

Exam Architecture of Information Systems (192320111)

June 27, 2017, 8:45 – 10:45 hrs, Room CR 2K

Please pay attention to the following:

- This exam has to be completed in 2 hours.
- It is **not** allowed to use the book or any other material.
- Each question mentions the related topic from the book/paper.
- Calculation of mark: $1 + X/10$, where X is the number of points you earn with answering the questions (90 points is maximum).

Question 1: Motivation (5 points)

- a. Why do we need (IT/Enterprise) architectures?
- b. What important characteristics should such architectures have?

Question 2: Middleware (20 points: 5 points for each part)

Large scale IT structures which entail heterogeneous applications used in various business processes, as in the BriteLite and Janssen B.V. cases, often comprise middleware solutions to achieve a necessary or desired level of integration. There are various architecture patterns for middleware solutions. Britton & Bye distinguish between bus, hub and web services. These can be used to realize different degrees of coupling between the systems/parties connected by the middleware.

- a. What are the benefits of loose coupling, compared to tight coupling? What are the possible downsides of loose coupling, if any?
- b. How would you (relatively) score the middleware architecture patterns mentioned by Britton & Bye with respect to loose coupling? Why?
- c. How would you compare the coupling of the middleware technologies RPC and MOM (Message-Oriented Middleware, based on message queues)? Please explain.
- d. Loose coupling becomes particularly important in the B2B (business-to-business) world and in other scenarios where business processes span multiple business units or organizations. Why?

Question 3: Enterprise applications (10 points: 5 points for each part)

Traditional application design approaches can be divided according to two schools of thought, referred to as 'design up front' (typical examples are Waterfall Development and Iterative Development, both adhering to the requirements-design-implementation lifecycle) and 'design as needed' (typical examples are Rapid Application Development and Extreme Programming). The design approach advocated by Britton & Bye is called 'design in levels'.

- a. Why are architecture and system integration design poorly handled by the first two categories of approaches?
- b. What are the essential differences (mention two) between 'design in levels' and the other two categories? Why is 'design in levels' considered a better approach when it comes to handling architecture and system integration?

Question 4: Business processes (10 points: 5 points for each part)

- a. What is the difference between the single centralized and multiple centralized process patterns. Discuss how this difference will show up in an ArchiMate representation (processes at the business layer, applications and databases at the application layer)? Give a real-life example that illustrates the difference.
- b. What is the difference between the pass through and copy-out/copy-in process patterns? Discuss how this difference will show up in an ArchiMate representation (processes at the business layer, applications and databases at the application layer)? Give a real-life example that illustrates the difference.

Question 5: Resiliency (15 points: 5 points for part a.)

- a. What is the un-slippery slope and why is it called un-slippery?
- b. The Municipality of Enschede wants to create a new and autonomous institute, called RAPID ACTION. Its purpose will be to provide relief to residents and coordinate agencies in case of emergencies and disasters. Examples of such situations can be a fire incident, a terrorist attack or a heavy storm. RAPID ACTION should have access to databases of the municipality, police and hospital. The municipality of Enschede has sought your advice to design a resilient system for RAPID ACTION. RAPID ACTION should be able to access data and instruct actions to other agencies in case of disasters. Take into account that in case of a disaster, the RAPID ACTION system may fail and databases may be unavailable. Design and aptly motivate a resilient system architecture for RAPID ACTION.

Question 6: Performance and scalability (5 points)

When a thread of control wants to obtain access to a data object, it requests a lock for that object. This lock ensures that:

- no other thread of control can read that object (if it is a write lock), and
- no other thread of control can modify that object

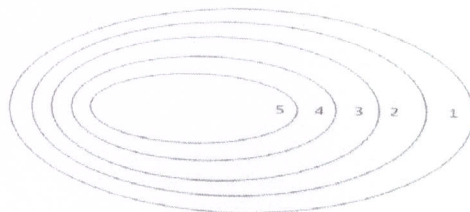
Locks can sometimes lead to a scenario of deadlock between two applications. Explain with a diagram a situation of deadlock.

Question 7: Security (5 points)

The following table shows names of some typical layers which should be considered when designing a SQL server security schema based on the onion model architecture principle.

Encryption	Authentication	SQL server permission	Data	Stored Procedures layer
------------	----------------	-----------------------	------	-------------------------

The onion model is schematically represented below (showing five layers, numbered 1-5). Match the names in the table with the layers in the figure. Provide an explanation of your choices.



Question 8: Service orientation (20 points: 5 points for each part)

With the application of the extended ArchiMate service modelling patterns, proposed by Nardi et al. (2016), one can model various outsourcing scenarios in a cross-organizational setting following XaaS strategies.

- a. Model (an example of) an IaaS application involving two companies.
- b. Model (an example of) an PaaS application involving two companies.
- c. Model (an example of) an SaaS application involving two companies.
- d. Do you think the proposed patterns can also be used to model the application of a BPaaS (Business Process as a Service) strategy? If yes, how? If no, why not?

Question 6: Performance and scalability (2 points)

When a thread of control wants to obtain access to a data object, it requests a lock. This lock ensures that:

- * no other thread of control can read that object (if it is a write lock), and
- * no other thread of control can modify that object.

Locks can sometimes lead to a scenario of deadlock between two applications. Design a situation of deadlock.

Question 7: Security (2 points)

The following table shows names of some typical layers which should be considered when designing a SQL server security scheme based on the onion model architectural principle.

Encryption	Authentication	SQL server permission	OS level protection	Physical protection

The onion model is schematically represented below (showing the layer numbers 1-5). Place the names in the table with the layers in the figure. Provide an explanation of your choices.

Question 8: Service orientation (20 points: 2 points for each part)

With the application of the extended Archimate service modeling patterns, provided by IAS4 SOA (2016), one can model various outsourcing scenarios in a cross-organizational setting following cloud strategies.