1. **What is a software architecture?**

A software architecture concerns the logical organization of the software components. And important software architectures in distributed systems such as SOA:

• Layered architecture.

• Object-based architecture.

• Service-Oriented Architecture (SOA).

1. **What is a system architecture?**

A system architecture concerns where these software components are placed across the various machines.

And important system architectures in distributed systems are:

• Centralised client-server architecture.

• Decentralised peer-to-peer architecture.

• Hybrid architecture.

• Edge computing

Describe the difference between a software architecture and a system architecture.

A software architecture concerns the logical organization of the software components, whereas a system architecture concerns where these software components are placed across the various machines.

1. **Describe the** **layered software architecture.**

layered software architecture can be used to construct programs that can be decomposed into subtask groups, each subtask is at a specific level of abstraction. The components in layered software architecture will be organized in a layered fashion, each layer provides a higher level of service for the lower level, which means a component at layer j can make a downcall to a component at a lower-level layer i (with i<j), such as UI layer call service layer. Usually the lower level layer cannot call higher level layer and only in exceptional cases will an upcall be made.

1. **Describe the object-based software architecture.**

Objects provide a way of encapsulating data and operations that can be performed on that data. And each object corresponds to a software component and these are connected through method calls.

1. **Describe the** **Service-Oriented Architecture (SOA) which is a type of software architecture.**

A Service-Oriented Architecture (SOA) is that the distributed system is constructed as a sized, suitable, independently deployed modules of different services such as web client, android client and a database service. Developing a SOA system involves service composition and each service must offer a well-defined interface, which allows component services in various systems to interact in a unified and universal way. SOA Will help us stand at a new height to understand the development and deployment forms of various components in an enterprise architecture.

**Give an example of how such an architecture may be implemented.**

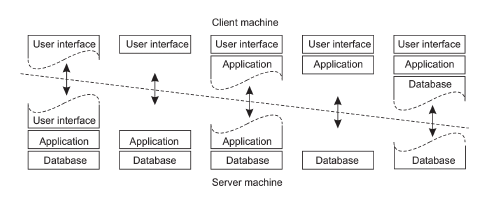
SOA can be implemented using Web services (e.g. SOAP and RESTful web services).

1. **Describe the** **client-server system architecture.**

The client-server system architecture is let its processes divided into two (possibly overlapping) groups. A client is a process that requests a service from a server by sending it a request and subsequently waiting for reply. This client-server interaction, also known as request-reply behavior.

1. **What is meant by the terms thin and fat clients?**

Thin and fat clients is a concept that exists in the context of a client-server architecture. In a fat client most of the processing and data storage is handled by the client. In a thin client most of the processing and data storage is handled by the server. Consider the image below. As we move from the left to the right image we move from a thin to a fat client.



1. **What does the network topology of a distributed system describe?**

Network topology is a graph describing the layout of the distributed system. Nodes can correspond to physical devices or processes. Edges correspond to communication channels.

1. **What is an overlay network?**

If the nodes correspond to processes, the graph is commonly referred to as an overlay network. Nodes in the overlay network can be thought of as being connected by virtual or logical links, each of which corresponds to a path in the underlying network.

1. **What is the difference between a structured and unstructured topology?**

The topology is structured if it “has a pattern”. The topology is unstructured if the edges are random. In unstructured networks many tasks, such as routing, are more complicated.

1. **Define what is meant by vertical and horizontal distribution in a distributed system.**

Vertical distribution is characterised by having logically different processes (e.g. client-server model). Whereas Horizontal distribution is characterized by having

logically equal processes (e.g. peer-to-peer model).

1. **Peer-to-peer systems support horizontal distribution. What does this statement mean?**

//TODO

1. **Describe the peer-to-peer (P2P) system architecture.**

In a client-server architecture there are two types of processes - the clients and the servers.

In a P2P architecture no such distinction exists and instead all processes are equal; that is,

all processes act as both client and server.

1. **Describe the difference between a structured peer-to-peer architecture and an unstructured peer-to-peer architecture?** **Describe how flooding may be used to search for data in an** **unstructured peer-to-peer architecture.**

The difference is the network topology. In a structured peer-to-peer architecture the network topology is structured, which means that this structured peer-to-peer architecture is more regular, and we can apply some tasks such as searching and routing on it more easily .Whereas In an unstructured peer-to-peer architecture the network topology is unstructured, which means that the unstructured peer-to-peer architecture is messy, and those tasks will be more difficult to perform on it.

In flooding algorithm, we start the searching from any node of this unstructured architecture, which is called issuing node, then the issuing node passes this data searching task to all neighbors, and these receiving nodes become a new issuing node, continually forwards the searching task to all its neighbors. Searching terminates when the data is found, or the searching times exceeded the maximum number of hops.

1. **In a peer-to-peer system, keys corresponding to data are often determined using a hash function. Explain how a hash function works.**

A hash function H(.) reduces a variable length input to a fixed length output.

A hash function will have the following properties: Computed m = H(M) is easy but computing the inverse is impossible, and if M ≠ M’ then H(M) ≠ H(M’).

1. **In a structured peer-to-peer system the chord algorithm can be used to efficiently determine the location of data. Describe with the aid of an example how this algorithm works.**

The steps:

Nodes organized in a ring topology.

Each node is assigned an identifier number.

Each data item is assigned a key number.

Data item with key k is mapped to node with smallest identifier p greater than or equal to k.

This node is referred to as the successor of key k and denoted succ(k).

Each node maintains shortcuts to other nodes.

This ensures length of the shortest path between any pair of nodes is O(log N) where N is the number of nodes.

To look up a key, a node will try to forward the request “as far as possible” but without passing it beyond the node responsible for that key.

Example:

Node 9 is asked to look up the node responsible for key 3 (which is node 4).

Node 9 has four shortcuts: to nodes 11, 14, 18, and 28.

Node 28 is the farthest node and still preceding the one responsible for key 3; it will get the lookup request.

Node 28 has three shortcuts: to nodes 1, 4, and 14.

Node 1 is the farthest node and still preceding the one responsible for key 3; it will get the lookup request.

Node 1 knows that its successor in the ring is node 4, and thus that this is the node responsible for key 3, to which it will subsequently forward the request.

1. **Describe with the aid of an example an algorithm for locating data in an unstructured peer-to-peer system.**

We can use random walk algorithm to locate data in an unstructured P2P system.

In random walk an issuing node passes a request for a data item to a randomly chosen neighbor. A receiving node forwards the request to a randomly chosen neighbor. Searching terminates when the data is found or maximum number of hops (time-to-live or TTL) is reached. To decrease waiting time, an issuer can start multiple random walks simultaneously.

1. **Describe with the aid of an example how a flooding algorithm can be used to locate data in an unstructured peer-to-peer system.**

In flooding an issuing node passes a request for a data item to all neighbors.

A receiving node forwards the request to all its own neighbors.

Searching terminates when the data is found or maximum number of hops (time-to-live or TTL) is reached.

In flooding algorithm, we start the searching from any node of this unstructured architecture, which is called issuing node, then the issuing node passes this data searching task to all neighbors, and these receiving nodes become a new issuing node, continually forwards the searching task to all its neighbors. Searching terminates when the data is found, or the searching times exceeded the maximum number of hops.

1. **Describe with the aid of an example how a random walk algorithm can be used to locate data in an unstructured peer-to-peer system.**

We can use random walk algorithm to locate data in an unstructured P2P system.

In random walk an issuing node passes a request for a data item to a randomly chosen neighbor. A receiving node forwards the request to a randomly chosen neighbor. Searching terminates when the data is found or maximum number of hops (time-to-live or TTL) is reached. To decrease waiting time, an issuer can start multiple random walks simultaneously.

1. **Describe how a hierarchically organized peer-to-peer architecture can be used to efficiently determine the location of data.**

In unstructured peer-to-peer systems, locating data can be slow in large networks (not scalable). One solution is to use super peers that maintain an index of data items on a subset of nodes. Every regular peer, called a weak peer, is connected to a super peer. All communication from and to a weak peer proceeds through that peer’s associated super peer. Reduces the number of hops required.

1. **Give an example of a hybrid system architecture?**

A BitTorrent example:

User downloads chunks of a file from other users and assemble together yielding the complete file. To get started a traditional client-server scheme is deployed. A tracker is a server that keeps an accurate account of active nodes that have (chunks of) the requested file.

It is a hybrid architecture in the sense that it combines a client-server architecture and a peer-to-peer (P2P) architecture.

1. **How would one describe the system architecture used by BitTorrent. Justify your answer.**

The hybrid system architecture could be used by BitTorrent.