TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculteit Wiskunde en Informatica

Examination Architecture of Distributed Systems (2II45),
on Tuesday January 29, 2013, from 09.00 to 12.00 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 parenthesis. The total score sums up to 25 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) Give the basic ingredients of an architectural description as specified by the IEEE 1471 standard. Illustrate your discussion with an appropriate UML model.

   Answer. The system of interest, its stakeholders, their concerns and a set of models grouped into views that conform to viewpoints that cover the concerns. The UML diagram is that part of the IEEE 1471 conceptual model (see slide 28 of the introductory slideset) that contains the above entities and their relationships.

2. (2 points) Consider the Chord implementation of a DHT. Describe the reallocation of data and the readjustments to the finger tables that have to be made when a peer leaves the community, i.e., when a node leaves the ring of nodes.
   Besides indicating what has to be done, also indicate how it can be done. You do not have to specify a complete algorithm but should describe the major steps and provide an indication of their computational complexity.

   Answer. Let n be the node that is leaving the ring, and let s be its successor, i.e., $s = succ(n+1)$. First of all, the data stored on node n must be transferred to node s. The complexity of doing that is proportional to the amount of data stored on n. Secondly, for each node on the ring other than n, the entries in its finger table equal to n must be replaced by s. This can be realized by sending a "replace n by s"- message along the ring of nodes. Assuming a ring with N nodes and finger tables with m entries, the worst case complexity is $\mathcal{O}(mN)$. Assuming, however, that $N << 2^m$ and that the distribution of node identities over the key space is roughly equal-spatial, the complexity is more likely to be $\mathcal{O}(N)$.

3. (1 point) Describe the publish-subscribe architectural style using the appropriate vocabulary, name the concepts involved, give a motivation for its usage and mention typical usage.

   Answer. See slide 18 of the slideset on architectural style.

4. (1 point) Name the key principles of Risk- and Cost Driven Architecture (RCDA).

   Answer. See the correspondingly named slide in the slideset on RCDA by Eltjo Poort.

5. (1 point) Explain how quorum-based protocols for replica management work. In particular, clearly state the conditions imposed on the quorums and the reasons for doing so. To which extent is a quorum-based protocol resilient against network partitioning?

   Answer. In each operation (read or write) a predetermined number of replicas, called the quorum, needs to participate. Let N_R be the read quorum and N_W be the write quorum. Moreover, let N be the total number of replicas. For a write operation, N_W replicas need to agree to update their value. Moreover, all values have a version number and upon writing all N_W replicas associate a single new version number with the updated value which is higher than any of version numbers held by the replicas before. Upon reading, the values of N_R replicas are obtained and the one with the highest version number is selected as the outcome of the read operation. In order for this scheme to work, the quorums need to satisfy $N_W > N/2$ and $N_R + N_W > N$.

   Next, assume that due to a failure the network partitions into two parts containing P and $N - P$ replicas respectively. When $N_R \leq P$, then a read operation in the part with P replicas will succeed. Moreover, since $N_W > N - N_R \geq N - P$, writes to the other part cannot succeed. It may be the case, however, that also $N_W \leq P$, implying that both read and write operations succeed in the same part of the network. In either case, a read operation will always return the latest value. By symmetry, a similar reasoning holds when $N_R \leq N - P$. Hence, network partitioning cannot lead to reading of outdated values. Of course, upon reestablishing connectivity between the parts, write operations that have taken place in one of the parts after the disconnection occurred have to be propagated to the other part before any read operation may take place.

6. (2 points) Describe the 4 architectural views advocated by Kruchten. For each view indicate its primary stakeholders and their concerns.

   Answer. See slide 33 of the introductory slideset.

7. (1 point) Explain the notion of sequential consistency. Name a protocol by which it can be obtained.

   Answer. Informally, sequential consistency means that *all processes see the same interleaving of operations*, i.e., the results are as if the read and write operations issued by clients of the system are performed as a sequence of indivisible actions by a single server. This server may reshuffle the order of non-conflicting operations originating from distinct clients but should leave the order of operations originating from a single client untouched. A definiton such as Lamport's (TvS page 252), or the one given on slides 14 and 15 of the slideset on replication, formalize this concept. Protocols that achieve sequential consistency are a variant of the primary back-up protocol in with both writes and reads are remote, active replication protocols and Gifford's quorum-based protocol.

8. (2 points) Give the definition of a component as used within component-based soft-

ware engineering. Name three reasons for using this engineering style.

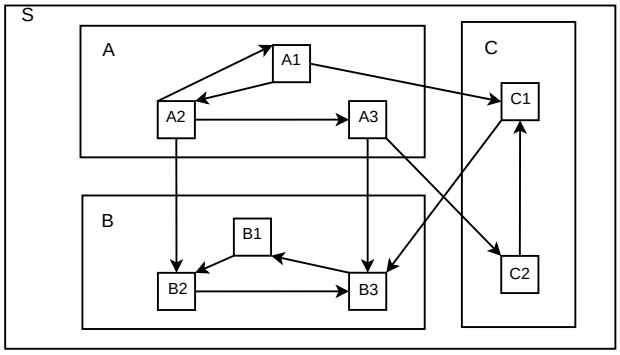
Answer. Following the definition by Szyperski: "A software component is a unit of composition with contractually specified interfaces and explicit context dependencies, i.e. no dependencies other than through interfaces. Moreover, a software component is independently deployable and subject to composition by third parties". Reasons for using this style can be found on slide 5 of the slideset on component-based software engineering.

9. (1 point) To reduce the network communication load, a system architect decides that processing should be done at the location where the data is acquired/stored. What architectural style(s) should she choose? Give an example of a real system where this occurs.

   Answer. Doing processing at the place where data is acquired requires code migration, e.g. in the form of mobile agents or applets, especially when the computation involves aggregation of data that is acquired at several distinct locations, e.g., by means of a sensor network. The Virtual Machine style enables code migration by providing a uniform execution platform. OSAS is an example of a system where this style is used.

   Also client-server architectures touch upon this issue through the decision between fat and thin clients. Thin clients put the processing at the servers side, where persistent data is stored, usually in a separate database tier. Fat clients put the processing at the client side, where the data usually originates and are considered when processing reduces the amount of data to be exchanged with the server. Virtual machines support code migration at run time. In the client-server style this issue relates more to data transport and is resolved at compile or deployment time.

10. Consider a system whose module view is given by



   (a) (1 point) Give its *part-of* relation P and its *uses* relation U.

Answer.

$$\begin{aligned} P &= \{\langle A_1, A\rangle, \langle A_2, A\rangle, \langle A_3, A\rangle, \langle B_1, B\rangle, \langle B_2, B\rangle, \langle B_3, B\rangle \\ &\quad , \langle C_1, C\rangle, \langle C_2, C\rangle, \langle A, S\rangle, \langle B, S\rangle, \langle C, S\rangle\} \\ U &= \{\langle A_1, A_2\rangle, \langle A_1, C_1\rangle, \langle A_2, A_1\rangle, \langle A_2, A_3\rangle, \langle A_2, B_2\rangle \\ &\quad , \langle A_3, B_3\rangle, \langle A_3, C_2\rangle, \langle B_1, B_2\rangle, \langle B_2, B_3\rangle, \langle B_3, B_1\rangle, \langle C_1, B_3\rangle, \langle C_2, C_1\rangle\} \end{aligned}$$

(b) (0.5 point) Indicate how relation algebra is used to determine whether this system can be (non-strictly) layered.

Answer. By using the layering rule on slide 70 of the slideset on module architecture control by Reinder Bril. Consider the order: layer A on top, layer B in the middle, layer C at the bottom. Then, the set $Absent = \{\langle B, A\rangle, \langle C, A\rangle, \langle C, B\rangle\}$ indicates dependencies that should be absent in a non-strict layering of system S. This is expressed by the following layering rule for system S

$$LR(S) \quad : \quad (Absent \downarrow P) \cap U = \emptyset$$

where $\downarrow$ denotes the "lowering"-operation. $LR(S)$ happens to be false, due to $\langle C_1, B_3\rangle \in U$. However, also the layering rules for the other 5 orderings of the layers should be considered. Doing that would reveal that by swapping layers B and C a non-strict layering is obtained.

11. For name spaces that are distributed across multiple name servers one distinguishes between iterative and recursive name resolution.

(a) (0.5 point) Explain the difference.

(b) (0.5 point) Give an argument in favor for iterative resolution.

(c) (0.5 point) Give an argument in favor for recursive resolution.

Answer. See TvS "Implementation of Name Resolution", pp 205 – 209.

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

(a) (0.5 point) There are 13 DNS root servers.
Answer. False. Although the top level of the domain name space hierarchy originally consisted of 13 categories, there are now more top domains. More importantly, however, the root server for each top domain is heavily replicated.

(b) (0.5 point) Using distributed hashing it is possible to implement constant time service discovery.
Answer. True. Through hashing the address of a peer, e.g. one holding a required service, can be looked-up in constant time, provided sufficiently large hash tables are used. As an aside, note that this also makes 1-hop routing along the overlay network possible, but that this does not guarantee any bound on the number of hops in the underlying physical network.

(c) (0.5 point) Gossiping can be used to obtain eventually consistent replicas.
Answer. True. Provided the selection of gossip partners is such that eventually each peer has been engaged in a gossip session to exchange state information with all of its neighbors. Random selection of gossip partners such that each neighbor has a positive probability to be selected will do so.

(d) (0.5 point) Scalability is determined by the process view of an architecture description.
Answer. False. The physical view is necessary as well, because it ultimately determines whether the assumptions on which the scalability of the architecture is based are fulfilled.

13. (1.5 points) Give a detailed breakdown of an RPC into basic steps and identify the architectural building blocks that are involved in executing those steps.

Answer. See slide 24 of the slideset on interaction styles. Note that the building of messages by stubs may involve marshalling (see slide 27).

14. (2 points) Give two examples of immediate service discovery. Indicate advantages and disadvantages of this type of discovery (three in total).

Answer. See slide 9 of the slideset on naming and references.

15. (1.5 points) Give a quantitative definition of availability. Name at least two tactics that can be applied to achieve this quality and give an example of each.

Answer. Availability is the probability of the system being ready to be used, as expressed by the formula $\frac{MTTF}{MTTF+MTTR}$, where $MTTF$ stands for the mean time to failure and $MTTR$ for the mean time to repair. Tactics fall in three categories: fault masking, redundancy, and fault prevention. See slides 30-37 of the slideset on quality attributes (availability and modifiability) for further details on specific tactics from these categories.

16. (2 points) Client-to-server binding can be done using a DCE daemon or a superserver. Explain the differences between and commonality of both approaches.

Answer. See slide 5 of the slideset on naming and references.