Today’s Topic

• Architectures for distributed systems
  – Architectural styles
  – Centralized, decentralized, hybrid architectures

Architectural Styles

• Layered architectures
• Object-based architectures
• Event-based architectures
• Shared data spaces
Layered Architecture

- Each layer uses the layer below to implement new functionality that is exported to the layer above.

Object-based Architecture

- Each object corresponds to a component
- Components interact via remote procedure calls
  - Popular in client-server systems
Event-based Architecture

- Components communicate via events
  - Use a publish-subscribe paradigm
  - Components can publish events, subscribe to events
  - Events are delivered to subscriber once published by publisher
  - Processes are decoupled in space (referentially decoupled)

Shared Data Space

- “Bulletin-board” architecture
  - Publishers post items to shared space; consumers pick up at a later time
  - Processes are decoupled in space and time
Client-Server Architectures

- Application layering
  - User-interface level
  - Processing level
  - Data level

Search Engine Example
Multitiered Architectures

- The simplest organization is to have only two types of machines:
  - A client machine containing only the programs implementing the user interface level
  - A server machine containing the programs implementing the processing and data level
Three-tier Web Applications

- Server itself uses a “client-server” architecture
- Very common in most web-based applications

Decentralized Architectures

- Peer-to-peer systems
  - Removes distinction between a client and a server
  - Overlay network of nodes
- Chord: structured peer-to-peer system
  - Use a distributed hash table to locate objects
  - Data item with key $k \rightarrow$ smallest node with id $\geq k$
Unstructured P2P Systems

• Topology based on randomized algorithms
  – Each node pick a random set of nodes and becomes their neighbors.
    • Gnutella
  • Data items randomly placed on nodes
  – Use flooding to look for an item.

SuperPeers

• Superpeers take on more responsibilities (need to be willing to donate more resources)
  – E.g. Keep an index over clients’ data
Edge-Server Systems

- Edge servers: from client-server to client-proxy-server
- Content distribution networks: proxies cache web content near the edge

Collaborative Distributed Systems

- BitTorrent: Collaborative P2P downloads
  - Download chunks of a file from multiple peers
- Use a global directory (web-site) and download a .torrent
  - .torrent contains info about the file
    - Tracker: server that maintains active nodes that have requested chunks
- Force altruism: If P sees Q downloads more than uploads, reduce rate of sending to Q