**TECHNISCHE UNIVERSITEIT EINDHOVEN**
**Faculteit Wiskunde en Informatica**

*Examination Architecture of Distributed Systems (2IMN10), on Monday November 7, 2016, from 13.30 to 16.30 hours.*

---

Before you start, read the entire exam carefully. Answers to all questions must be motivated and stated clearly. For each question the maximum obtainable score is indicated between parentheses. The total score sums up to 20 points. This is a closed book exam, i.e., you are not allowed to use books or other lecture material when answering the questions.

1. (2 points) Describe the REST architectural style using the appropriate vocabulary, name the concepts involved, give a motivation for its usage and mention typical usage.

   **Answer.** See slide 23 of the slide set on architectural styles.

2. (2 points) Describe two approaches to locate mobile entities. For each approach give an example where it is used and indicate the extent to which it provides location transparency.

   **Answer.** Two distinct approaches are:

   **Pointer forwarding via stubs** This is done for remote method invocations to mobile objects. When an object moves to a new server, it leaves behind a client stub at its old location and installs a server stub at its new location. As long as the client stub that is left behind forwards its own location as the reply address, this provides location transparency. If, however, the client stub forwards the address of the original caller as the reply address, location transparency is broken.

   **Contact via home location** This approach is taken by Mobile IP. Here clients contact a mobile host through a home agent that is located at a fixed a priori known home address. Mobile hosts, on their journey, register at so-called foreign agents, which in turn notify the home agent of the current location (the care-off address) of the mobile host. Home agents use tunneling of IP packets to forward messages from clients to hosts. The mobile host may answer via the home agent or directly to the client thus creating triangular routing. In the first case, there is complete location transparency. In the latter case, the mobile host uses its home address in its answers, which is noticeable by routers. Alternatively, the home agent may inform the client of the host's care-off address, in which case location transparency is also broken.

3. Describe in some detail how a quorum-based protocol deals with read and write operations on a distributed data store that maintains multiple replicas for each stored object. In particular, discuss

(a) (0.5 point) the rules that state when an operation can take place,

(b) (0.5 point) the type of failures the protocol can tolerate,

(c) (0.5 point) its effect on the availability of the storage service, and

(d) (0.5 point) the resulting type of consistency for the data store,

**Answer.** Assume there are $N$ replicas per object, each equipped with a version number. Let $N_R$ and $N_W$ be numbers satisfying $N_R + N_W > N$ and $N_W > N/2$. Then a read takes places, when the client can assemble a *read quorum*, i.e., a set of at least $N_R$ replica managers that have sent the client their version of the object. The value read will be the one with the highest version number. A write takes place when the client can assemble a *write quorum*, i.e., set of at least $N_W$ replica managers that agree to update their object replica to the version supplied by the client. A quorum-based protocol can deal with situations in which the network becomes *partitioned*, because of host or link failures. In the partitions that still meet the quorum conditions operations can continue, thereby increasing the availability of the store as a whole. In cases where there exist at least two partitions, writes remain possible in at most one partition. In the other partitions, reads may still be possible but the value read may be stale. Choosing $N_R$ and $N_W$, within the given constraints, allows for a trade-off between blocking reads and writes. Note that choosing $N_R + N_W > N + 1$ results in unnecessary blocking of reads. Once the partitioned network status is resolved, outdated values are updated and consistency between replicas is restored. Hence, the data store exhibits *eventual consistency.* Although the book by Tanenbaum and van Steen is not very specific about this, the condition $N_W > N/2$ in fact guarantees that for every pair of consecutive writes there is at least one replica that sees them both. The group that sees both, then must and will agree on the order in which the writes are committed. Hence, for those replica managers sequential consistency is maintained. Note that for temporary disconnected replica managers, a single update may incorporate the effects of several writes, so although for them, in the actual execution, some writes are replaced by a single update, the result is still the same as if those writes would have all been executed, and in the order agreed upon.

4. Component-based software engineering is an approach to system design that creates software systems from independently developed components.

   (a) (1 point) Give the (generic) definition of a component.

   **Answer.** See slide 10 of the slide set on component-based software engineering (CBSE).

   (b) (1 point) What is the role of a component-framework in this approach?

   **Answer.** See slides 12 and 13 of the slide set on component-based software engineering (CBSE).

5. Consider the Chord scheme for DHTs. Assume a 6-bit identifier space, and assume that the node set $N$ is given by $id(N) = \{3, 8, 23, 44, 50, 51, 53\}$.

(a) (0.5 point) Give the finger table of node 23.

**Answer.** For a 6-bit identifier space, all finger tables have 6 entries. Recall that $FT_p[i] = succ(p + 2^{i-1})$. Hence Table $FT_{23}$ is given by:

$FT_{23}[1] = FT_{23}[2] = FT_{23}[3] = FT_{23}[4] = FT_{23}[5] = 44$, and $FT_{23}[6] = 3$.

(b) Assume that each node is aware of the identity of its predecessor and, consequently, resolves all keys for which it is responsible in zero steps.

- (0.5 point) Give a key $k, 0 \leq k < 64$ that takes 4 hops to resolve, when resolution is started at node 23.
- (1 point) How many keys take exactly 2 hops to resolve, when resolution is started at node 23?

**Answer.** First we compute the tables $FT_{44}$ and $FT_{50}$, which are given by:

$FT_{44}[1] = FT_{44}[2] = FT_{44}[3] = 50$, $FT_{44}[4] = 53$, $FT_{44}[5] = 3$ and $FT_{44}[6] = 23$.
$FT_{50}[1] = 51$, $FT_{50}[2] = 53$, $FT_{50}[3] = FT_{50}[4] = FT_{50}[5] = 3$ and $FT_{50}[6] = 23$.

Analyzing the number of hops using $FT_{23}$, $FT_{44}$, and $FT_{50}$ we find

For keys $k$, $9 \leq k \leq 23$, it takes 0 hops: $succ(k) = 23$.
For keys $k$, $24 \leq k \leq 44$, it takes 1 hop: $23 \rightarrow succ(k) = 44$.
For keys $k$, $45 \leq k \leq 50$ it takes 2 hops: $23 \rightarrow 44 \rightarrow succ(k) = 50$.
For key $k = 51$, it takes 3 hops: $23 \rightarrow 44 \rightarrow 50 \rightarrow succ(k) = 51$.
For key $k = 52$, it takes 4 hops: $23 \rightarrow 44 \rightarrow 50 \rightarrow 51 \rightarrow succ(k) = 53$.
For key $k = 53$, it takes 2 hops: $23 \rightarrow 44 \rightarrow succ(k) = 53$.
For keys $k$, $(54 \leq k \leq 63) \vee (0 \leq k \leq 2)$, it takes 3 hops: $23 \rightarrow 44 \rightarrow 53 \rightarrow succ(k) = 3$.
For key $k = 3$, it takes 1 hop: $23 \rightarrow succ(k) = 3$).
For keys $k$, $4 \leq k \leq 8$, it takes 2 hops: $23 \rightarrow 3 \rightarrow succ(k) = 8$.

Hence, the only key that requires 4 hops is $k = 52$, and there are $6 + 1 + 5 = 12$ keys that require 2 hops for their resolution.

6. Scalability is an important, but often vaguely defined, quality attribute of a distributed system.

(a) (1 point) The scalability framework presented in the lectures remedies this situation by giving a quantitative definition of scalability. What are the principle ingredients of such a framework, what are their roles, and how are they related?

**Answer.** A *scale parameter* $k$ that indicates the size of the system (both its load and its resources), a *scalability metric* $m(k)$ that measures the system at scale $k$, and a *scalability criterion* $C(k)$ that specifies a target value for the metric. These targets are expressed in the form of a relation between the metric and the criterion. Criteria can specify lower and/or upper bounds, or specify asymptotic behavior.

(b) (1 point) Name at least 2 tactics to improve the scalability of a system in practice.

**Answer.** Hide latency, load balancing, caching, etc. See slide 37 of the slide set on scalability for more details.

7. (2 points) Give the basic ingredients of an architectural description as specified by the ISO/IEC/IEEE 42010 standard. Illustrate your discussion with an appropriate UML model.

   **Answer.** For a UML-diagram see the architectural description (meta-)model (slide 31 of the introduction slide set). Of that diagram at least the boxes labelled System of Interest, Stakeholder, Concern, Viewpoint, View, Model, and Architectural Description should be present, and of course the relationships that hold between them.

8. Indicate for the following statements whether they are true or false. Motivate your answer with a short argument.

   (a) (0,75 point) Sending staleness notifications instead of full web-pages to a web-cache is a good tactic for a web-server in case the read-over-write ratio is less than 1.

   **Answer.** True.
   If this ratio is less than 1, clients will not to read all updates (writes). So, in general, their caches receive multiple staleness messages before they need to retrieve a fresh page due a read operation. Hence, this tactic avoids sending potentially large pages that will never be read.

   (b) (0.75 point) Signing messages is a security tactic that ensures accountability.
   **Answer.** True.
   In a security context, accountability means that each action can be traced to the party that performed it. A message's signature unambiguously and irrefutably identifies its sender. Hence, it provides accountability.

   (c) (0.75 point) John sits in his office behind his laptop and successfully changes his password. Then he has to go to a meeting. So, he logs out and goes to the meeting room, where he tries to login with his new password, but is denied access. He tries again with his old password and is granted access. From this he may conclude that the system violates *monotonic-write* consistency.
   **Answer.** False.
   The property that is violated is *read-your-writes* consistency, since at the login in the meeting room the password previously *written* by John in his office needs to be *read* for comparison with the password offered by John in the meeting room. Note that for read-your-writes consistency, it is essential that the same variable (the password) is accessed by the same person (John). From this scenario no conclusions can be drawn w.r.t. *monotonic-write* consistency.

(d) (0.75 point) Streaming data in isochronous transmission mode ensures bounded jitter.
**Answer.** True.
Bounded jitter implies a two-sided bound (both a lower and upper bound) on the transmission delay. This means that data packets arrive (roughly) at a fixed rate, i.e., the time between any pair of successive packets is (roughly) the same.

(e) (0.75 point) DNS cannot be used for service discovery.
**Answer.** False.
Standardly, DNS contains MX-records to locate mail servers. In addition SRV-records can be used to discover custom services.

(f) (0.75 point) In the absence of caching, recursive resolution of domain names takes less time than iterative resolution.
**Answer.** True.
At the lower levels of recursion the (sub)domains to be resolved tend to be located in geographical vicinity of each other. Hence, the RTT for those queries is shorter than for similar queries from the location where all lookups of the full and subsequent partial domain names originate during iterative resolution.

(g) (0.75 point) An architectural description must contain at least one model for each viewpoint of the viewpoint library it uses.
**Answer.** False.
Any architectural description is only relative complete. As long as all concerns of all stakeholders are properly addressed, it does not matter that some viewpoints are not covered. (Of course, most viewpoint libraries carefully select their viewpoints, so if a viewpoint is missing, i.e. there are no compliant models for it, this may indicate serious omissions in the architectural description.)

(h) (0.75 point) In CODA, a distributed file system with server-side replication, the servers that store the replicas are stateless.
**Answer.** False.
When a CODA server sends a file to a client machine for storage on its local disk cache, it includes a callback promise. This means that the client will be notified when that file is updated at the server side (through the actions of another client). Hence, CODA servers need to maintain for each replicated file a list of clients to whom they have an outstanding promise.