System architecture design
Basic concepts

- **System** – a set of elements organized to accomplish a purpose and described by a set of models, possibly from different viewpoints.
- **Subsystem** – a part of the system grouping of elements of which some constitute a specification of the behavior offered by other contained elements.
System architecture modeling

Use Case View

Design View

Implementation View

Process View

Deployment View
System architecture types

• Integrated architecture
• Distributed architecture:
  – server based
  – client based
  – client-server
  – peer-to-peer
Multilayer architecture

Presentation logic layer

Application logic layer

Data access logic layer

Data storage layer

Client

Server
Server types

- Mainframe
- Minicomputer
- Microcomputer
Client types

• Personal computer (desktop, laptop, tablet)
• Terminal
• Special terminal
  – ATM machine
  – info-kiosk
  – smartphone
  – smartwatch
Server based architecture

- Mostly usage at 1960-1980
- Simple design
- Server overload risk
- Very costly

Diagram:

Client (terminal) → Server (mainframe)

- Presentation logic
- Application logic
- Data access logic
- Data storage
Client based architecture

- Since 1980
- Simple applications

Client

(personal computer)

Server

(microcomputer)

Presentation logic
Application logic
Data access logic
Data storage
Client-server architecture

- Since 1990
- Application logic may be divided between client and server
- Scalable
- Various server and client types
- Simple applications
- 1000 x cheaper than server-based architecture at the same performance level
Peer-to-peer architecture

- Whole logic and data storage at the clients
- Server only joins clients together

Diagram:
- Server
- Client connection set
- Two Clients
- Presentation logic
- Application logic
- Data access logic
- Data storage
Multitier architecture

- **Two-tier architecture** – application logic, data access logic and data storage at one server
- **Three-tier architecture** – application logic at one server, data access logic and data storage in the separate server
- **Multitier architecture** – many application servers and data servers
Two-tier architecture

Thick-client architecture:
• Application login at the client
• Special application needed
• Server spoofing risk

Thin-client architecture:
• Whole logic at the server
• Web browser at the client
• Simple applications (simple user interface)
• Server overload risk
Three-tier architecture

- Application server overload risk
Multi-tier architecture

- Scalable
- Highly secure
Distributed Object Computing

• Software objects (entities) are distributed between client and server
• Entity identification must be preserved at both sides
• Middleware – an intermediate (hidden) layer between client and server that transports client requests and server responses with serialized objects
Components vs. Classes

• Classes represent logical units; components represent physical units of the system
• Components represent the physical packaging of otherwise logical components and are at a different level of abstraction
• Components may live on nodes, classes may not
• Classes may have attributes and operations directly. Components have operations that are reachable only through their interfaces
Component diagram

List view

Observable collection

components

Data collection

interface
Bibliography

• Roger S. Pressman: Software Engineering. A Practitioner's Approach (book, PDF)
• Dennis A., Wixom B.H., Tegarden D., Systems Analysis & Design. An Object-Oriented Approach with UML, John Wiley and Sons, USA, 2002