Motivation

Exercise 1.1

A. An IT/enterprise architecture is a high-level design that fosters *integration* and prevents *chaos*. Explain in general terms how an architecture achieves these qualities.
B. How can an architectural approach cope with silo applications?
C. A good architecture is designed for evolution. Explain in your own words what this means.
D. Why do we need (IT/Enterprise) architectures?
E. What important characteristics should such architectures have?

Peshmerge's answers

A. Enterprise architecture provides a holistic view of the whole enterprise. It helps us create an overview of the structure of an organization, its business processes, the application support and the technical infrastructure. It helps us with capturing the essentials of the business processes and IT and its evolution. EA enables us to have a means of communication between the various domains within an organization which is normally not the case because we often deal with a heterogeneous environment. In addition, EA facilitates change within the organization.
B. One strategy is having a "surrounding architecture" which means surrounding the existing applications with front-end and back-end without touching the code of the existing application.
   Second possible strategy is to replace those silo applications with packages. At the end, packages are also applications with their own issues. After choosing a strategy, we have then to choose between "rewrite or evolve"
C. Designing an architecture doesn't mean design once and then put on the shelf and never touch again. It's an evolutionary process and because a good architecture means that all parts of the enterprise are well and precisely defined, it's clear what every part does and which functionality it offers in the simplest way. Businesses change and therefore the architecture must also change and the software managing the application and facilitating it must also change.

D. To answer this question, take an example of an online e-commerce company. Employees of that company are people who work together to produce the firm's profits using their various platforms, such as infrastructure, software, equipment, building, and so on. Enterprise has the structure/arrangements of all these pieces/components to build the complete organization. This is the exact place where enterprise architecture plays its key role. Every enterprise has an enterprise architect.

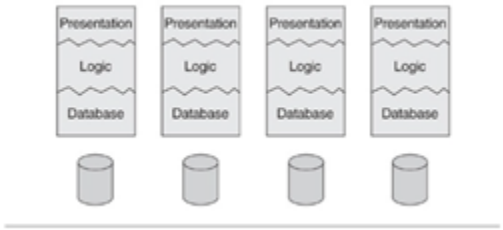
E. Robustness, scalability, reliability, and feasibility

Derek's

a. "Our view is that caring about the high level design, trying to get it right, and ensuring that all the technical aspects (application, infrastructure, security, etc.) fit together are important and are much neglected in IT design." Pg 39 Integration turns many small applications into one large application.

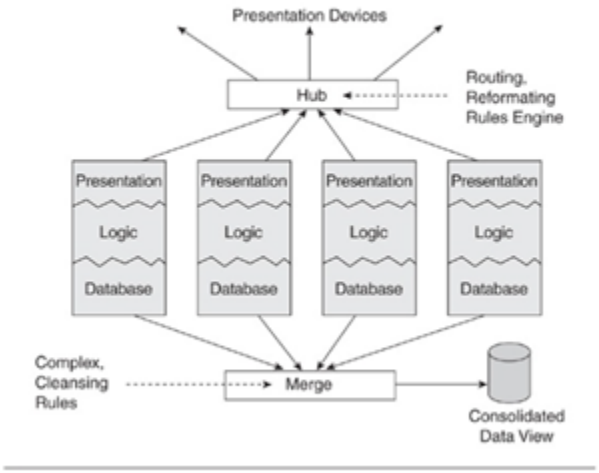
b. Silos are standalone applications.

Figure 1-6 Silo applications



One alternative we call the “Surround Architecture.” The idea is to surround the existing applications with a front end and a back end. Pg 55

Figure 1-7 Surround architecture



c. A good architecture should be open to future changes and to be resilient.

Middleware technologies

Exercise 2.1

Imagine we have two applications A and B which communicate using Message Queuing. A sends a request to B and B sends one response back for each request. Because processing a request is computationally expensive, there are three servers running application B. For reasons of resiliency, there are two other servers running application A. There are no databases involved in the applications.

Draw a picture with both applications, all servers and all queues needed in this setup. Mark all arrows with “request”, “response” or “both”. Explain the picture.

Peshmerge’s answers

Two queues are needed here: One queue where A sends a message to it and B subscribes to it. Another queue where B sends a message and A subscribes to it.
In addition, a load-balancer is needed for each application. The load-balancer would connect each application with the two servers behind them.

Exercise 2.2

A. It is said that it is important for distributed transactions to be conforming to the ACID properties. Explain what ACID properties are. Give a short explanation for each of them.

B. Message Queueing is one of the communication concepts often used by middleware. Explain the concept of message queueing.

A. Peshmerge’s answers

A is for atomic; the transaction is never half done. If there is any error, it is completely undone.

C is for consistent. the transaction changes the database from one consistent state to another consistent state. Consistency here means that database data integrity constraints hold true.

I is for isolation: data updates within a transaction are not visible to other transactions until the transaction is completed.

D is for durable; when a transaction is done, it really is done and the updates do not at some time in the future, under an unusual set of circumstances, disappear.

B.Peshmerge's answers

The concept is very powerful. In a normal situation we have three parties involved. The publisher, the subscriber party which subscribes to the queue and the message queue itself. The publisher makes connections with the message broker and publishes a message to the message queue. The message queue stores the message, the message stays in the queue until a subscriber requests the message and consumes it.

Derek's :

a. The definition of ACID properties for distributed databases is that it is a set of properties that guarantee the reliability of database transactions. ACID defines properties that traditional transactions must display; they are Atomicity, Consistency, Isolation and Durability.

Atomicity refers to the ability of the DBMS to guarantee that either all of the tasks of a transaction are performed or none of them are. Atomicity states that database modifications must follow an "all or nothing" rule. If some part of a transaction fails, then the entire transaction fails, and vice versa.

The Consistency property ensures that the database remains in a consistent state, despite the transaction succeeding or failing and both before the start of the transaction and after the transaction is over.

Isolation refers to the requirement that other operations cannot access or see the data in an intermediate state during a transaction. The Isolation property can help to implement concurrency of databases.

Durability states that once a transaction is committed, its effects are guaranteed to persist even in the event of subsequent failures. That means when users are notified of success, the transactions will persist, not be undone and survive from system failure.

In the book page 83.

B.The basic architecture of a message queue is simple; there are client applications called producers that create messages and deliver them to the message queue. Another application, called a consumer, connects to the queue and gets the messages to be processed. Messages placed onto the queue are stored until the consumer retrieves them.

Message queues

A message queue provides an asynchronous communications protocol, which is a system that puts a message onto a message queue and does not require an immediate response to continuing processing. Email is probably the best example of asynchronous communication. When an email is sent, the sender continues to process other things without needing an immediate response from the receiver. This way of handling messages decouples the producer from the consumer so that they do not need to interact with the message queue at the same time.

In the book page 88

Exercise 2.3

A simple debit/credit transaction is one in which a person wants to move money from Account *A* to Account *B*. Figure 2-8 from the book illustrates a solution using distributed transaction processing. In this solution, the debit on Account *A* and the credit on Account *B* are both done in one distributed transaction. Figure 2-9 from the book gives an alternative solution: debit/credit transaction using message queuing.

A. Are the solutions of Figure 2-8 and Figure 2-9 conforming to the ACID properties?
B. What disadvantages of the solution of Figure 2-8 are solved by the solution in Figure 2-9?
C. Discuss why it is, or is not, a good choice for an organization to use message queuing for real time processing of distributed transactions.

Peshmerge's answers

A. For 2-8, Yes it's conforming to the ACID properties because any failure anywhere would stop the whole operation. FOr 2-9 no, because it's not Atomic.
B. The disadvantages are the degradation of performance and if any involved part is down, then the whole operation won't go further.
C. It's in my opinion not a good choice because it's likely that the ACID properties will be violated by this approach. I would go for the first one which is more secure and reliable.

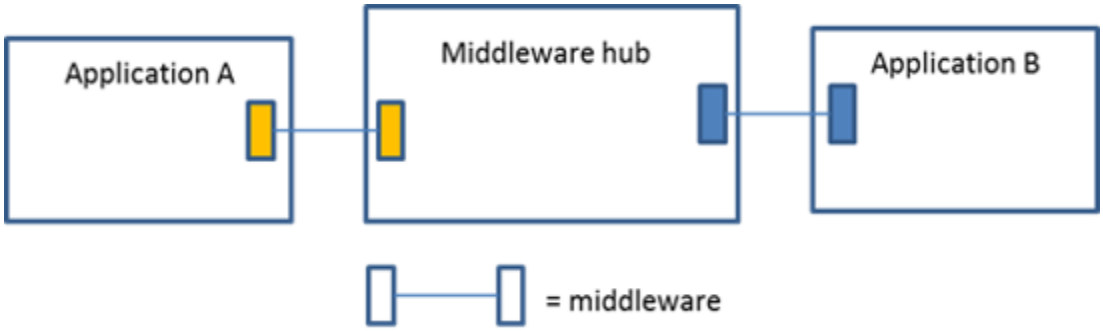
Derek's

There are 2 reasons: page 199-200, chapter 6.1

1. If two transaction servers communicate by message queuing, they can't support distributed transaction processing across them.

2. Real-time calls have an end user waiting for the reply; if there is no reply, the user eventually gives up and goes away. Real-time calls always have a timeout. With message queuing, if the end user goes away, a reply message may still be put into the output queue. This message ends up as a "dead message," and the message queuing software will typically put messages that haven't been read for a long time into a "dead letter box."

Exercise 2.4



The figure depicts a middleware hub. Consider applications *A* and *B*.
Assume that *A* needs to request a service from *B*.
A is designed to use message queues (MQ) to talk to other applications: to use a service, *A* sends a request message to *B* and then waits for a message sent by *B* that contains the response.
B is designed to use remote procedure calls (RPC).

A. The middleware hub is used to convert from MQ to RPC. Explain how this can be done. Which message queue(s) are needed? Where are the stubs?

B. Suppose that *B* is the account management system of a bank. The service that *A* requests is to increase the balance of a certain account with *x* Euros. Two types of MQ technologies are available for the middleware hub: one that guarantees *at least once* delivery and one that guarantees *at*

most once delivery of each message. In case of at least once delivery, what may go wrong? Can the hub and *B* avoid this? Same questions for at most once delivery.

This example is mentioned in the MID book on page 185. Read there for more inspiration. Chapter (5) and this relates to Chapter (13) (task/message diagrams). The keyword in this whole problem is the integrity problem!

Peshmerge's answers:

A. Hub middleware is one of the techniques that enables middleware interoperability. Application A publishes a message to a message queue. The Hub in this case is a consumer or a subscriber. Once the HUB detects that there's a message sent by application A it starts a RPC call to application B. The hub is now a client and B is acting as a server. The stubs would reside in the hub because it will act as our client in this case.
B. In case of at least once, money could be sent twice to B if B crashes after consuming the message received from A. At most once, B may crash after receiving the message from A and then would not send the request again. We can avoid this in both situations by sending a confirmation response back to A, once this happens, A must consider the operation completed. This will be done at the cost of network overhear and multiple delays.

Derek's

a. The middleware represents the links between a lot of servers, and application B is among the servers' connections. Both message queues are needed. The stub will be at the middleware, it converts the parameters into strings of bits.
b. 1. The message could wrongly add the message with future trying.

   2. The future trying may not be included in the balance.

Exercise 2.5

Compare the middleware classification of He & Xu relate with the classification of Britton & Bye? What are the (relative) benefits and limitations per class identified by He & Xu? For similar classes, are these benefits and limitations the same as the ones identified by Britton & Bye?

Peshmerge's answers:

He & Xu Middleware is designed to help developers and make distributed systems design less challenging. By adopting the middleware, we can easily separate applications from the underlying operating systems, hardware, and network environments. There are many types of middleware such as RPC-based middleware, message-oriented middleware, event-based middleware, database middleware, transaction processing (TP) monitors, security middleware, agent-based middleware, and service oriented middleware.

Derek's

Classification by He&Xu:

1. Distributed Computing environment.
2. Distributed Component Object Model (DCOM)
3. Common Object Request Broker Architecture
4. Java Remote Method Invocation (RMI)
5. Message Oriented Middleware (MOMpage 3 in thesis

Classification by Britton&Bye :

1.real-time transaction-oriented middleware

2. message queuing

3. remote database access Chapter 2

Exercise 2.6

In IV.A.3), last paragraph, He & Xu say that "Service-oriented integration depends on Enterprise Service Bus (ESB)" and later that "SOA-based ESB is often viewed as a new middleware technology". What is ESB? The basis for service-oriented integration, or based on SOA? Can you explain how ESB accomplishes integration using Fig. 2? How would an application (instance of a blue box in Fig. 2) of an arbitrary supplier be able to exchange and use information from another application through ESB?

Peshmerge's answer

What is ESB?

An Enterprise Service Bus (ESB) is fundamentally an architecture. It is a set of rules and principles for integrating numerous applications together over a bus-like infrastructure. ESB products enable users to build this type of architecture, but vary in the way that they do it and the capabilities that they offer. The core concept of the ESB architecture is that you integrate different applications by putting a communication bus between them and then enable each application to talk to the bus. This decouples systems from each other, allowing them to communicate without dependency on or knowledge of other systems on the bus.

The basis for service-oriented integration, or based on SOA? Can you explain how ESB accomplishes integration using Fig. 2?

SOA is an architectural pattern where services are loosely coupled. The service would communicate using Rest API or SOAP. Because the services are loosely coupled and they have the basis of interaction with each other, the ESB comes very handy as it facilitates the connection between the services. The whole idea of ESB is integrating and connecting distributed applications and since SOA applications are built with this idea in mind (loosely coupled, self contained, independent services). It helps to realize it.

The communication between the application and the bus is realized through an adapter which marshals and un-marshalls the data when sending and receiving the data from and to the ESB.

Derek's

ESB is able to work across different middleware products and standards to implement enterprise-wide SOA. ESB can shield from different protocols to implement smooth data flow and exchange among different application systems.

Based on the figure, different apps& systems communicate, share and exchange info with ESB. If an application want to exchange info with another application it can share it through ESB and then the other application can access it via ESB.

Exercise 2.7

In IV.B of the article of He & Xu, grid and cloud computing are mentioned as integration solutions from a users' perspective. What is the difference between grid and cloud computing? (What about cluster computing and distributed computing?) What is a private cloud, public cloud, and hybrid cloud (not mentioned in article, but relevant to enterprises which consider adoption of cloud computing)?

Pehsmerge's answer:

What is the difference between grid and cloud computing? (What about cluster computing and distributed computing?)

GRID vs Distributed computing

The key distinction between distributed computing and grid computing is mainly the way resources are managed. Distributed computing uses a centralized resource manager and all nodes cooperatively work together as a single unified resource or a system. Grid computing utilizes a structure where each node has its own resource manager and the system does not act as a single unit.

GRID vs Cloud computing

Grid computing involves virtualizing computing resources to store massive amounts of data, whereas cloud computing is where an application doesn't access resources directly, rather it accesses them through a service over the internet. In grid computing, resources are distributed over grids, whereas in cloud computing, resources are managed centrally. Let's take a brief look at the two computing technologies

What is a private cloud, public cloud, and hybrid cloud (not mentioned in article, but relevant to enterprises which consider adoption of cloud computing)?

Public clouds are the most common type of cloud computing deployment. The cloud resources (like servers and storage) are owned and operated by a third-party cloud service provider and delivered over the internet. With a public cloud, all hardware, software, and other supporting infrastructure are owned and managed by the cloud provider. Microsoft Azure is an example of a public cloud.

Private cloud:A private cloud consists of cloud computing resources used exclusively by one business or organization. The private cloud can be physically located at your organization's on-site

datacenter, or it can be hosted by a third-party service provider. But in a private cloud, the services and infrastructure are always maintained on a private network and the hardware and software are dedicated solely to your organization.

Hybrid cloud: A hybrid cloud is a type of cloud computing that combines on-premises infrastructure—or a private cloud—with a public cloud. Hybrid clouds allow data and apps to move between the two environments.

Derek's

Grid computing has been adopted to solve some complex integration problems while cloud computing has been used to support the integration of distributed application.

Grid Computing VS Cloud Computing

Criteria	Grid Computing	Cloud Computing
User Management	Decentralised management	Centralised management
Dependancy	Other computer picks up the work whenever the computer stops	Totally dependent on internet
Operation	Operates within a corporate network	Can also operate through the internet
Accessibility	Through Grid middleware	Through standard Web protocols
Domains	Multiple Domains	Single Domain
Scalability	Normal	High
Architecture	Distributed computing architecture	Client-server architecture
Virtualization	Data and computing resources	Hardware and software platforms
Computation	Maximum computing	On-demand
Application Type	Batch	Interactive

Distributed refers to splitting a business into different sub-services and distributing them on different machines. Cluster refers to a group of servers that are grouped together to achieve the same business and can be considered as one computer. Web

Private Cloud: the cloud services used by a single organization, which are not exposed to the public. A private cloud resides inside the organization and must be behind a firewall, so only the organization has access to it and can manage it.

Public Cloud: the cloud services are exposed to the public and can be used by anyone. Virtualization is typically used to build the cloud services that are offered to the public. An example of a public cloud is Amazon Web Services (AWS).

Hybrid Cloud: the cloud services can be distributed among public and private clouds, where sensitive applications are kept inside the organization's network (by using a

private cloud), whereas other services can be hosted outside the organization’s network (by using a public cloud). Users can them interchangeably use private as well as public cloud services in every day operations.

Exercise 2.8

Are grid and cloud computing really integration solutions, as He & Xu claim, or are they just packaging solutions that internally apply integration techniques? Consider different stakeholders (especially grid/cloud providers and grid/cloud users) and integration in relation to their business goals.

Peshmerge's answer

They are indeed integration solutions from the users' perspective. However, for the grid/cloud providers

Exercise 2.9

In V.A.1), He & Xu mention ESA as an integration architecture. However, ESA has not been mentioned before. What is ESA?

The Enterprise Services Architecture (ESA) on the other hand, is SAP's enhanced version of the Service-oriented Architecture (SOA). The function of ESA is to deploy web-based services at the enterprise level.

Derek's

The Enterprise Services Architecture (ESA) on the other hand, is SAP's enhanced version of the Service-oriented Architecture (SOA). The function of ESA is to deploy web-based services at the enterprise level.

It streamlines the process of idea generation and its subsequent development of innovative application based programs within the existing infrastructure while minimizing the cost and allowing timely implementation of new business processes.

ESA is based on an architectural pattern using which multiple applications, components and even systems can communicate with each other in accordance with the SOA principles. The limelight is mostly on heterogeneous applications that interact with each other and achieve a common business objective.

Exercise 2.10

Whereas the 'interoperability stack' (lecture 1) focuses on functional requirements on interoperability and integration, non-functional aspects are also important for effective integration. He & Xu mention the increasing importance to ensure end-to-end Quality of Service

(QoS). Can you give an integration example where QoS properties and/or requirements hamper successful integration?

Peshmerge's answer

If we take performance as a requirement for QoS then integrating two systems using message queueing to ensure that no messages are lost, means that the system would be dependent on each other and wait for responses before going further with any operation.

Exercise 2.11

When discussing future trends and research challenges in V, He & Xu hint several times at the potential of ontologies for addressing integration problems. What is ontology-based integration, and why/how could it be used for "some research challenges"?

Ontologies can be used in an integration task to describe the semantics of the information sources and to make the contents explicit. Ontologies help us mapping all elements, details and semantics of a system. They ensure a common understanding of information and that they make explicit domain assumptions. This is requirement #1 when working on enterprise architecture.

Enterprise applications and business processes

Exercise 3.1

Based on what you have read in section 6.1, can you identify in the case that you have chosen:

1. **Communication among applications that fall under the real-time or request-response category?**
2. **Communication among applications that fall under the deferrable or send-and-forget category?**

Describe the properties of the identified instances of communication that make you decide to categorize it as real-time or deferrable. Page 194

Peshemerge's answers:

Real-time would be chosen in cases where the answer to a question/request must immediately be fulfilled or given. For example, a user trying to login onto a system. This is only and only a real-time process. You can't make a deferrable kind of work or whatsoever.

A user placing an order on the webshop and paying the due amount via a third party payment gateway could use deferrable communication.

Exercise 3.2

Based on what you have read in section 6.1, how would you characterize the support provided by applications (for business processes, for collaboration, for business intelligence) in the organization of the case that you have chosen? Is there only one class of support, or multiple classes? Can you give examples of each, in case of multiple classes of support?

Exercise 3.3

What are the middleware-based architecture patterns for building distributed applications? Give characteristics, relative advantages, and relative disadvantages of each pattern.

Peshmerge's answer:

1. **Middleware bus (or ring) architecture:**
   In most cases, the primary aim was to separate the presentation channels from the

business services.
advantages: fast, secure and flexible
disadvantages: become fast, very outdated. Tightly coupled architecture. difficult to test, deploy, and change.

2. **Hub-and-spoke architecture:**
   The basic idea of a hub is a server that routes messages. A hub can also be used to bridge different networking or middleware technologies.
   advantages: route messages, reformat message, multicast or broadcast messages.
   disadvantages: The hub is potentially a single point of failure, so you will probably want to have a backup hub and failsafe software.
3. **Loosely coupled architecture (Web services architecture)**
   Web service architectures use the technologies that implement the Web services standards such as SOAP, WSDL, and UDDI.
   Advantages: widely implemented, cheap, often bundled with other technology, The Web services standards are designed to work over the Internet so, for instance, they don't run into difficulties with firewalls.
   Why are web services better?
   Because Web services are cheaper, integrate with more applications, and run over backbone TCP/IP networks without problems.
   Disadvantages: slower because of the need to translate messages into XML format.
4. Ad-hoc or point-to-point architecture: this is what most organizations do. It's a mess DON'T USE IT.

Exercise 3.4

Traditional 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'.

1. **Why are architecture and system integration design poorly handled by the first two categories of approaches?**
2. **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?**

Peshmerge's answers:

1. Because of the assumption that the requirements are easily grasped and seldom change. In addition, both methods don't have any long-term architectural vision. THey more focused on requirement gathering.

2. The most striking difference is that the design in levels has levels—it forms a hierarchy. The second large difference is the role of analysis. It is not an analysis of the requirements, but an analysis of the solution to see whether it meets the requirements. Doing business process level design prevents silo development.

Exercise 3.5

The three levels identified by Britton & Bye are business process level, task level, and transaction level. They suggest that the business process level design is always done first. (In terms of ArchiMate, which is also based on the 'design in levels' approach, this would translate to: the business level is always done first.) Does this contradict with the rule from Chapter 1 in the book: "Ensure that the technical issues are addressed up front to avoid developing systems that work functionally but are operationally unusable"? Why (not)?

No this doesn't contradict. Once the business processes are mapped and are understood by the team, then the work can start to address the technical issues. In the end, IT is not the end, but the mean. IT must serve the business processes.

Exercise 3.6

Can you think of possible IoT applications that could benefit the organization of the case you have chosen? Think of the three IoT categories of enterprise applications identified by Lee & Lee: (1) monitoring and control, (2) big data and business analytics, and (3) information sharing and collaboration.

1. Briefly describe the IoT applications that you see fit, and how each of them potentially adds value to the organization.
2. Are there privacy, security and safety challenges associated with these applications? Explain the difference between privacy, security, and safety by giving examples of a challenge for each of them in the same IoT application (among those that you have described in (a)).

DUNNO

Exercise 3.7

Can you think of possible big data applications that could benefit the organization of the use case you have chosen? (Groups that have chosen the Janssen Logistics case may find inspiration from the “Big Data at Schneider National” case study). Think of the objectives for big data, as identified by Davenport and Dyché: (1) cost reduction, (2) time reduction, (3) new product and service offerings, and (4) decision support.

DUNNO

Exercise 3.8

Related to exercise 3.7, discuss for one of the identified big data applications any needs or issues you see to reconcile or integrate big data with the existing (and presumably more traditional) sources of data, and to merge big data technologies with the existing infrastructure.

Exercise 4.1

Any business organization can be seen as an aggregation of several business processes. According to Britton & Bye, a clear understanding of the business processes in the organization is necessary for building a robust IT architecture for supporting the business.

1. Why do you think that is necessary ? You may also use an example to explain.

   It is necessary bcs business processes are units of internal behavior and almost everything that a business does can be described as a business process. A process is a series of activities that delivers sth, be it concrete or abstract, a solid thing or info. Even the IT department has processes.

   Ex: building a house . Pg 405

2. What is the difference between a business function and a business process. Explain with the use of an example.

   A business function is the area that the organization wants to pay attention to in order to support its business goals. Ex: by putting energy into, structurally committing resources etc. a business function can be positioned as a grouping of internal behavior based on certain criteria for ex location (same department), communication, required skills, shared resources and shared knowledge.

   A business process is a unit of internal behavior or a collection of causal related units of internal behavior with the goal of producing a predefined collection of products and services. It is triggered by a business event or by other processes.

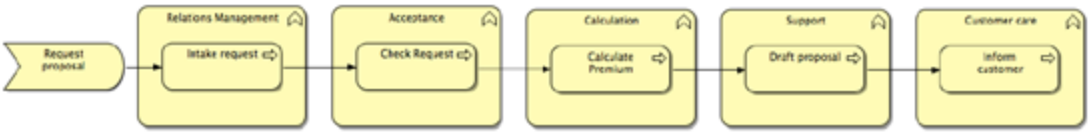
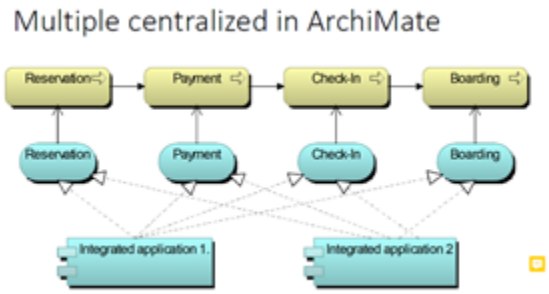
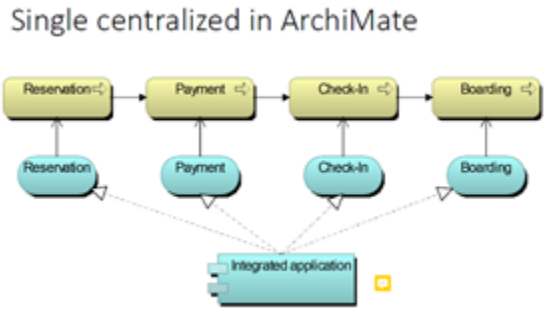


Figure 2-1: Example of business functions in relation to processes

Exercise 4.2

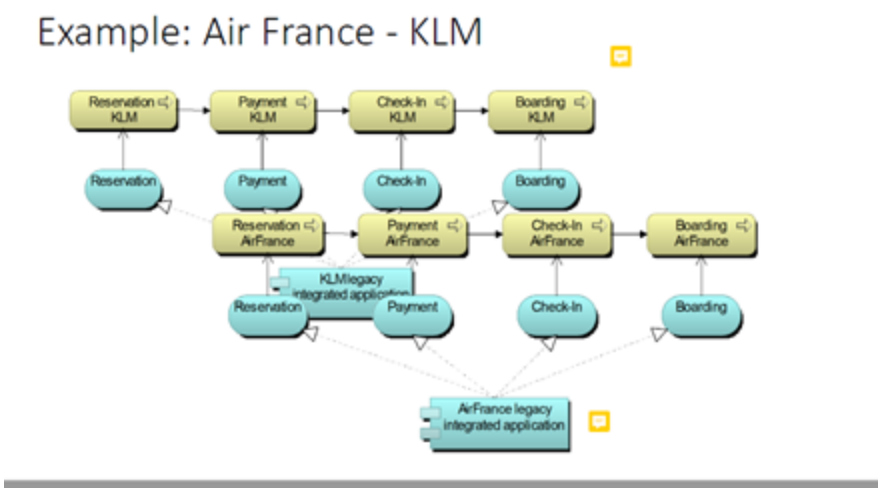
Show the difference between the “single centralized “ business process pattern and the “multiple centralized” business process pattern, proposed by Britton & Bye in section 12.4, by modelling the airline example form section 12.4 in ArchiMate. The models should include one or more business processes, application services, and application components. Can you also model the example with each of the business process patterns in BPMN?





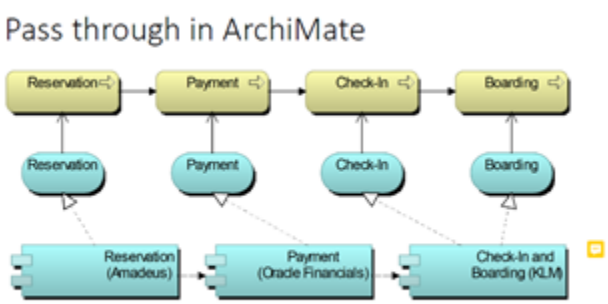
Exercise 4.3

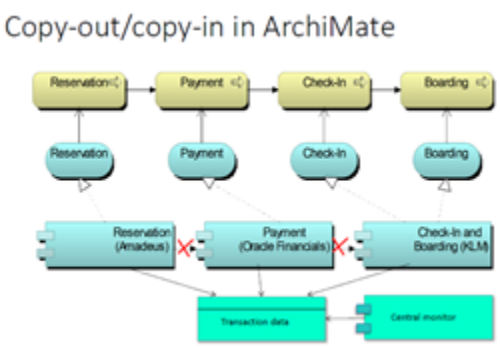
Consider the airline example from section 12.4. Assume that the airline company is Air France-KLM. Further assume that this company has two integrated applications for supporting all processes, one for Air France and one for KLM. Model this situation using the “multiple centralized” business process pattern with ArchiMate.



Exercise 4.4

Model the airline example with the “pass through“ business process pattern and the “copy out/copy in” business process pattern, proposed by Britton & Bye in section 12.4, in ArchiMate. The models should include one or more business Processes, application services, and application components. Can you also model the example with each of the business process patterns in BPMN?





Exercise 4.5

We all have gone to supermarkets to buy household things. Consider a supermarket scenario with the following start and end event:

- Start Event: A customer places the things s/he has bought on the conveyer belt at the cash counter.
- End Event: The customer exits the supermarket.

Identify actions, persons, roles, processes, triggers, intermediate events etc. between the start and end event. Using these draw a business process diagram also showing any IT device used. Make necessary assumptions.

Exercises: Non-functional requirements

Exercise 5.1 (Resiliency)

The use of backup servers is a simple and widely used technique for increasing resiliency. When the main server fails, the workload is taken over by the backup server.

1. Britton & Bye mention 4 steps involved in the recovery from failures using a configuration with a main server and a backup server. Enumerate the steps and provide a brief description of each.

   Recovery consists of four steps:

   1. Detect the failure-> having a heartbeat feature, whereby the backup system sends the primary system a message on a regular basis.

   2. Clean up work in progress-> switching to backup requires that all the data updated on the production system is copied to the backup system. Pg 239

   3. Activate the application on the backup system->once the database and any message queues are tidied up, the next task is to restart the complete running environment and put it into a state ready to accept more work.

   4. Reprocess “lost” messages-> usage of special recovery. On recovery the client sends a special message to interrogate the last input or the 1 before, was the last processed transaction.

2. Can you think of any demerits of the above configuration?

Exercise 5.2 (Resiliency)

The alternative to switching to backup after a failure is having the backup always active and waiting. This is called dual active, explained in section 7.2. There are two dual active approaches: (1) clustering, and (2) two-databases.

1. Explain (the difference between) the two approaches during normal operation (no failures occur).

Clustering: The database (which should be mirrored) is shared by both systems. Both systems can read and write to all the disks.Two-database approach: Each system has its own database.Each system has its own database; each transaction is processed twice, once on each database; and two-phase commit ensures the systems are kept in sync. If one of the systems is down, then the other system simply stops sending it sub transactions.

2. Which one of the two has more processor overhead? Why?

The clustering solution has potentially less processor overhead than the two-database solution.Clustering is less work for the applications developer (but more for the vendor). Pg 251

3. Which one of the two has more network overhead? Why?

The two-database solution has less networking overhead than the clustering solution. The two-database solution is better at handling new versions of software or the database schema.

Exercise 5.3 (Performance and scalability)

1. What is the difference between performance and scalability?

Scalability" is the ability of software to handle a sudden increase in workload. The increase could come from a surge in simultaneous users, a rising amount of data being processed, more individual requests for access, or anything else that placed higher demands on available resources.

Scaling can be achieved by "scaling up" or "scaling out".

Performance refers to the capability of a system to provide a certain response time. server a defined number of users or process a certain amount of data.

So performance is a software quality metric. Unlike what many people think it is not vage, but can be defined in numbers.

If we realize that our performance requirements change (e.g. we have to serve more users, we have to provide lower response times) or we cannot meet our performance goals, scalability comes into play. Scalability refers to the characteristic of a system to increase performance by adding additional resources

2. What is the difference between horizontal and vertical scalability?

Scaling up (or scaling vertically) means using better or higher capacity resources. For example, running an application on a machine with a faster CPU would be vertical scaling. Scaling up is incredibly simple; the same application can be run on the more powerful machine without changing it. However, this solution can't be applied infinitely since there's only so much more advanced the hardware can get within a reasonable time frame. It also gets more expensive as it scales into pricier equipment. Scaling out (otherwise known as horizontal scaling)

involves adding more resources to a system by adding more nodes within it. Running an application on multiple computers as opposed to one machine is a way to scale horizontally. Because the added

machines can be of the same type, scaling out carries a lower cost. It's easy to do without significant downtime. Without the proper application architecture, though, scaling out can lead to a drop in performance as the nodes struggle to communicate with each other.

Exercise 5.4 (Performance and scalability)

1. Briefly explain what is meant by load balancing?

Load balancing is the process of distributing network traffic across multiple servers. This ensures no single server bears too much demand. By spreading the work evenly, load balancing improves application responsiveness. It also increases availability of applications and websites for users.

Load balancer will distribute the work-load of your system to multiple individual systems, or groups of systems to to reduce the amount of load on an individual system, which in turn increases the reliability, efficiency and availability of your enterprise application or website.

2. "A two phase commit transaction entails significant network overhead and memory usage." Do you agree or disagree with this statement? Explain why.

A two-phase commit protocol is a blocking protocol. If the coordinator fails permanently, some participants will never resolve their transactions: After a participant has sent an agreement message to the coordinator, it will block until a commit or rollback is received

blockage problem". 2PC occurs after a transaction is processed, and thus necessarily increases the latency of the transaction by an amount equal to the time it takes to run the protocol. These blocked transactions in turn block other transactions from running, and so on, until 2PC completes and all of the blocked transactions can resume. This blockage further increases the average transaction latency and also decreases transactional throughput

3. "Remote database access technology is inherently inefficient for real-time transactions." Explain why this statement is true? Mention a common workaround approach for using a remote database in real-time processing.

Remote Database Access provides standard protocols for establishing a remote connection between a database client and a database server. The client is acting on behalf of an application program while the server is interfacing to a process that controls data transfers to and from a database.

It has poor performance since network traffic is greater than other solutions, especially if SQL commands must be parsed by the database server.

• Some remote database access technology does not support two-phase commits, so updates cannot be synchronized.

• The structure of the shared database is hard coded into many dispersed applications, making schema changes difficult to manage and implement. Put another way, the SQL commands that access the shared data are scattered everywhere. Page 80

Exercise 5.5 (Security)

1. Typical layers which should be considered when designing a SQL server security schema based on the onion model architecture are: encryption layer, authentication layer, SQL server permission layer, data layer, and stored procedures layer. What are the relative positions of the mentioned layers in the union model? Provide a brief explanation of your choice. Page 348
2. Why is it difficult for organizations to move from a boiling lava security model to an onion model? Page 353
3. What is meant by the access control layer in the security paradigm? What is its function? Explain with a figure.

Exercise 5.6 (Security)

Consider the use case that you have chosen and look at the data that is used in the organization.

1. What are the major data elements, and what categories of persons need to see them?
2. Which access roles can you identify with these data elements? Note that access roles are not the same as business roles. You could characterize a role by stating: (1) what read or write access is required?, (2) are special privileges applicable to a subset of the data?, (3) can this role be used anywhere or is it location dependent?
3. Which applications belong in which security context (or is there only one security context in your organization)?
4. Where is the authorization going to take place?
5. Where is the access control going to take place?
6. Where is the security management going to take place?
7. What threats exist?

Exercise 5.7 (Information access and information security)

Consider a logistics company, Logistics B.V. It takes shipment orders from customers and uses its own infrastructure of trucks and warehouses to deliver goods according to these orders. It is expanding to new markets, which entails a change in its current business processes.

Current Situation. There is an online order entry system for allowing orders to be sent to Logistics B.V. by filling an online form. Not all fields of the form are filled by every customer. Moreover, customers frequently have special constraints on orders which are not yet in the form. Customers often (also) call or email when they want to place a new order. As a consequence, Logistics B.V. employees sometimes have to enter orders manually. All current orders are saved in a database. An experienced team of planners is responsible for planning optimal routes for each order (or batch of orders). Using their knowledge, contacts and intuition they can meet most order deadlines. Sometimes, there are disruptions (road blockages, heavy snow, traffic jams) which can delay an order. These are registered manually in a different database. A disruption can cause a re-planning in the order schedule, depending upon which order(s) are affected by a particular disruption. Logistics B.V. also has a subscription to 5 information providers which provide regular updates on the traffic situation (3 providers) and the weather (2 providers). The employees of Logistics B.V. work on their systems (i.e., computer/laptop) in the office. If a customer wants to know about the status of his/her orders, s/he can call the office of Logistics B.V. and by using the assigned order number s/he can get information about the corresponding order.

Desired Situation. Logistics B.V. wants to implement a more desired situation as follows:

Logistics B.V. bought a route planning software tool, PlanRoute. Using a robust routing algorithm, optimal routes can be planned from source to destination location for goods transportation. PlanRoute is run on alternate days, so it considers execution of orders on the same day and on the next day. It also has the feature to include contextual information (like weather forecasts, traffic forecasts, road construction plans) in route planning.

Also, to improve resiliency, a controlled duplication mechanism for data storage is proposed.

Logistic B.V. wants to support the customers in viewing the status of their orders on any device (laptop/mobile/tablet), which is expected to reduce the number of calls to the company.

Answer the following questions (a-c) based on the above case. You can make assumptions wherever required (if you do, please mention them explicitly).

1. What are, according to Britton and Bye (Chapter 15), the different techniques which exist for presenting query responses from existing applications to a new display medium? Which of them do you think is best suited in this case (for order status response messages to customers) and why?
2. What problems can Logistic B.V. expect in the implementation of PlanRoute? How can they be solved?
3. In controlled duplication, the same data is stored in two or more locations. Explain with a simple example (related to the case above) how the controlled duplication can lead to two divergent data copies ? Assume that the connection between the two databases is intact and working.

Service-oriented enterprise

Exercise 6.1

1. According to Nardi et al. (2016), ArchiMate has three limitations when it comes to modelling social relations connected to services at the business level. What are these limitations?
   Derek's

   1. Can't distinguish between the actors-customer (if they are target customers or actual service customers)
   2. Among which service participants is the "special cable Tv Contract" established.
   3. If the contract represents a specific service agreement between a hired service provider and a particular customer, or is it a type of service contract that applies to any target customer.
2. These limitations are addressed by proposing patterns for ArchiMate that use the existing model elements, but introduce relations that are not in the ArchiMate metamodel. The patterns are referred to by the names: service offering type, service offering, and service agreement. How are the limitations resolved by these patterns?

   Derek's

   The proposed modeling patterns use the existing service, product and contract modeling elements, as well as the association relationship. Each pattern is composed by four groups of elements: (i) a product and related services, (ii) the roles/actors that provide the product/service, (iii) the roles/actors that consume the product/service, and (iv) the respective contracts. The contracts are in the center of each modeling pattern, representing service commitments and claims as fundamental elements in service relations. The associations in which a contract is involved establish the semantics of each pattern: in a service offering type, the contract connects the provider role with the service customer role; in a service offering, the contract connects the service provider actor with the customer role; and (iii) in a service agreement, the contract connects the (hired) provider actor with a particular customer actor.

   Page 4 thesis

Exercise 6.2

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

1. Model (an example of) an IaaS application involving two companies.
2. Model (an example of) an PaaS application involving two companies.
3. Model (an example of) an SaaS application involving two companies.
4. Can the proposed patterns also be used to model the application of a BPaaS (Business Process as a Service) strategy?
5. Can you think of any useful applications of the proposed patterns in your use case? For example, consider settings in which services are offered by external/internal actors to/in the company according to a XaaS strategy.

Exercise 6.3

Consider a company Weather Analytics Inc. that provides analytics services in areas concerned with weather forecasting, environmental monitoring, and disaster management. It originally started as an institute that specialized in making weather forecasts in Switzerland, but it gradually broadened its application domain and expanded services to many countries. Until 2005 forecasts were based on historical data and complex computational models. Since then it also offers real-time analytics based on data capturing from a variety of sensor-based data sources. For this purpose it developed its own big data storage and analytics facility. Since 2010 this facility is both externalized as a service to external customers and contracted out internally as a service to its own departments. For external customers four types of contracts are distinguished: normal contract, high volume contract, high speed contract, and plus contract, where the latter three are specializations of the first. For internal customers there is a separate internal contract. The service offered includes three ancillary services: interactive helpdesk (telephone), self-service helpdesk (website with answers to frequently asked questions and an online form for questions), and a virtual device manager (application service to manage devices).

1. Model the service externalization and insourcing using the ArchiMate patterns proposed in Nardi et al. Use Enterprise Studio to represent your solution in ArchiMate.
2. What cloud service models are used (compare to tables 2 and 3 in Nardi et al.)?