1a Explain what is meant by separating policies from mechanisms in distributed systens and give an example of such a separation. *5pt*

A policy in a distributed system dictates, for example, when and how data is replicated. However, such policies may be good for one application but bad for another. Therefore, ideally, a system should provide only the means for enforcing policies. In the case of replication, a distributed system could permit an application to specify the constraints that need to be satisfied before some data is actually replicated.

1b What is a multitiered client-server architecture? *5pt*

It is a logical division in functional layers that can be implemented on the client and server machine. The usual division is into a user-interface layer, a processing layer, and a data (base) layer.

2a How are runtime remote objects commonly realized in distributed systems? *5pt*

The essential component here is to make use of an object adapter, which provides a general means for invocation. An adaptor implements a wrapper around an adaptor-specific combination of code and data, together forming an object. The adaptor is hosted by an object server capable of accepting incoming client invocations and passing these to the appropriate adaptor.

2b What are crucial differences between message-queuing systems and e-mail systems? *5pt*

Message-queuing systems provide persistent communication, and generally also support fault tolerance, guaranteed message delivery, message priorities, and efficient multicasting.

3a Besides user interfaces, what is typically considered client-side software in distributed systems? *5pt*

Basically, the software needed to (partly) implement distribution transparency: fault tolerance, replication transparency, security, access transparency, etc.

3b How can a stateless server maintain information on its clients? *5pt*

Because the server cannot store information on clients, only the clients can do it. In practice, the server sends back state information to the client that it should provide the next time it issues a request at the server. In the Web, this is usually done by means of cookies.

4 Consider a distributed system in which each mobile object has one or more human-friendly names, as well as one or more addresses. To locate an object's address using its name, it is useful to introduce a location-independent object identifier. Why? *5pt*

Each (name,address) pair needs to be stored. However, efficient naming systems require that such bindings are cached, which cannot be done effectively if addresses change all the time. By using an intermediate ID, (name,ID) pairs can be cached (because they are stable), while we then need to concentrate only on the efficient management of (ID, address) pairs.

5a Lamport timestamps do not capture causality. Explain. *5pt*

What these timestamps only do is achieve a globally consistent view on the ordering of events. However, if $C(a) < C(b)$, then we cannot conclude that a really happened before b. The problem is that Lamport timestamps do not record on which previous messages a sent and subsequently received message depends. This information is needed to capture causality.

5b Explain Ricart and Agrawala's algorithm for achieving mutual exclusion. *5pt*

Grading: *The final grade is calculated by accumulating the scores per question (maximum: 45 points), and adding 5 bonus points. The maximum total is therefore 50 points.*