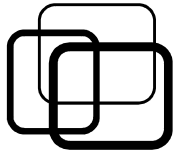
Software Architecture

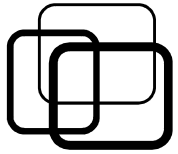
Inst. Nguyễn Minh Huy

Contents



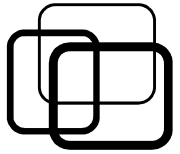
- Architecture Concepts.
- Architecture Models.
- Distributed Technologies.

Contents

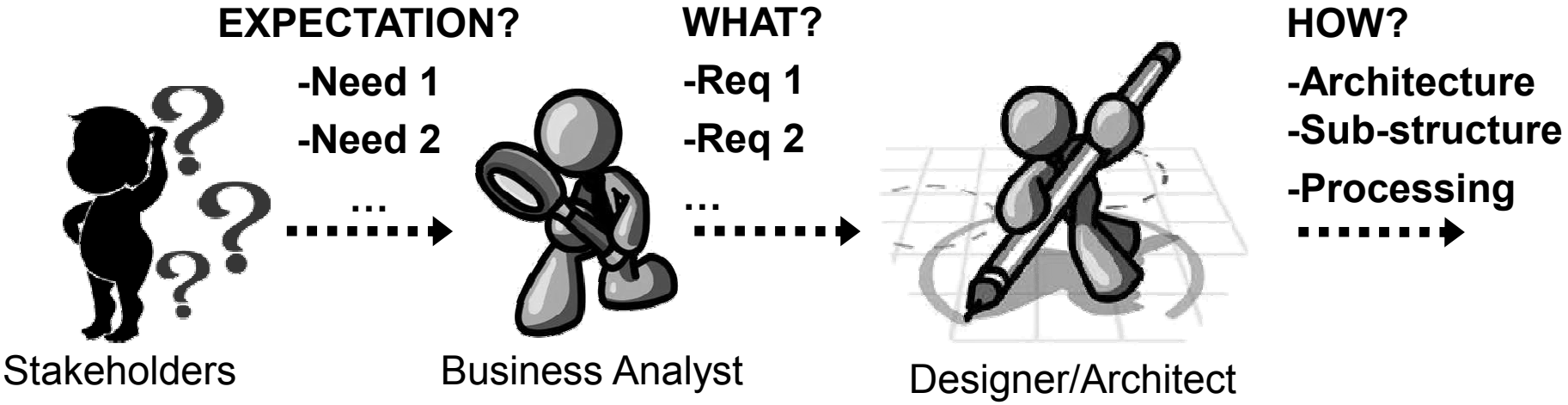


- **Architecture Concepts.**
- Architecture Models.
- Distributed Technologies.

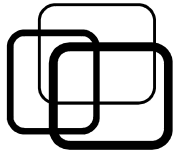
Architecture Concepts



- Software Design?
  - Answer question **HOW**.
    - Find solutions for the software.
    - Draft the way for implementation.
  - Design abstraction level:
    - Architectural design: sub-systems, architecture model.
    - Sub-structure design: screens, classes, data.
    - Process design: Use Case screenplay in details.



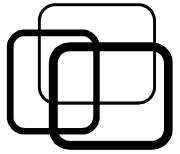
Architecture Concepts



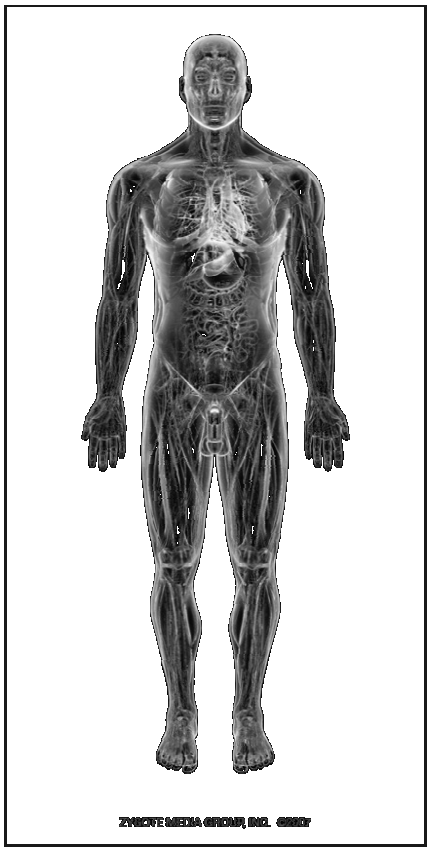
- What is architecture?
  - Simple program → single component.
  - Complex programs → multiple components.
  - Questions about components:
    - How they are organized?
    - How they interacts?
    - Component sub-structure?
  - Architecture

→Answers for the above.

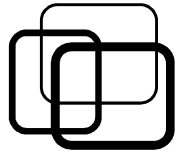
Architecture Concepts



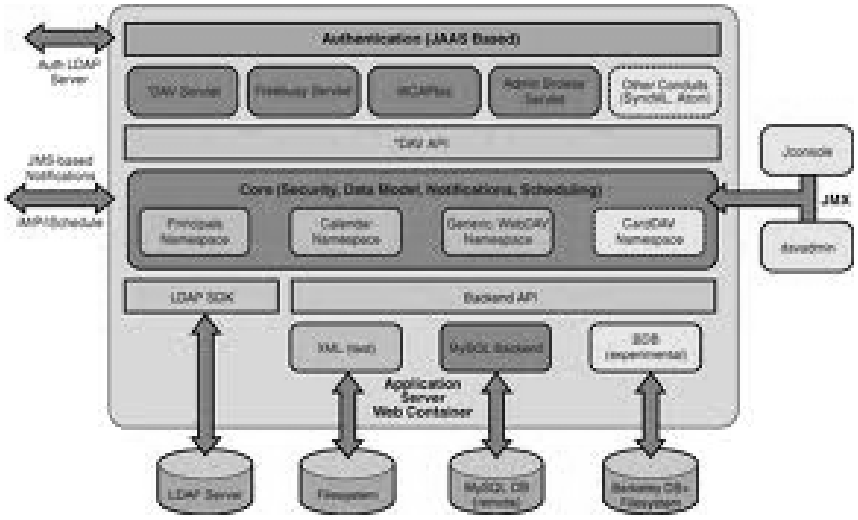
- Importance of Architecture:
  - Software efficiency:
    - Performance.
    - Reliability.
    - Security.
    - Fault-tolerance.
  - Software cost:
    - Deployment.
    - Operation.
    - Maintenance.
  - Software implementation.



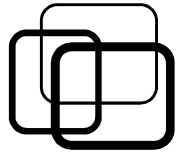
Architecture Concepts



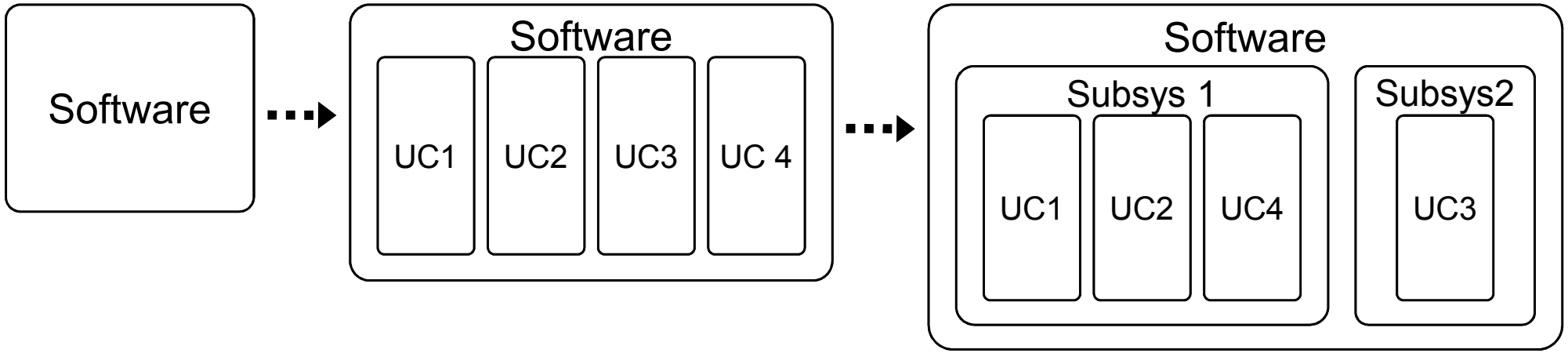
- Architectural Design:
  - First part of design phase.
  - Define system framework.
  - Activities:
    - Vertical grouping: identify sub-systems.
    - Horizontal grouping: select architecture model.



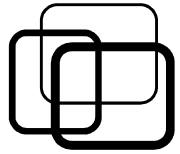
Architecture Concepts



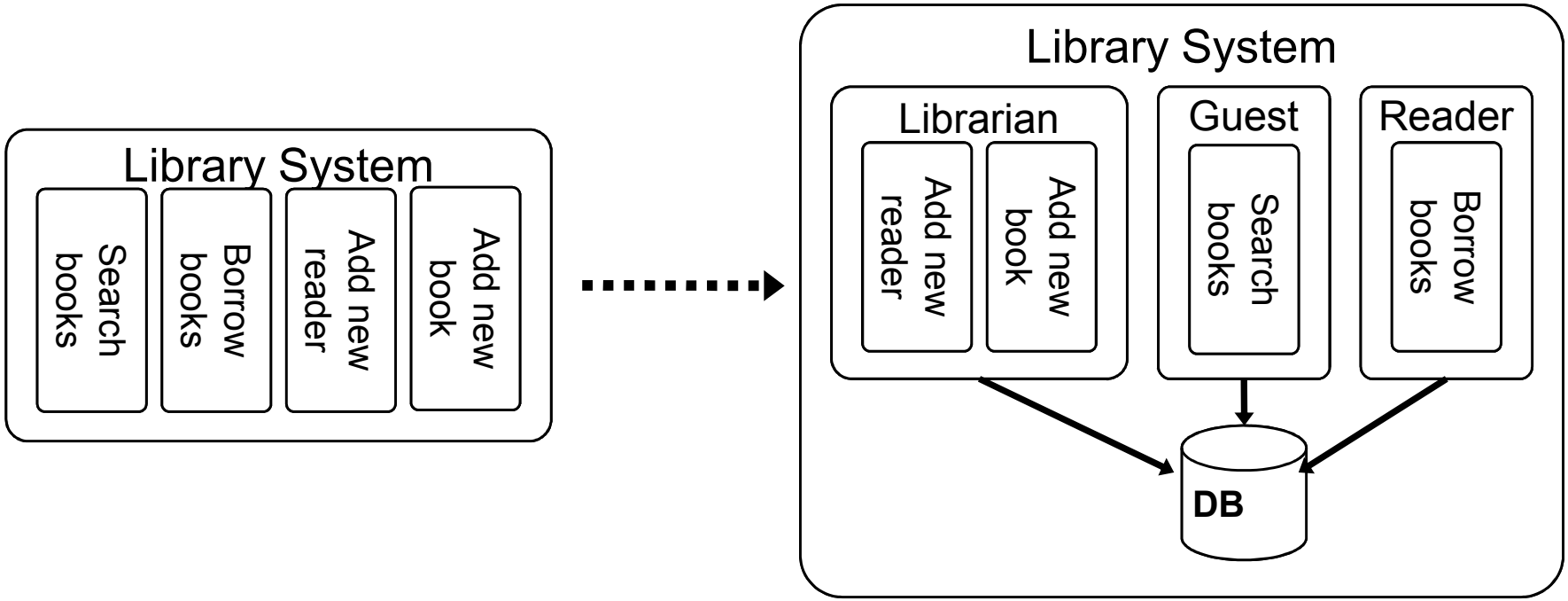
- Vertical Grouping:
  - Split into sub-systems.
    - A stand-alone component inside system.
    - Can be implemented & run dependently.
    - Group related features.



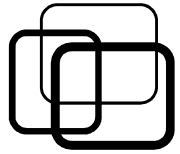
Architecture Concepts



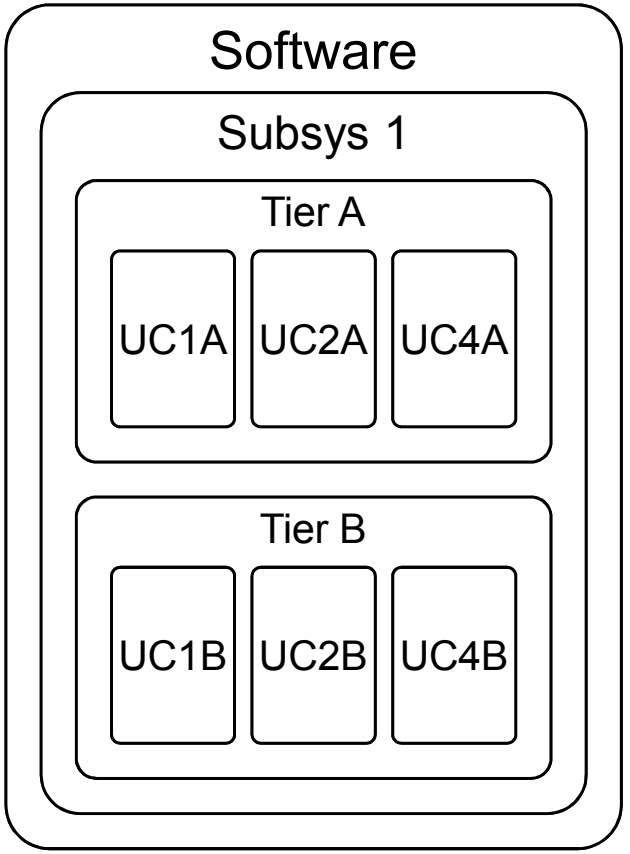
- Vertical Grouping :
  - Criteria for sub-system:
    - User security.
    - Shared libraries.



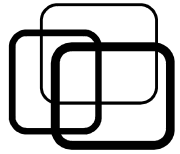
Architecture Concepts



- Horizontal Grouping:
  - Distribute level of process.
  - Based on architecture models.
- Architecture Models:
  - Monolithic Model.
  - Distributed Model:
    - 2-Tiers (Client-Server).
    - 3-Tiers.
    - Peer-To-Peer.

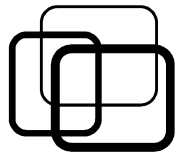


Contents



- Architecture Concepts.
- **Architecture Models.**
- Distributed Technologies.

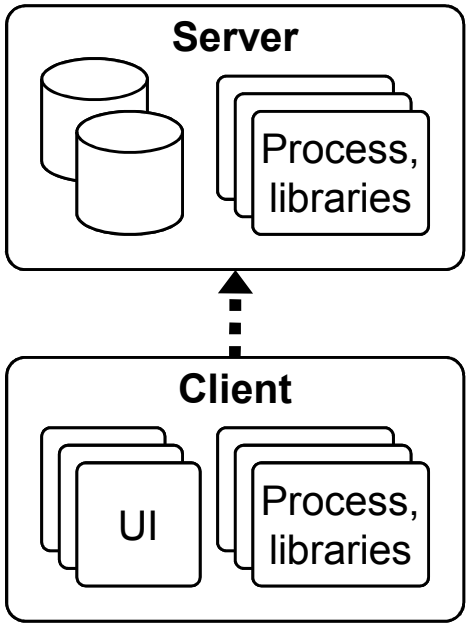
Architecture Models

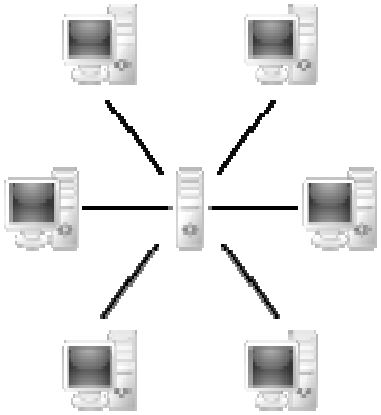


- Monolithic Model:
  - Single-tier model.
  - One integrated component.
  - Advantages:
    - Simple implementation & deployment.
    - Performance.
  - Disadvantages:
    - Data sharing.
    - Maintenance.

Architecture Models

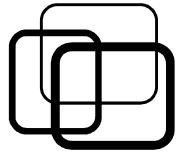
- Client-Server Model:
  - 2 sub-systems:
    - Server:
      - Service provider.
        - ➔ Data, libraries.
      - Centralized & shared.
    - Client:
      - Service consumer.
        - ➔ UI, libraries.
      - Distributed.





  - Flow of process:
    - Inside sub-system: free.
    - Between sub-systems: 1 directional from client to server.

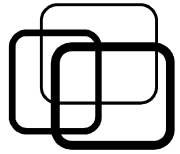
Architecture Models



- Client-Server Model:
  - Thin-Client:
    - Server: shared data + process.
    - Client: UI.

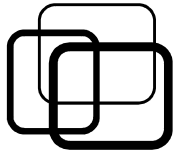
      ➔ Dump terminal.
  - Fat-Client:
    - Server: shared data..
    - Client: UI + process.

Architecture Models

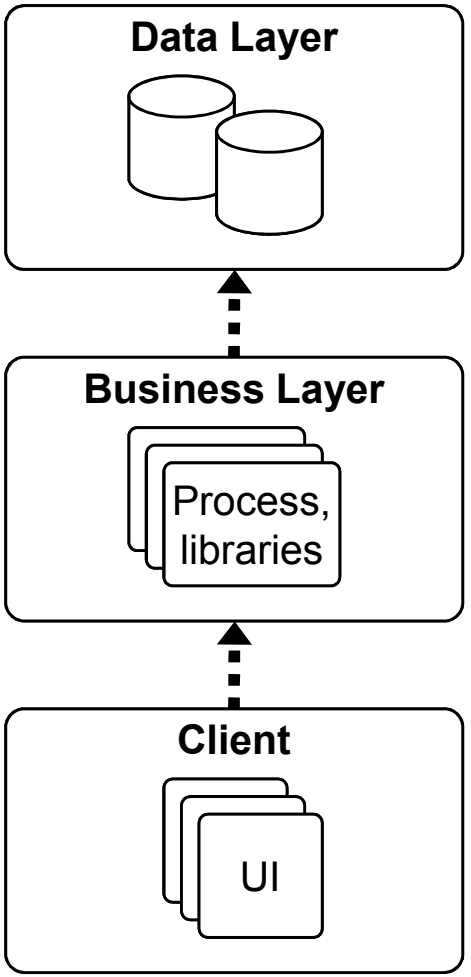


- Client-Server Model:
  - Advantages:
    - Data sharing & synchronization.
    - Flow of process:
      - ➔ Bugs isolated.
      - ➔ Maintenance.
  - Disadvantages:
    - Deployment cost.
    - Performance.

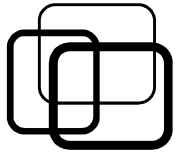
Architecture Models



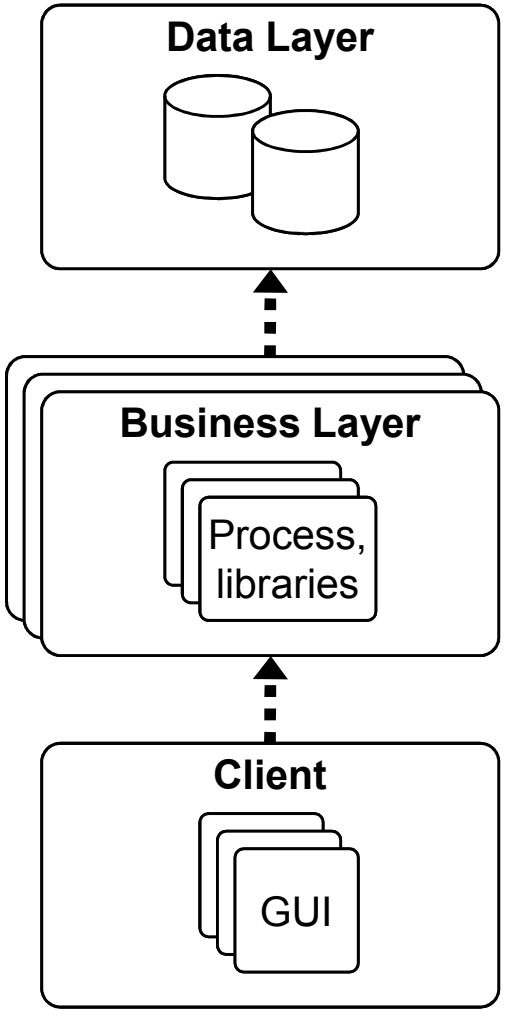
- 3-Tiers Model:
  - 3 sub-systems:
    - Data layer:
      - Data service.
      - Data server.
    - Business layer:
      - Libraries service.
      - Application server.
    - Presentation layer:
      - UI.
      - Thin-Client.
  - Flow of process:
    - From client to business to data layer.



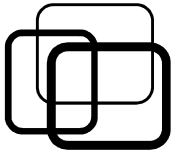
Architecture Models



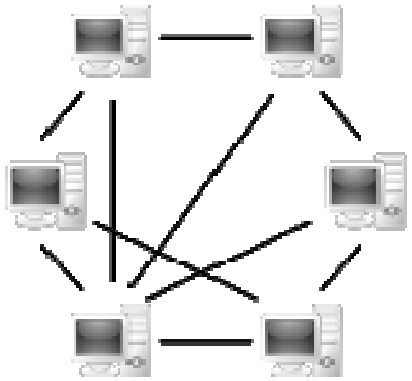
- 3-Tiers Model:
  - Multi-tier model:
    - Multiple business layers.
    - Use in complex web applications.
  - Advantages:
    - Same as Client-Server model.
    - Separated server process.
  - Disadvantages:
    - Same as Client-Server.



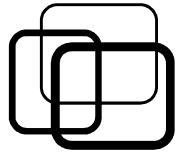
Architecture Models



- Peer-to-Peer Model:
  - Distributed monolithic model.
    - One integrated component.
    - Deploy on network.
    - Can interacts to each other.
    - Each component is Client-Server.
    - Data & process shared on network.
  - Advantages:
    - No centralized server.
    - Storage space & performance.
    - Deployment.
  - Disadvantages:
    - Implementation & data management.

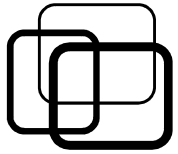


Contents



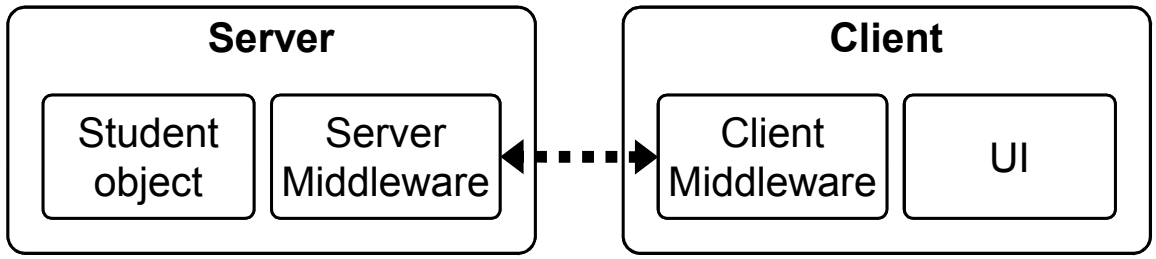
- Architecture Concepts.
- Architecture Models.
- **Distributed Technologies.**

Distributed Technologies



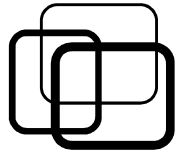
- Middleware:
  - How distributed components interacts?

    ➔ Middle component regulation.

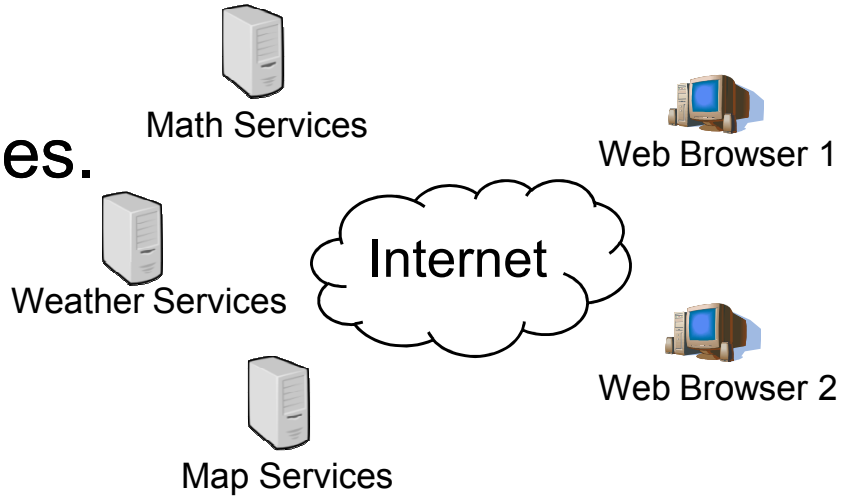


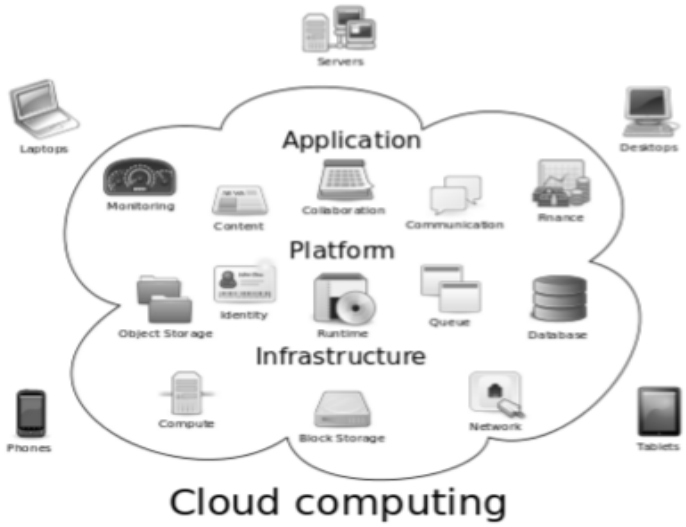
  - Middleware standards:
    - CORBA (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture).
    - COM (**C**omponent **O**bject **M**odel).
    - JavaBeans.

Distributed Technologies



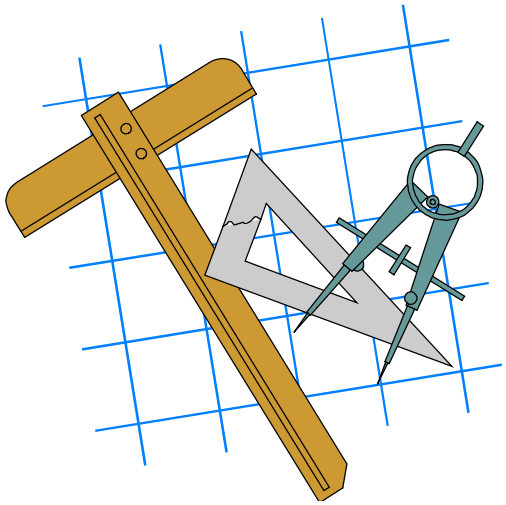
- Web Service:
  - Online programming libraries.
  - Rent as service.
  - Access through internet.
  - Popular services:
    - Math services.
    - Google map.
    - Amazon services.
  - Cloud computing.





Practice

- Architecture Design:
  - Project “Online Bookstore”.
  - Tasks:
    - Vertical Grouping:
      - Identify stakeholders & user requirements.
      - Group related requirements based on user security & performance.
      - Draw sub-systems architecture.
    - Horizontal Grouping:
      - Select an architecture model.
      - Use at least 1 web service.
      - Draw architecture for whole system.



Architecture Design Example