Distributed Systems

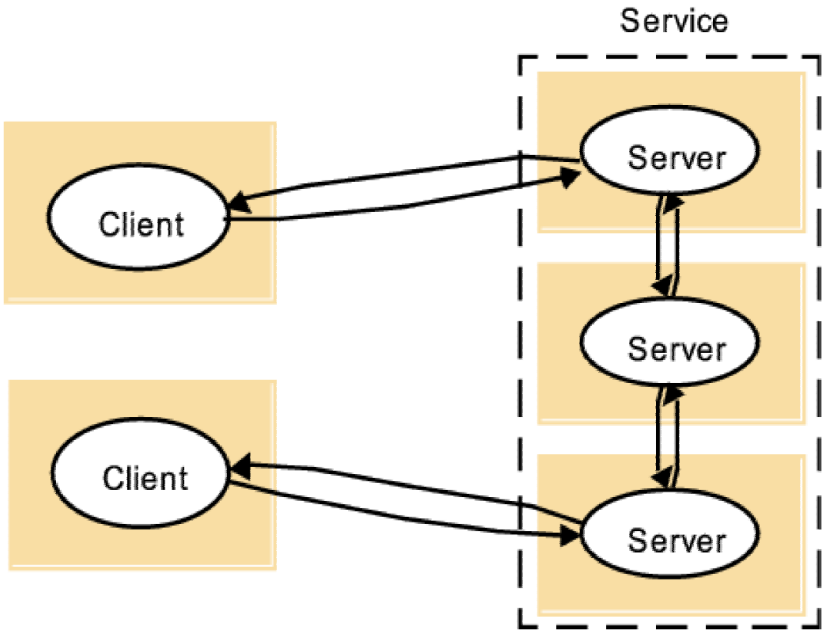
COMP90015 2018 Semester 1
Tutorial 3

Q1. Briefly explain each of the following distributed system architecture variations, giving also a reason or a benefit for its use:

- Services provided by multiple servers
- Proxy servers and caches
- Mobile code and Mobile Agents
- Network computers
- Thin clients
- Tiered Architecture

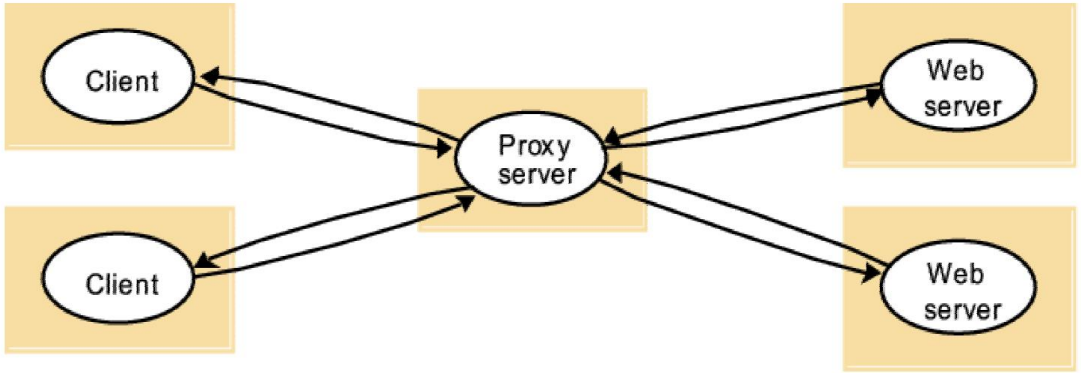
Services provided by multiple servers

- Services may be implemented as several server processes in separate host computers interacting as necessary to provide a service to client processes
- Servers may
  - Partition the set of objects on which the service is based and distribute those objects between themselves
  - Maintain replicated copies of them on several hosts
- Improve performance and reliability



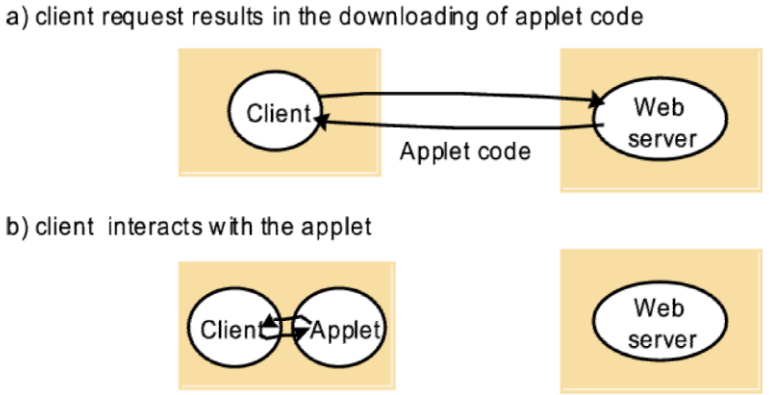
Proxy servers and caches

- Cache
  - A store of recently used data objects that is closer to the client
  - They may be co-located with each client or they may be located in a *proxy server* that can be shared by several clients
- Increase the availability and performance of the service by reducing the load on the wide area network and web servers
- Proxy servers can take on other roles --- better reliability
- Improved security
- Access restriction
- Privacy protection



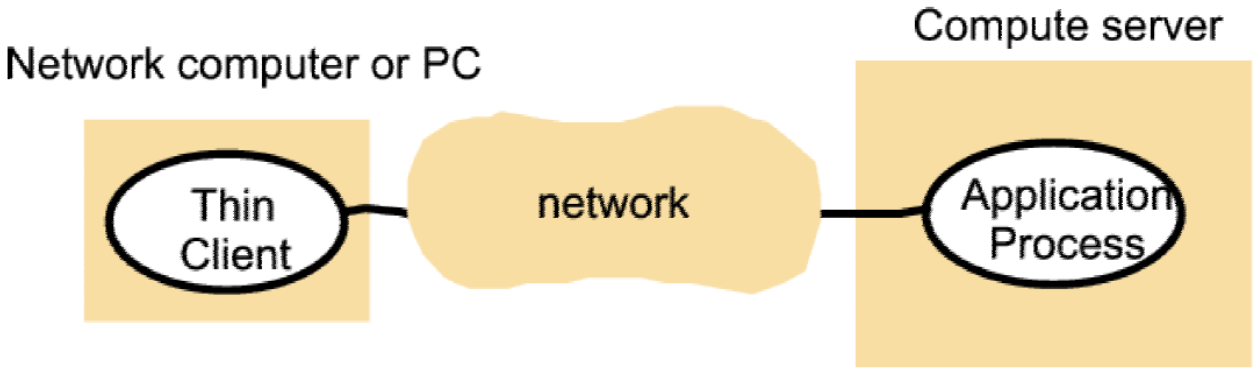
Mobile code and Mobile Agents

- Mobile code
  - Mobile Code is down loaded to the client and is executed on the client (e.g. applet).
  - Good interactive response
  - Security threat
- Mobile agents
  - Mobile agents are running programs that includes both code and data that travels from one computer to another.
  - They process data at the data source, rather than fetching it remotely
    - Less communication overhead by replacing remote invocations with local ones
  - Security threat



Network computers and thin clients

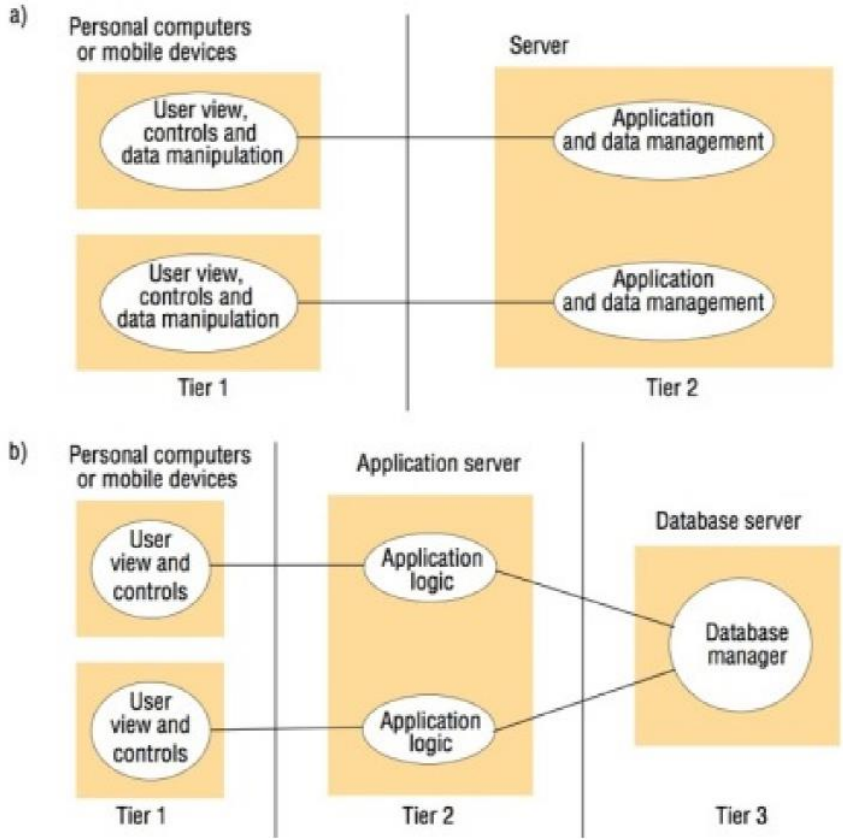
- Network Computers
  - Download their operating system and application software from a remote file system. Applications are run locally.
- Thin clients
  - Move complexity away from the end-user device
  - Local user interface, remote services or applications
  - Few assumptions or demands on the client device



Tiered Architecture

- Tiered architectures are complementary to layering, which deals with horizontal organization of services.

- Layering deals with vertical organization of services



Fundamental Models

Q2. Briefly explain the purpose of the following fundamental models and explain two important considerations for each:

- Interaction Model.
- Failure Model.
- Security Model.

Interaction Model

Models the interaction between processes of a distributed system - e.g. interaction between clients and servers or peers.

- Performance of communication channels.
  - Latency
  - Bandwidth
  - Jitter
- Computer clocks and timing events
  - Why the timestamps between two processes can vary?
    - Initial time setting being different
    - Differences in clock drift rates

Failure Model

Classifies the failures of processes and communication channels in a distributed system

- Omission failures
  - Omission failures refers to cases where a process or a communication channel fails to perform what is expected to do
- Arbitrary failures
  - Refers to any type of failure that can occur in a system
- Timing failures
  - These failures occur when time limits set on process execution time, message delivery time and clock rate drift.
  - More relevant to synchronous systems

Security Model

Identifies the possible threats to processes and communication channels, as well as protecting encapsulated objects against unauthorized access.

- Encryption.
- Authentication.
- Secure Channel.

Security Model - How to Perform Encryption

- Secret Key.
- Public/Private key pair.
- Establishing a secure channel (SSL).

Asynchronous Protocol vs Synchronous Protocol

Q3. Explain the difference between a synchronous protocol and an asynchronous protocol.

Asynchronous Protocol vs Synchronous Protocol

Q3. Explain the difference between a synchronous protocol and an asynchronous protocol.

- Synchronous communication blocks on both send and receive operations.
  - When a send is issued the sending process is blocked until the receive is issued.
  - Whenever the receive is issued the process blocks until a message arrives.
- In Asynchronous communication the send is nonblocking.
  - The sending process returns as soon as the message is copied to a local buffer and the transmission of the message proceeds in parallel.
  - Receive operation can be blocking or non-blocking (non-blocking receives are not normally supported in today's systems).