EINDHOVEN UNIVERSITY OF TECHNOLOGY
Faculty of Mathematics and Computer Science

Partial examination Architecture of Distributed Systems (2XI45)
on Thursday, November 5th, 14.00h-15.30h

Work clearly. Read the entire exam before you start. Motivate each answer concisely and to the point. Maximal grades are given between parentheses. The total score sums up to 20 points.

1. (1) A distributed system can be defined as "*the hard- and software of a collection of* ***independent*** *computers that* ***cooperate*** *to realize some functionality*." Explain the meaning and relevance of the terms "***independent***" and "***cooperate***" in this definition.
   Answer: see ADS-Architectures slide 3.

2. (2) What is an architecture, by what is it described, and for what is it used?
   Answer (see ADS-Introduction slide 20):
   - An architecture is: "The fundamental organization of a system embodied by
   - its components, their relationships to each other and to the environment, and principles guiding its design and evolution."
   - It is used for *understanding*, *analysis*, *communication* and *construction*.

3. (1) Give *rules* for the Layered architectural style.
   Answer (see ADS-Architectures slide 20): Layer *N* may know only layer *N-1* and not use knowledge of lower layers than *N-1*. Control flow can be both synchronous and asynchronous.

4. (2) A motivation for naming is to delay binding. Give at least two types of transparencies in distributed systems that are supported by naming through delayed binding. Include a description of these transparencies, i.e. *what* is hidden.
   Answer (see ADS-Naming slide 3 and ADS-Architectures slide 6 or book p. 5):
   - *mobility* (or *migration*) *transparency*: name hides that an entity may move to another location
   - *replication transparency*: name hides that an entity is replicated, i.e. a name may map to different entities;
   - *location transparency*: name hides where an entity is located.

5. (3) Distributed resolution of structured naming can be iterative or recursive. Briefly describe both approaches and discuss caching and communication costs for each.
   Answer: see ADS-Naming slides 28 till 32:
   - *communication costs*: The total *number* of messages is the same for both cases. The number of messages send/received is highest for the client for the iterative case and for the servers in the recursive case. In case of co-location of (some of) the servers, the recursive case may give rise to less long distance communication (see slide).
   - *caching*: There are two issues to be considered: (*i*) *where* information is cached (client and/or server) (*ii*) for the recursive case, the *amount* of information that is cached by which server. Note that, according to the slides, the same amount information is cached at the client side for both cases.
   - *states*: The question didn't ask for the amount of state- ($\neq$ cached) information that needed to be to stored where to facilitate the resolution.

6. (2) Give three reasons for introducing processes.
   Answer: see ADS-Process slide 5.

7. (2) Client-to-server binding can be using a “DCE daemon” or a “superserver”. Explain the differences between and commonality of both approaches.
   Answer: See ADS Process slide 15 or book p. 89 + 90; essential elements:
   - *end-point*: You need to ask a daemon (at a specific end point) for an end point of a service (i.e. end points need not be pre-assigned). A superserver is associated with multiple end points.
   - *actual server*: in case of a daemon, a server registers itself, in case of a superserver, a dedicated server is created upon a request;
   - the daemon and the superserver are both a server and both mediate between client and (destination) server. The actual service is provided by another server than the daemon or superserver.

8. (2) Relays (or routers) are special queue managers in a message-queueing system that forward incoming messages to other queue managers. Describe at least two reasons including brief explanations for relays.
   Answer (see ADS-Communication slide 42 or book p. 147 - 149):
   - *scalability*: only few routers (rather than all queue managers) need to know (and therefore to be updated upon changes into) the network topology and queue-to-location mapping.
   - *secondary processing*: e.g. logging of messages for security or fault-tolerance, or transforming messages;
   - *multi-casting*: sending a message to multiple queues.

9. (1) Give a motivation for stream-oriented communication and illustrate the motivation by an example.
   Answer: See ADS-Communication slide 45 or book.

10. (1) Explain the notions data-centric consistency and client-centric consistency in your own words.
    Answer: See ADS-Replication slide 11 or book.

11. (3) Describe the notion of a monotonic-write consistent store, motivate the need for this notion by means of an example, and illustrate both a monotonic-write consistent data store and a data store that does not provide monotonic-write consistency.
    Answer: see ADS-Replication slides 31 & 32 and book p. 292 & 293.