CISC434: Distributed Systems Architecture - I

- Architectural Styles - logical
- System Architectures - physical
- Self-management in Distributed Systems
Architectural Styles

Based on Software Engineering Principles

- Layered architectural style
- Object-based architectural style
- Event-based architectural style
- Data-centered architectural style
Layered Architectural Style

(a)
Object-Based Architectural Style

(b)
Some Definitions

- **Component** – modular unit with well-defined required and provided interfaces
- **Connector** – a mechanism that mediates communication, coordination, or cooperation among components
  - (Remote) procedure calls, message passing
Event-Based Architectural Style

- Component
- Event delivery
- Event bus
- Publish
- Component

(a)
Data Centric Architectural Style

- Processes communicate through a common (passive or active) repository
  - Web-based data services
- Event-based architectures combined with data-centered architectures

![Diagram of Data Centric Architectural Style](image)
System Architectures

- Centralized Architectures
  - Application layering
  - Multitiered architectures
- Decentralized Architectures
- Hybrid Architectures
Clients and Servers

- General interaction between a client and a server
- Connection-oriented vs. connectionless protocol – cost for setting a connection and degree of reliability in communication

![Diagram of client-server interaction]

- Client
- Request
- Wait for result
- Reply
- Provide service
- Time
Application Layering – Client Server Model

- **Client-server applications to serve user access to database**
  - The user interface level – contains all that is necessary to directly interface with the user
  - The processing level – typically contains the application
  - The data level – the actual data that is being acted on
Application Layering - Example

The general organization of an Internet search engine

User interface

Keyword expression

Query generator

HTML page containing list

HTML generator

Ranked list of page titles

Ranking component

Database queries

Web page titles with meta-information

Database with Web pages

User-interface level

Processing level

Data level

CISC 434

Lecture 4 - 11

Qu atroc
The simplest organization is to have only two types of machines:

- A client machine containing only the programs implementing (part of) the user-interface level.
- A server machine containing the rest - the programs implementing the processing and data level.
Alternative client-server organizations – two tiered architecture

Part (a): User interface, Application, Database

Part (b): User interface, Application, Database

Part (c): User interface, Application, Database

Part (d): User interface, Application, Database

Part (e): User interface, Application, Database
Examples of Alternative Client-Server Organizations

- a) Only the terminal dependent part of UI is on the client m/c and give the applications remote control over the presentation of data
- b) The UI communicates with the rest of the applications on the server through an application specific protocol, no processing at the front end
- c) The application makes use of a form that needs to filled in entirely before it can be processed. The front end then can check the correctness, and if needed, communicate with the user interactively
- d) The client is a PC or workstation and is connected through a network to a distributed file system or database
- e) Represents the situation where the client’s local disk contains part of the data – can gradually build huge cache
Multitiered Architectures

An example where the server can also act as a client – three-tiered architecture

User interface (presentation)  
Application server  
Database server

Request operation  
Wait for data  
Request data

Wait for result  
Return result  
Return data

Time
Summary

- **Architectural Styles**
  - Layered architectural style
  - Object-based architectural style
  - Data-centered architectural style
  - Event-based architectural style

- **System Architectures**
  - Centralized Architectures
    - Application layering
    - Multitiered architectures
  - Decentralized Architectures
  - Hybrid Architectures

- **Self-Management in Distributed Systems**