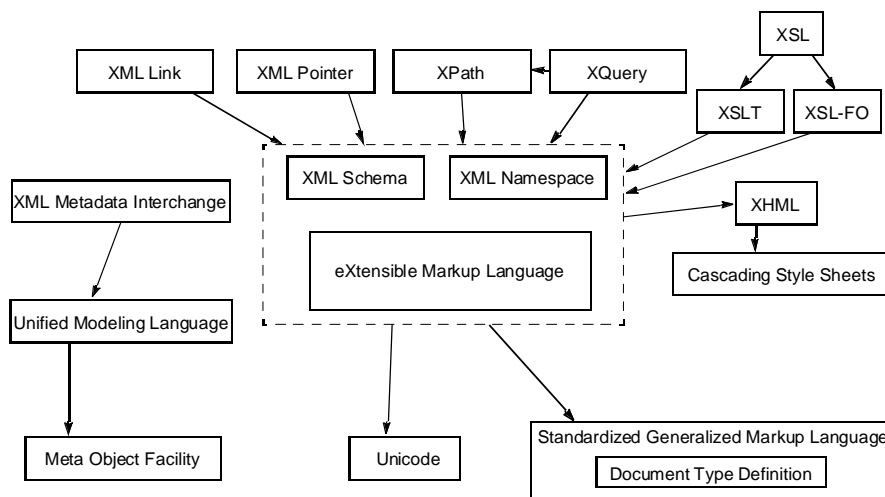


© Prof. Dr.-Ing. Stefan Deßloch

13

Middleware for Information Systems

## XML Data Access, Processing, Transformation



© Prof. Dr.-Ing. Stefan Deßloch

14

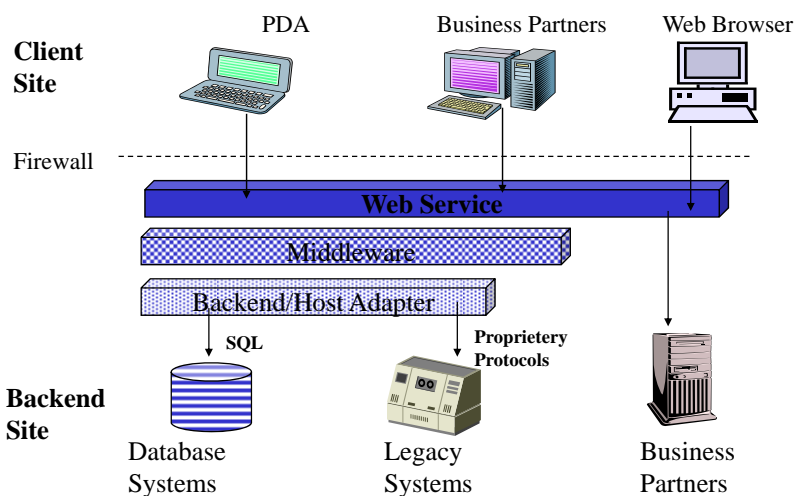
Middleware for Information Systems

## 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



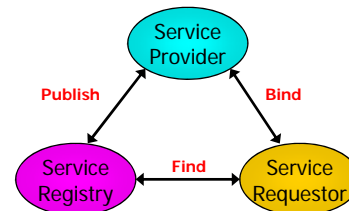
## Web Service System Architecture





## 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



© Prof. Dr.-Ing. Stefan Deßloch

17

Middleware for Information Systems

## 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



© Prof. Dr.-Ing. Stefan Deßloch

18

Middleware for Information Systems

## Outlook on EIS

---

- Data/Information Integration
  - integrated access to (heterogeneous) data originating from multiple sources
    - queries range over data from multiple DBs!
  - virtual integration: integrate on access/query (e.g., federated DBMS)
  - materialized integration: extract, transform, load data into a single materialized data warehouse in advance (e.g., data replication, data warehousing)
  - needs a strong foundation to overcome multiple kinds of heterogeneity
- Enterprise Application Integration
  - integration of (heterogeneous, coarse-grained) applications within an enterprise (vs. development of new application)
  - integration across different middleware platforms
- Business-to-business Integration
  - support interactions, integration of business processes among trading partners, across company boundaries
  - foundation for e-business, e-commerce

