mobile business

Lecture 03

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Business Informatics 2 (PWIN)
SS 2017

Information Systems II
Models and Architectures

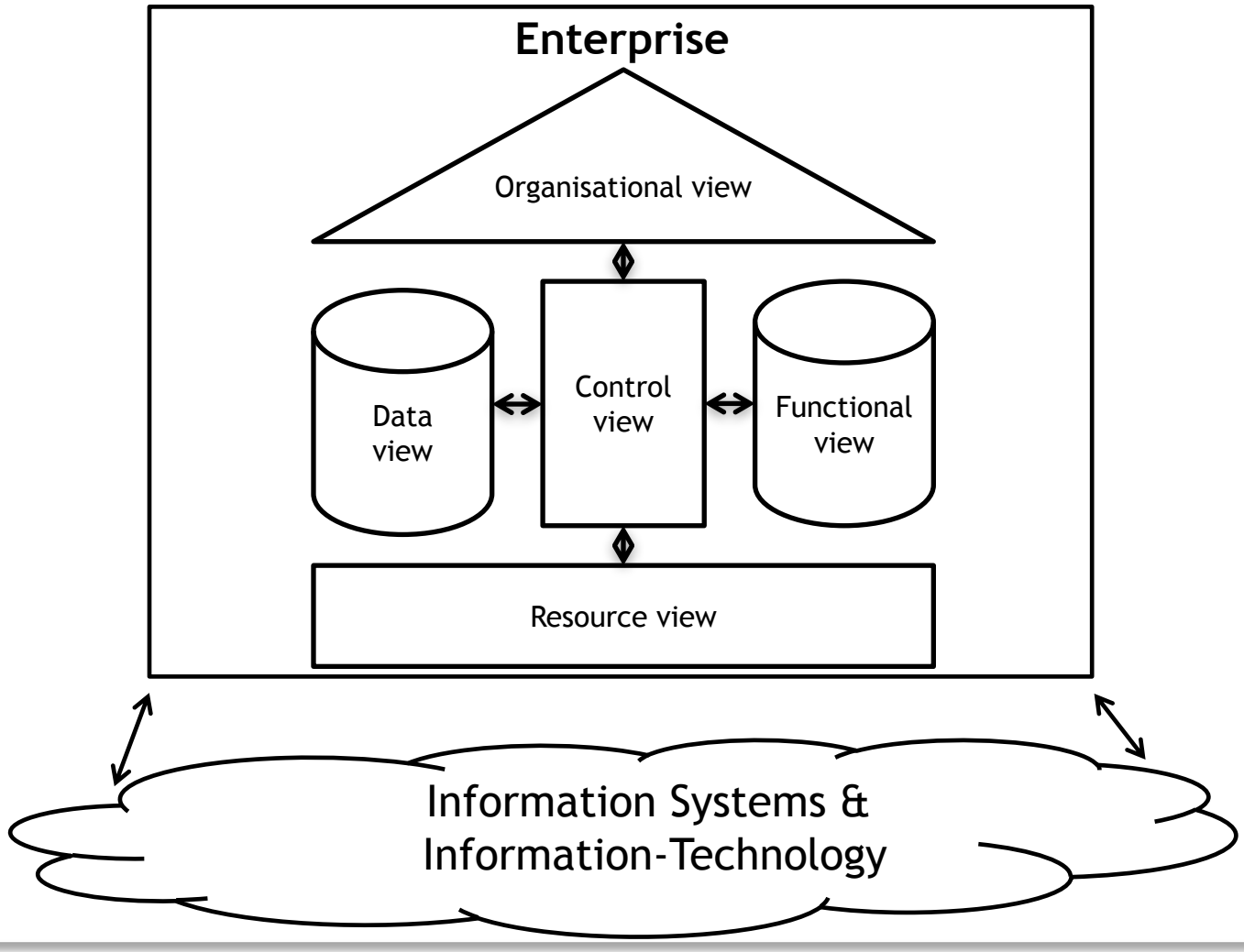
Prof. Dr. Kai Rannenberg

Deutsche Telekom Chair of Mobile Business & Multilateral Security
Johann Wolfgang Goethe University Frankfurt a. M.

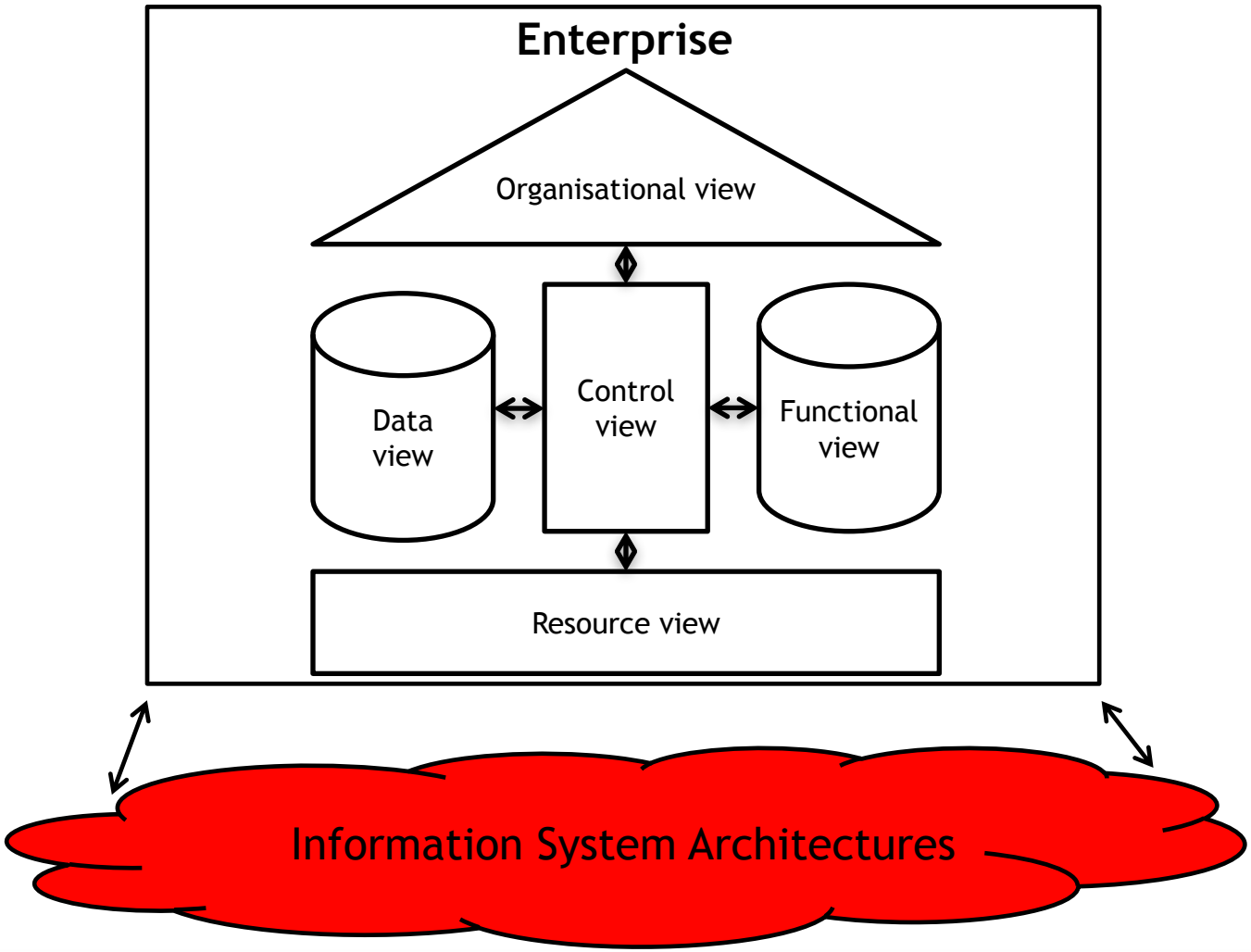
Agenda

- Enterprise Models vs. IS Architecture Models
- Structural Models for IS Architectures
- IS Architecture Concepts

Enterprise Models vs. IS Architecture Models



Enterprise Models vs. IS Architecture Models



mobile business

Agenda

- Enterprise Models vs. IS Architecture Models
- Structural Models for IS Architectures
- IS Architecture Concepts

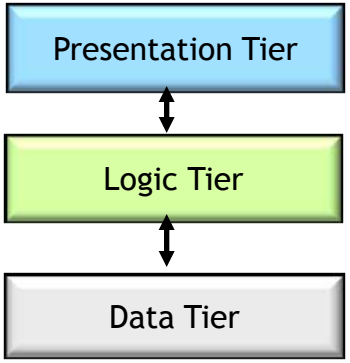
Requirements for the Structure of IS Architectures

- Minimisation of Complexity for IS Components
- Scalability of IS Components
- Portability of IS Components
- Maintainability of IS Components
- Standardisation of IS Components
- Well-defined interfaces between IS Components
- Independence of IS Components

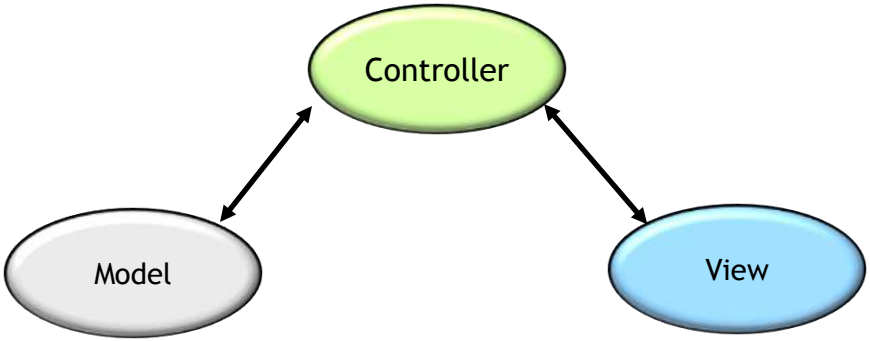
Modularisation of IS Components

Two Common Structural Models for IS Architectures

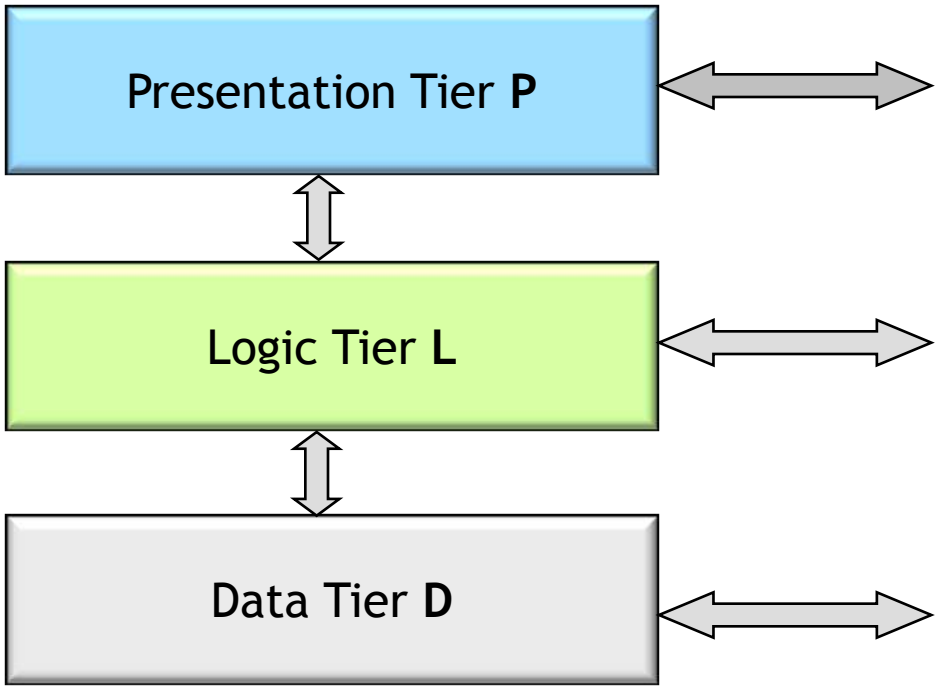
- Three-Tier Concept



- Model-View-Controller (MVC) Concept



Three-Tier Concept



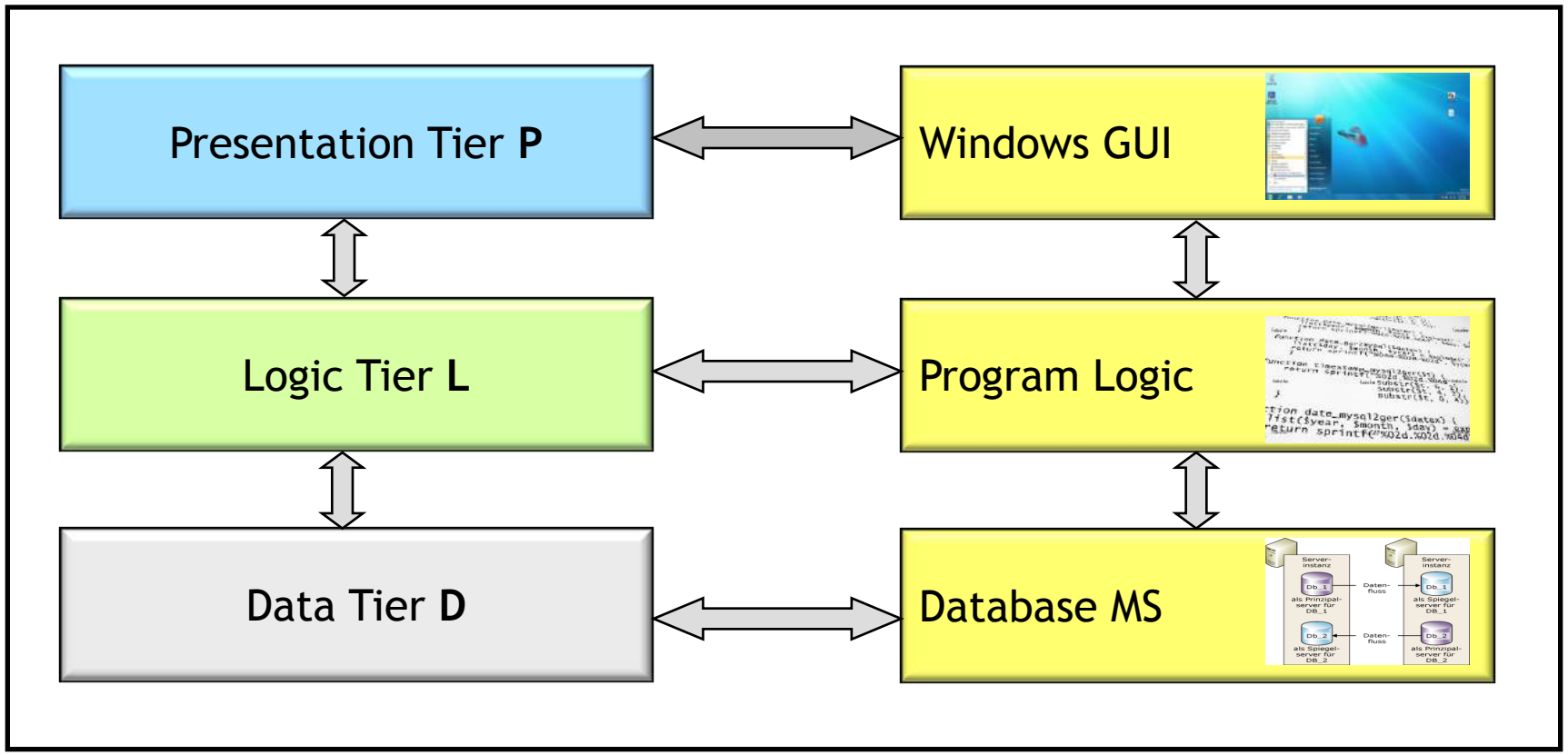
Receives user input / display of data to the user (GUI)

Contains all the program logic

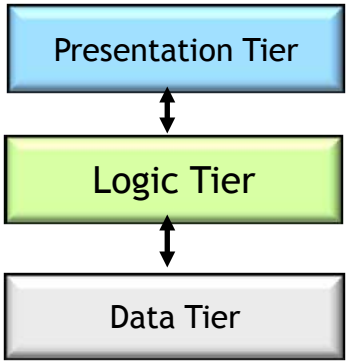
Stores and manage the data

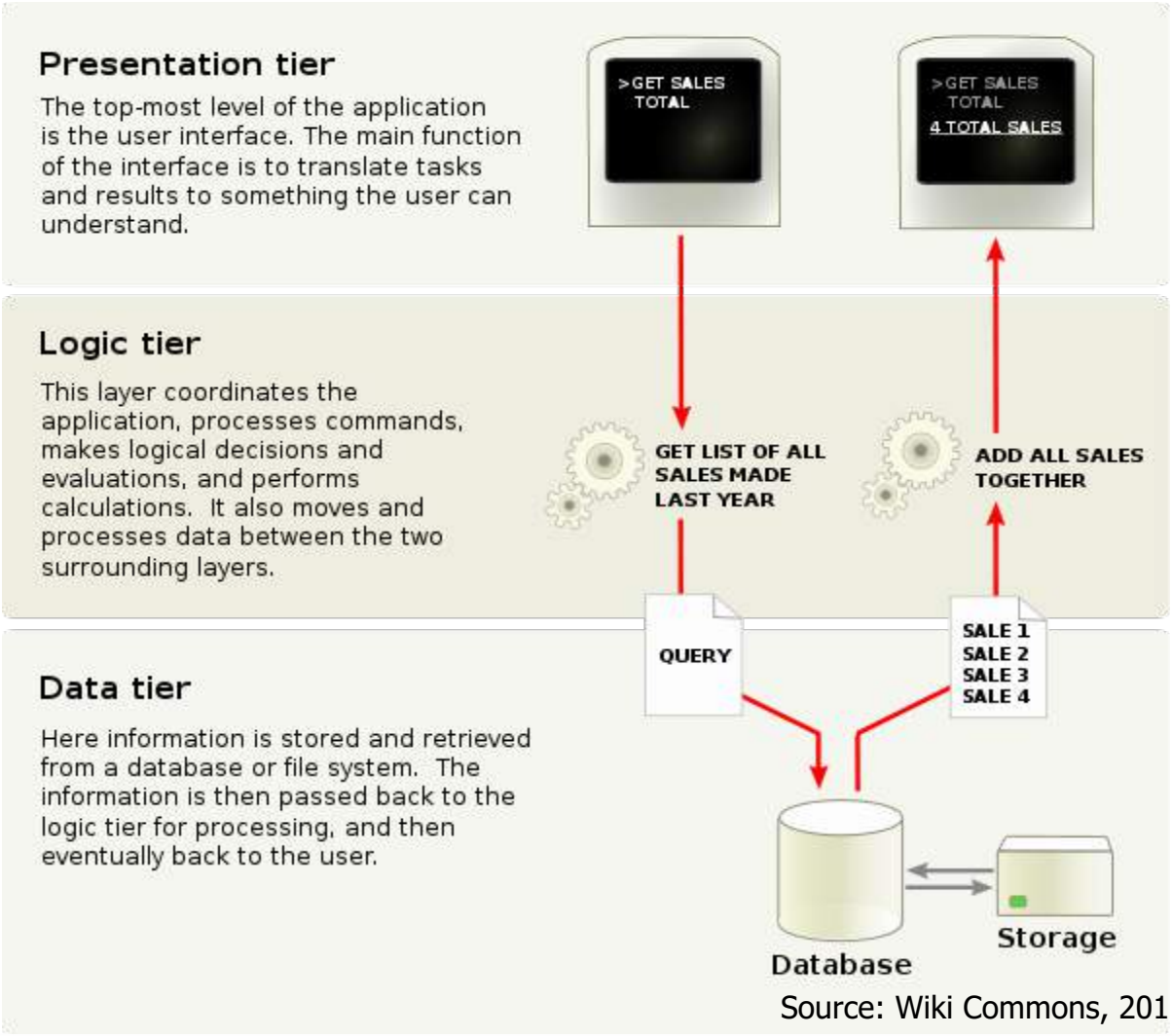
Three-Tier Concept Example (1)

Conventional IS



Three-Tier Concept Example (2)

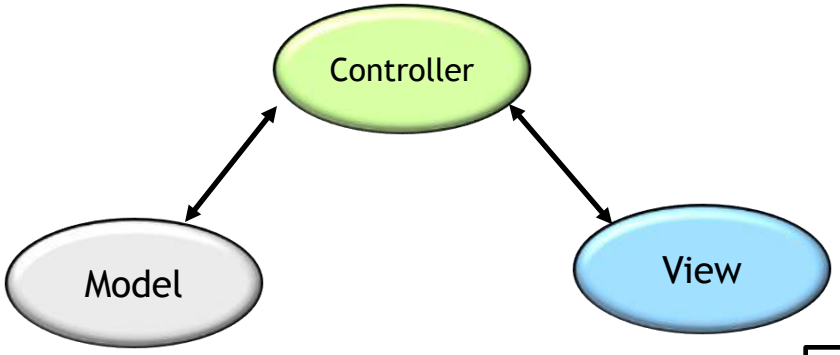




Source: Wiki Commons, 2011

Model-View-Controller Concept

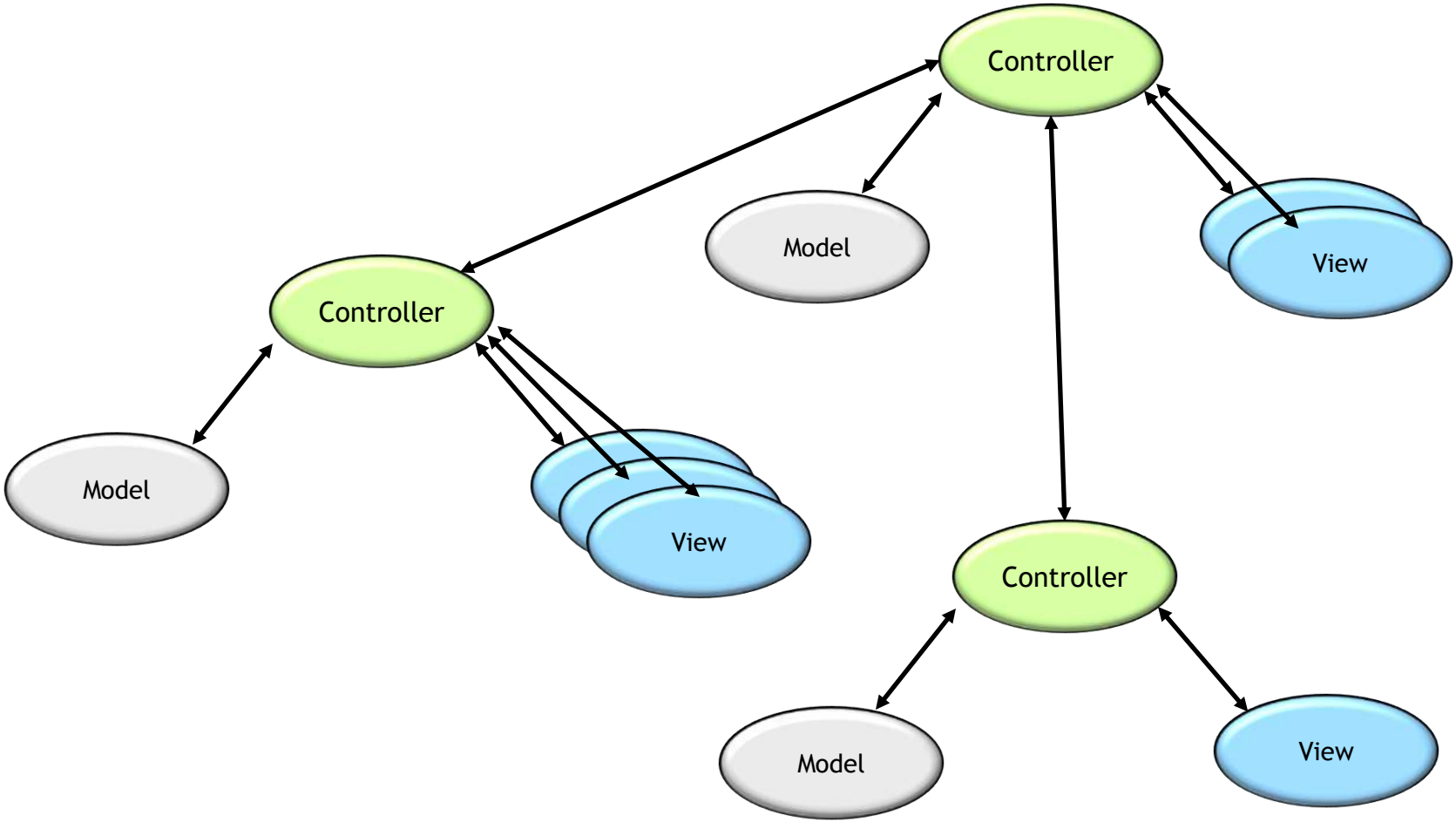
Controller controls *view(s)* and initiates the relevant data updates



Manages data and, if applicable, contains the program logic

Receives user input / displays data from *model* to the user (GUI)

More Complex Model-View-Controller Concept



Summary on Three-Tier and MVC Concept

- Similar concepts for structuring IS architectures
- Neither one of the concepts is universally defined or specified, e.g.
  - Two-tier concepts are also in existence (Tier Architecture)
  - Program logic resides sometimes in the *model* and other times in the *controller* (MVC Architecture)
- **In conclusion:**

  Independent of the underlying structural models for IS architectures, make sure to modularise certain categories of functionality in an IS.

Agenda

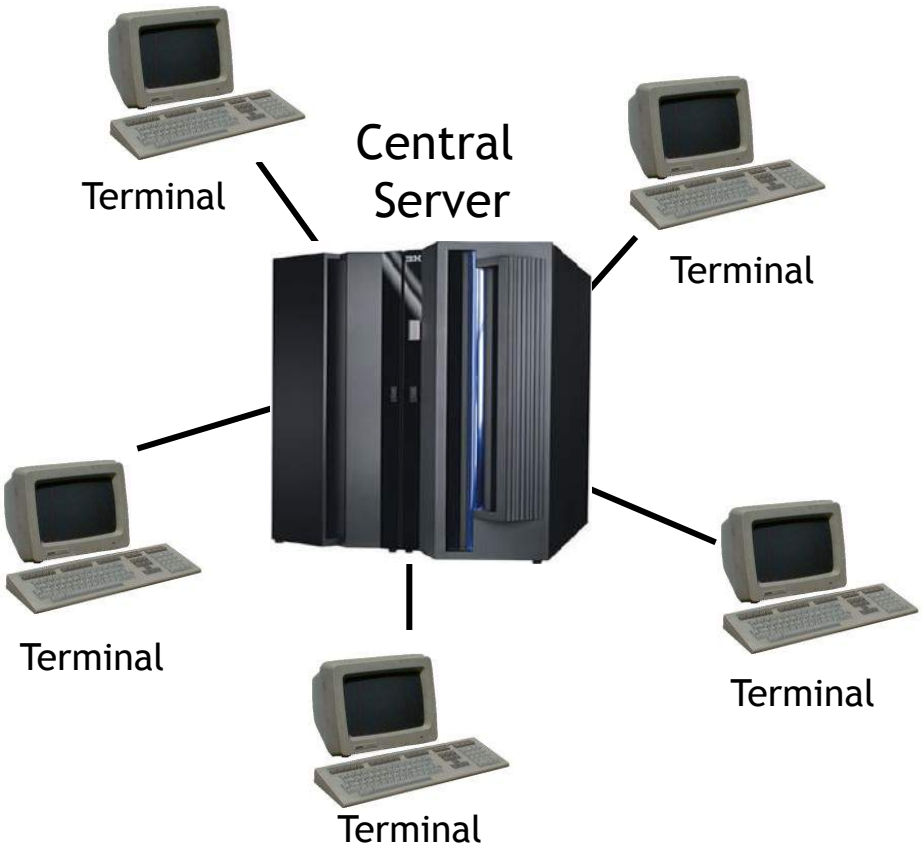
- Enterprise Models vs. IS Architecture Models
- Structural Models for IS Architectures
- IS Architecture Concepts

Architecture Concepts of Networked IS

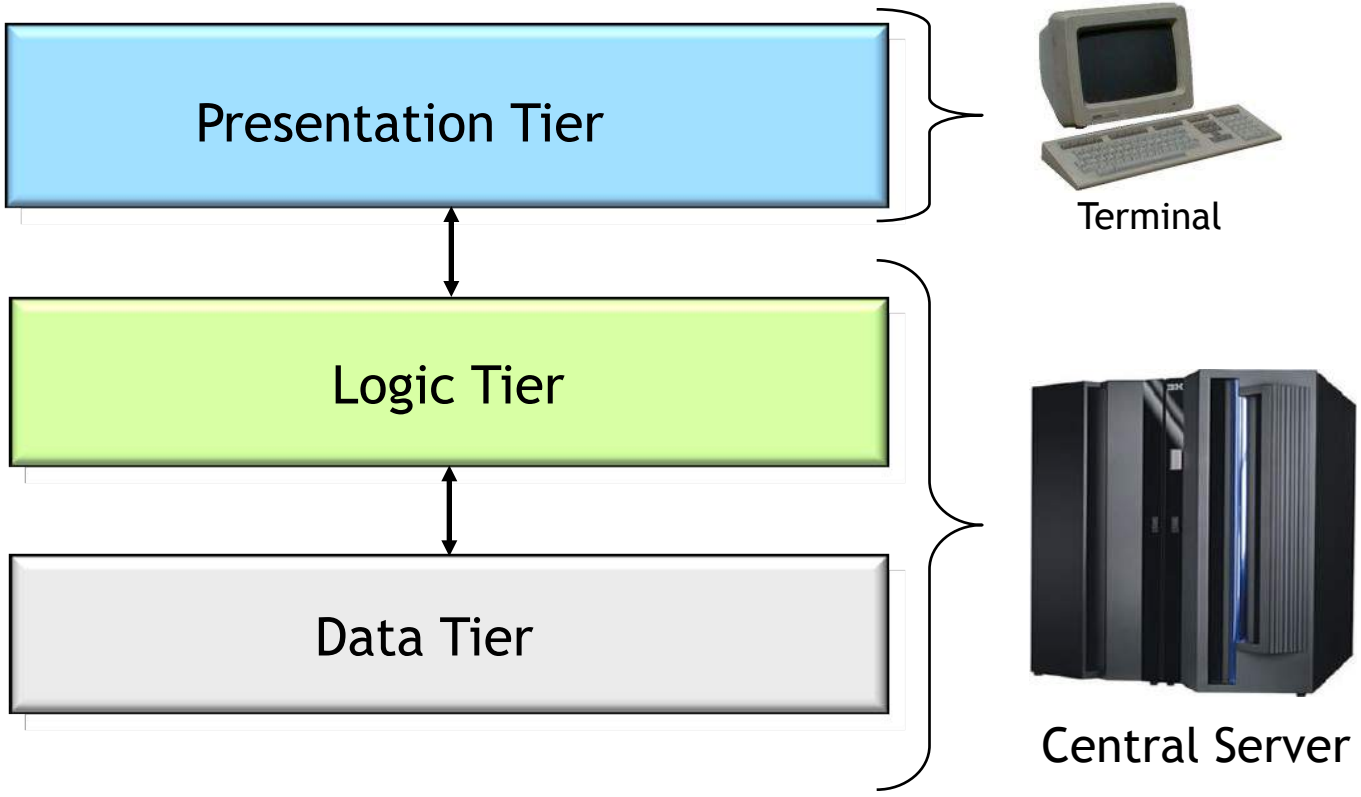
- **Central Server Architecture**
  Low-feature terminals (receiver of services) attached to a powerful central computing unit (provider of services)
- **Client / Server Architecture**
  Network of computers, which can take the role of a server (provider of services), a client (receiver of services) or both.
- **Cloud Computing Architecture**
  Network of computers in the role of a client (receiver of services) connected to a “cloud” of computers (provider of services), which act as a single central server
- **Peer-to-Peer Architecture**
  Network of computers holding equal rights (provider / receiver of services)

Central Server Architecture

- One powerful Central Computer
- „Dumb“ low-feature terminals (often even without hard drive)
- Terminals provide only the graphical user interface (GUI)
- Central Server in charge of processing applications
- Central Server takes care of database and its management



Central Server Concept along the Structural Three-Tier Architecture



Review of the Central Server Architecture Concept

- Benefits
  - Central, common data storage
  - Homogenous application environment
  - No terminal administration required
  - Low-cost terminals
- Issues
  - Single Point of Failure
  - Fixed Network Structure
  - Monolithic
  - Cost-intensive Central Servers
  - Problematic in case of huge traffic and amounts of data

Industry Central Server Solutions

Hardware

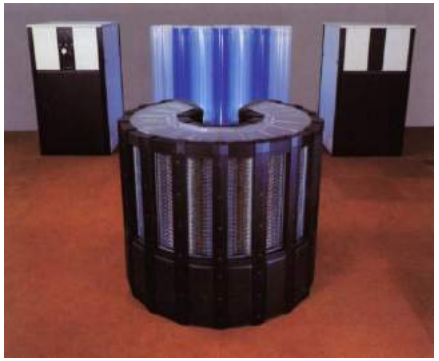
FUJITSU COMPUTERS SIEMENS

Sun microsystems

take it to the nth

hp invent

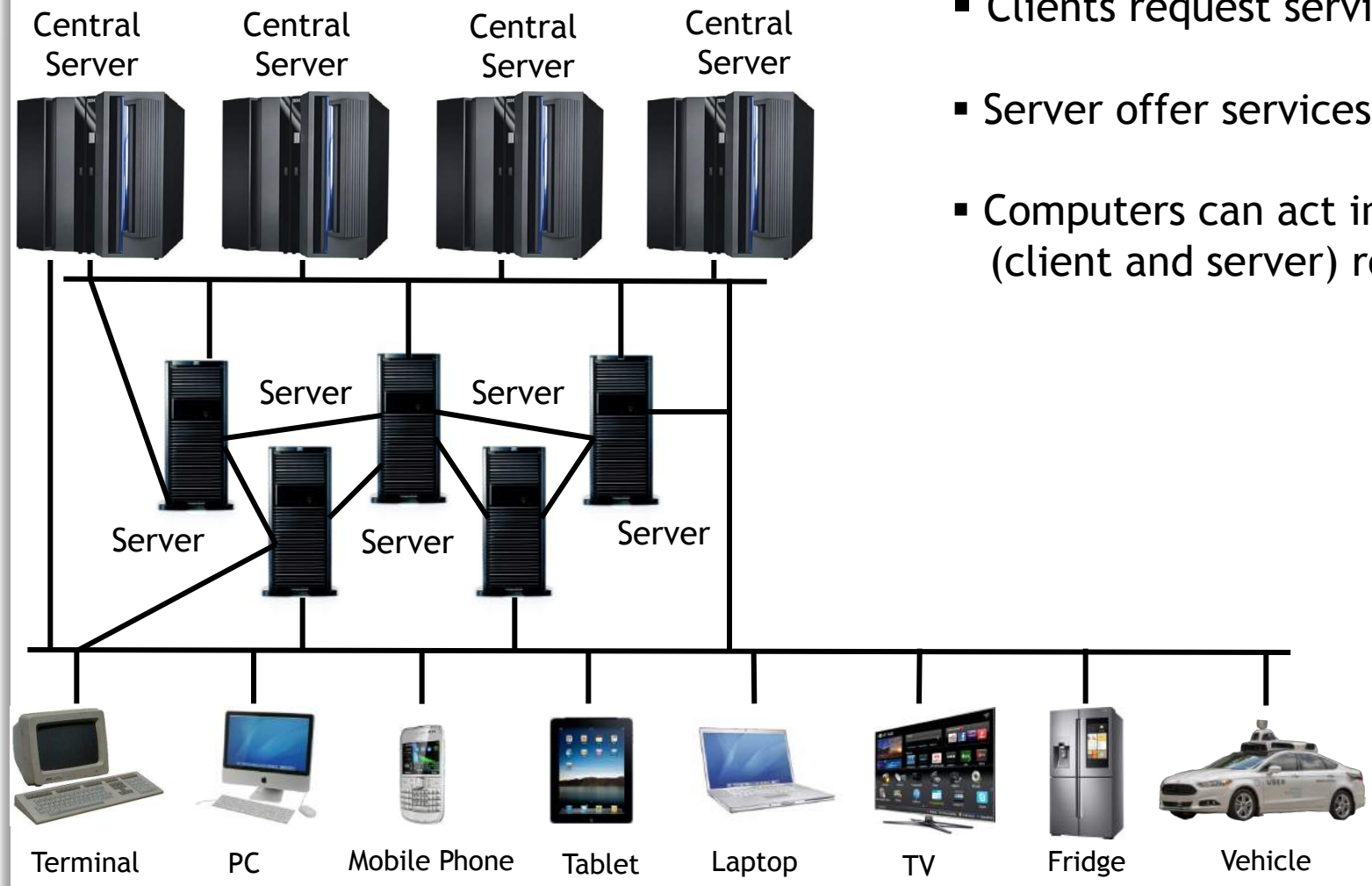
IBM



Operating Systems

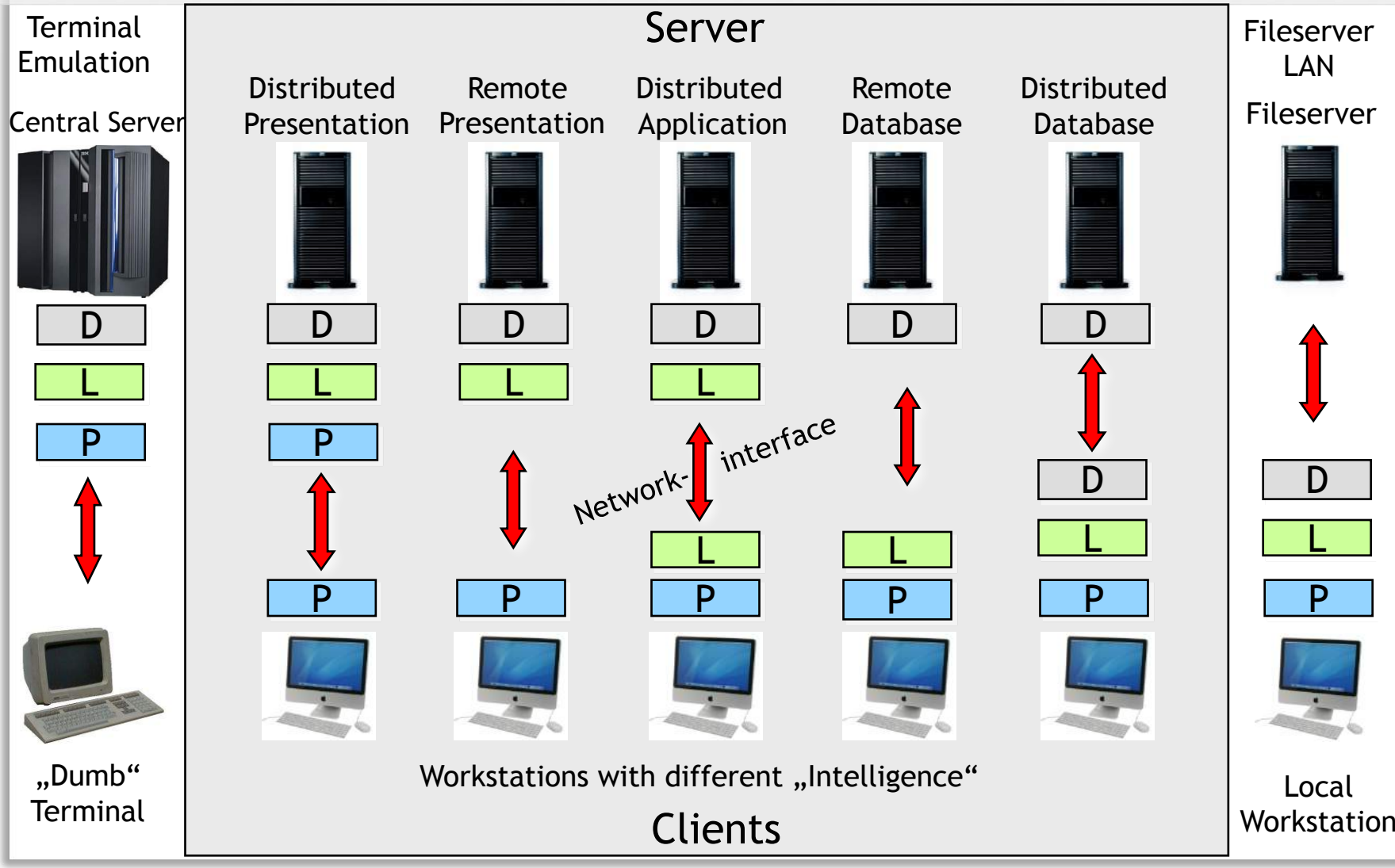
- Unix
- BS 2000
- OS/390
- MVS
- z/OS
- …

Client/Server Architecture



- Clients request services.
- Server offer services.
- Computers can act in both (client and server) roles.

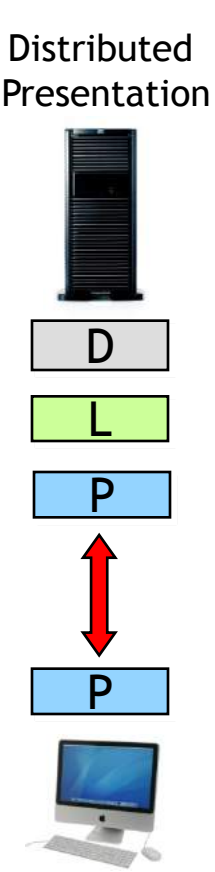
Client/Server Architecture along the Three-Tier Structural Concept



Distributed Presentation

Division of the presentation between server and client:

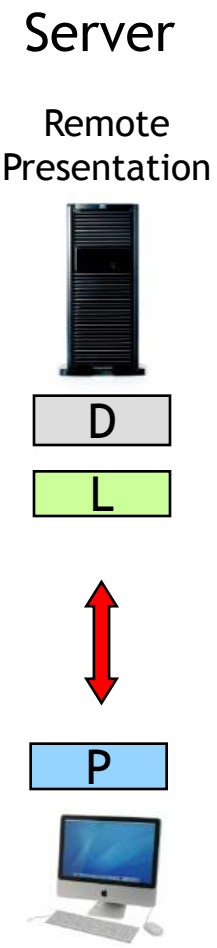
- Abstract part of the presentation (server)
  Objects (e.g. a window) are created in an abstract manner, i.e. without any concrete representation and functionality.
- Platform-specific part of the presentation (client)
  Abstract objects are created and represented in a platform-specific manner (e.g .making use of the platform's GUI).
- Benefits of this approach
  Heterogeneous application systems can be integrated into a unified user interface or used on different platforms.
- Application example:
  - X-Windows: A user interface using X-Windows can be represented on multiple platforms.
  - Mobile Web App within Native App: Spiegel Online



Remote Presentation

Presentation is outsourced to the client:

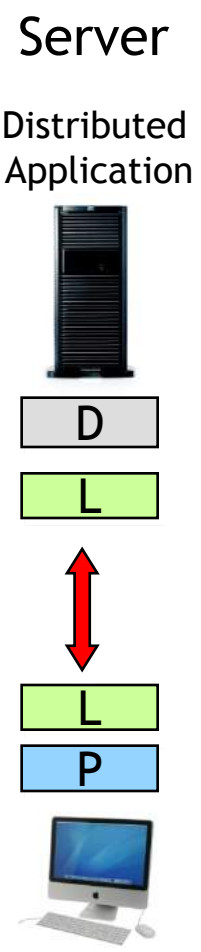
- Outsourcing of the presentation to the client is especially beneficial, if the central server has no own user interface.
- Clients are able to run on several different platforms.
- User interfaces can be individually customised according to users' needs (e.g. GUI).
- Client can not be a „dumb" terminal.
- Examples: Citrix XenDesktop, TeamViewer, Apple Airplay



Distributed Application

Division of the application functions (logic) between server and client:

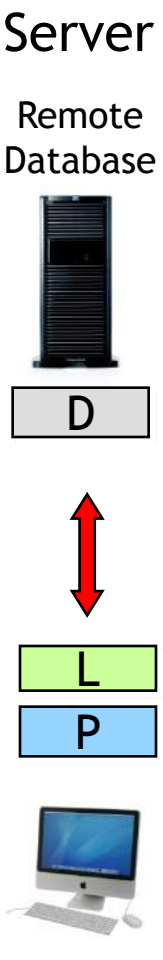
- Centrally used application functions are hosted on the server in order to be available for everyone.
- Decentralized applications reside on the respective client.
- Central application functions will only be used on demand.
- Advantages: Development and maintenance of application functions get simplified; complexity is reduced.
- Example: Groupware, Facebook App, DB Navigator App, Siri



Remote database

Data management resides on the server:

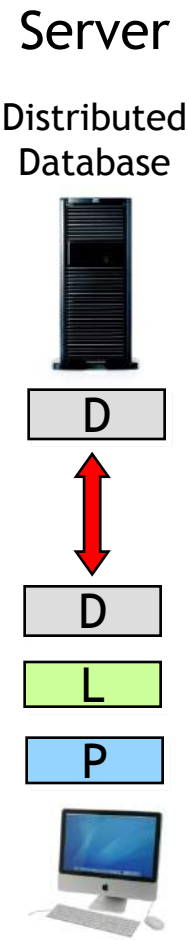
- Traditional approach for database applications
- Multiple application systems use the same database.
- Data management can also be distributed across multiple servers.
- Problem: There are several implementations of the popular database query language „SQL“ with many proprietary extensions and differences.
- Classic example: Customer Information System, Dropbox App, DB Navigator App (previously)



Distributed Database

Data management is distributed between server and client:

- Two incarnations of a distributed database exist:
  - Partitioning of data storage between server and client
    - Organisational structure: Centralized directory of an enterprise vs. personal address book
    - Frequency of use: Current business figures vs. archive
    - Access time: Current stock market values vs. archive
    - …
  - Partitioning of database management system (DBMS) between server and client
    - Data access functionality (frequently used) on the client
    - Database administration (less frequently used) on the server
    - Examples: Here Maps App, Navigon App



mobile business

Review of the Client/Server Architecture

- Advantages
  - Can be designed and extended flexibly
  - High interaction and communication capabilities
  - Dependability through redundant resources
- Disadvantages
  - High server workload because of multi-user access
  - High planning and coordination efforts
  - High network bandwidth required
  - High administrative workload

Cloud Computing Architecture

Internet-centric Computing Architecture:

- Providers are offering complex services based on hard- and software in an abstract form.
- Storage, computing power, or complex services can be accessed by client via defined interfaces via the Internet.
- Underlying hard- or software of a cloud is not relevant for a client.
- Types of Cloud Computing Services
  - Infrastructure as a Service
  - Platform as a Service
  - Software as a Service
- Providers, e.g.
  - Amazon, Google, Microsoft, Deutsche Telekom, etc.

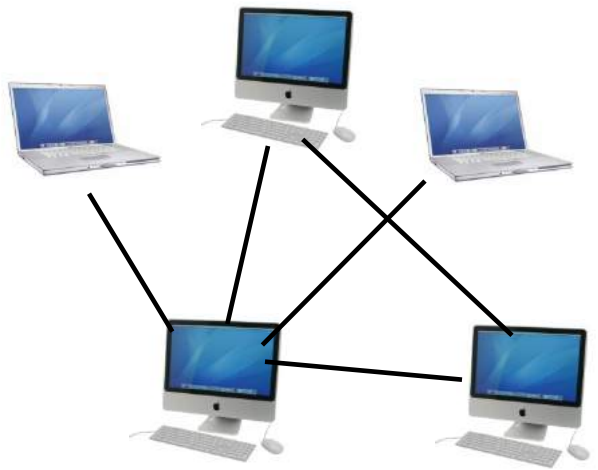
Cloud Computing Architecture

- Advantages
  - Information system become highly scalable.
  - Central data storage and backup
  - Cost efficient (one has only to pay for the actually used computing power and time)
  - Anytime, anywhere access to applications and data
  - Allows to run sophisticated applications on low-powered systems (e.g. mobile devices' voice recognition systems)
- Disadvantages
  - Enterprises or end users have to rely on the cloud service provider and the legal and political environment.
  - Potential threats
    - Data leakage
    - Data unavailability
    - Provider bankruptcy, lock-in effects
    - Internet connection failures

Peer-to-Peer Architecture

Network of computers with equal capabilities

- Properties
  - No central instance coordinating the required interactions
  - No centralized database
  - Peers act autonomic.
  - Each peer is only aware of those other peers it is currently communicating with.
  - Peers, connections, and information flows within this concept are not guaranteed.
- Advantage
  - Required resources are provided by many parties (e.g. for the distribution of large files)
- Disadvantages
  - High complexity
  - Requires critical mass of peers



Literature

- Hennekeuser J.; Peter G. (2004) "Rechner-Kommunikation für Anwender", Springer Verlag, Berlin.
- Schwickert, A. (2003) "Grundzüge der Wirtschaftsinformatik", Universität Gießen.
- WikiCommons (2011), http://en.wikipedia.org/wiki/Wikimedia_Commons, last visited 03-07-2013