# Seminar - styles

*Distributed systems - tutor version*

**Dr Leonardo Mostarda,**
**School of Science and Technology,**
**Camerino University, Italy**

Version 2.0, 1 March 2013

## Question 1

Discuss why a developer might choose to run his application over UDP rather than TCP.

## Solution 1

An application developer may not want its application to use TCP's congestion control, which can throttle the applications sending rate at times of congestion. Often, designers of IP telephony and IP videoconference applications choose to run their applications over UDP because they want to avoid TCPs congestion control. Also, some applications do not need the reliable data transfer provided by TCP. UDP is faster since does not require the setup of the connection, acknowledgment and there is less information transferred.

## Question 2

Describe the following architectural styles:

- Layered architectures
- Object-based architectures
- Data-centered architectures
- Event-based architectures
- Shared-data space architectures

For each style you should provide a short description and a graphical representation.

## Solution 2

- **layered style**: components are organised in a layered fashion where a component at layer $L_i$ is allowed to call components at the underlying layer $L_{i-1}$, but not the other way around;
- **object-based architectures**: each object corresponds to what we have defined as a component, and these components are connected through a (remote) procedure call mechanism;

- **Data-centred architectures**: evolve around the idea that processes communicate through a common (passive or active) repository;
- **event-based architectures**: processes essentially communicate through the propagation of events, which optionally also carry data;
- **shared data spaces**: Event-based architectures combined with data-centred architectures;

## Question 3

Outline advantages and disadvantages of Layered architectures

## Solution 3

**Pros.** Layered architectures increases flexibility and maintainability.

**Cons.** There might be a negative impact on the performance as we have the extra overhead of passing through layers instead of calling a component directly.

## Question 4

Outline advantages and disadvantages of pub/sub systems

## Solution 4

**Pros.** Loosely coupled. Pub/sub provides better scalability than traditional clientserver.

**Cons.** Publisher and subscriber need to be synchronised on time. More specifically many pub/sub systems will try to deliver messages for a little while, but then give up. Scalability for large systems is still a problem.

## Question 5

If a client and a server are placed far apart, we may see network latency dominating overall performance. How can we tackle this problem?

## Solution 5

It really depends on how the client is organised. It may be possible to divide the client-side code into smaller parts that can run separately. In that case, when one part is waiting for the server to respond, we can schedule another part. Alternatively, we may be able to rearrange the client so that it can do other work after having sent a request to the server. This last solution effectively replaces the synchronous client-server communication with asynchronous one-way communication.

## Question 6

What is a three-tiered client-server architecture?

## Solution 6

A three-tiered client-server architecture consists of three logical layers, where each layer is, in principle, implemented at a separate machine. The highest layer consists of a client user interface, the middle layer contains the actual application, and the lowest layer implements the data that are being used.

## Question 7

Discuss why thin clients are preferred over fat clients.

## Solution 7

The reason for this is simple: having more functionality on the client machine makes client-side software more prone to errors and more dependent on the client's underlying platform (i.e., operating system and resources). Instead the thin clients are much easier, perhaps at the cost of less sophisticated user interfaces and client-perceived performance.