university of groningen

Faculty of Mathematics and Natural Sciences
Department of Computing Science

Software Patterns - First Assignment - Team 1

PASSLOCK

Pattern-Based Design

Authors:
Thomas Hoeksema
Maarten Kollenstart
Toon Albers
Michel Medema
Jasper Mohlmann

Coach:
Jorrit Idsardi

Lecturer:
prof. dr. ir. Paris Avgeriou

Version 0.3
November 29, 2015

Revision History

The following students have contributed to the elaboration of this document:

ID	Author	Student no.	Email address
TH	Thomas Hoeksema	S2349639	T.M.Hoeksema@student.rug.nl
MK	Maarten Kollenstart	S2209853	M.Kollenstart@student.rug.nl
TA	Toon Albers	S2768372	T.Albers.2@student.rug.nl
MM	Michel Medema	S2396009	M.Medema@student.rug.nl
JM	Jasper Mohlmann	S1988794	J.C.Mohlmann@student.rug.nl

Table 1: Information on authors

The table below lists all versions and revision details of this document. Changes since the last version are marked with a red sidebar in the document itself. All versions and revisions have been reviewed by our coach with the exception of version 1.0, as that is the final deliverable.

Ver.	ID	Date	Revision
0.1	TH	11/11/15	Initial template of document.
	JM	14/11/15	First idea description.
0.2	TH	19/11/15	Minor adjustments to document structure
		21/11/15	Writing sections 1-2
		22/11/15	Refining and revising parts in sections 1-4
	MK	20/11/15	Creating first three architectural decisions
	TA	21/11/15	Writing sections 3.4 and 3.5
		22/11/15	Writing section 3.6, bibliography, refining sections 1 and 3.5
	MM	21/11/15	Analysis section
		22/11/15	Analysis section
		22/11/15	Reviewing sections 1-4
	JM	19/11/15	Writing stakeholder and key-drivers sections
		22/11/15	Writing non-functional requirements
0.3	TH	25/11/15	Refining section 2, refining sections 3-4 and Appendix, document structure
		28/11/15	Refining sections 2-3, commented draft of section 9
		29/11/15	Refining sections 2-3, reviewing and refining sections 4-7, small draft of section 9
	MK	28/11/15	Writing section 5
		29/11/15	Writing sections 7.2.2 & 7.3
	TA	24/11/15	Refining sections 3.4 - 3.6
		25/11/15	Writing A1 use-cases, refining section 3.6
		26/11/15	Writing A2 risk assessment
		28/11/15	Refining sections 3.6 and 4.1
		29/11/15	Writing section 7 software architecture
	MM	25/11/15	Analysis, System Architecture
		27/11/15	System Architecture

		29/11/15	Analysis, Software Architecture
		29/11/15	Reviewing sections 4-6
	JM	24/11/15	Corrections on SH and key-drivers
		25/11/15	Writing A1 use-cases
		28/11/15	Refining A2 risk assessment
		29/11/15	Writing Hardware Architecture, Software Architecture
0.4	TH	30/11/15	
		01/12/15	
		02/12/15	
		03/12/15	
		04/12/15	
		05/12/15	
		06/12/15	
	MK	00/00/15	
		00/00/15	
	TA	00/00/15	
		00/00/15	
	MM	00/00/15	
		00/00/15	
	JM	00/00/15	
		00/00/15	

Table 2: Detailed revision history

Contents

List of Figures

List of Tables

Glossary

ERP - Enterprise resource planning, business-management software used to support business processes

HVAC - Heating, ventilation and air conditioning

PASSLOCK - The system which is the subject of this assignment, which provides networked access control and flexible office scheduling (and the enforcement thereof) for buildings of enterprises and governmental institutes.

PDAP - Acronym for the Pattern-Driven Architectural Partitioning approach. [1]

QR-code - Acronym for Quick Responsive Code, a machine-readable optical label which is a trademark for a type of matrix barcode, popular for fast readability and decent storage capacity.

VPN - Acronym for Virtual Private Network, a technique used to extend a private network over a public network.

1 Introduction

This document is part of an assignment for the course Software Patterns at the University of Groningen. In this assignment, we take the role of a software architecture team that is designing a system that provides access control for entire office buildings. This new product tries to simplify the process of flexible office space planning as well as providing simple and advanced access control pertaining to individuals within a building.

The goal of this assignment is to describe the architecture of this system. The design of this system should be documented according to the Pattern-Driven Architectural Partitioning (PDAP) methodology. [1] This section of the document serves as an introduction to the product by describing the system context in which the system-to-be is relevant. The scope and core goals of the system will be described, after which we outline the structure of this document.

1.1 System Context

Enterprises and governmental institutes often impose rules upon the mobility of employees within the buildings of these organisations, where certain areas within the building can only be accessed by an employee when they have been given sufficient clearance to do so. The reason why companies and institutes adopt such measures is usually related to security, to protect sensitive data or expensive property within the building or to restrict access to dangerous environments. However, there is another use case for access control apart from security, namely in the field of flexible offices. In this field, an office building is managed by so-called facility management companies, who subsequently rent individual offices or entire floors of these buildings to external companies or individuals. The offices can in theory be rented on a day-to-day basis, so there is a large amount of administration involved if access control is performed and checked manually by the facility management staff. With the current advent of flexible offices to be implemented in more locations, [8] this will surely become a bigger issue in the future as well.

The scope of the system is limited to buildings of enterprises and governments which have adopted or are planning to adopt some form of access control for rooms or areas within those buildings, and also includes the application of flexible offices. Furthermore, we include within the scope that the system should only affect areas within this building that have clear connections with other areas and thus can be easily secured without a large investment or structural changes to the building layout.

1.2 The PASSLOCK System

The new product of our company, named PASSLOCK, is a networked locking and access control solution for enterprises, governments and facility management staff of flexible offices. The system consists of a large array of locks to be placed on doors or other type of access points on the outside and within the buildings, and secure web/remote interfaces for building management staff as well as employees and/or tenants. For each employee or tenant, the building management staff can manage the list of areas that can be accessed by the respective individual, and can add information to the system that can help identify the individual (such as adding biometric data or printing a key-card).

Subsequently, if the person is equipped with a method of identification, they can enter the building and areas within, by interacting with the locks that are placed. By providing a method of identification that is appropriate for the type of lock that is used, the lock will contact a remote system to verify the identity of the person and that they may access the corresponding area, and subsequently can unlock the door for that person.

What is particularly interesting about this system is its use for flexible offices. Namely, tenants can register themselves at a web interface and reserve offices through this web interface. This scheduling information is taken into account when a lock is interacted with, so that we can give access to individuals who hired an office space if he is trying to access that respective office space within his time slot. If the tenant only has to show a code (obtained while paying for their rented office space) to accomplish this, then the tenant does not have to interact with the building management staff at all, which saves administration time and costs, since the scheduling is enforced automatically. We should note that building management will always be necessary, since tenants should be enforced to exit their office spaces when their time slot has expired, since locks cannot enforce this.

1.3 Overview

The structure of the rest of this document will now be described.

- Section 2 (*Business Information*) describes the system in the context of its business environment, which includes the business vision and rationale, and a description of the product, its audience, and its key features.
- Section 3 (*Requirements*) identifies the stakeholders of the system, its key drivers, and describes a set of requirements and use cases that the architecture should fulfil.
- Section 4 (*Analysis*) contains the analysis of the system, including the design decisions that were taken and the software patterns that are used within the architecture. Furthermore, alternatives for these decisions and patterns will be given and the choices will be duly motivated.
- Section 5 (*System Architecture*) outlines the high level, reverse engineered description of the architecture of the hardware platform.
- Section 6 (*Hardware Architecture*) describes the architecture on the level of hardware.
- Section 7 (*Software Architecture*) describes the architecture of the software that is involved with the system.
- Section 8 (*Architecture Evaluation*) focuses on evaluating the architecture that has been designed and verifies if the requirements of the system have been met.
- Section 9 (*System Evolution*) will entail possible points of improvements to the system and defines how the system should evolve through its life cycle.
- Section 10 (*References and Acknowledgements*) and *Appendices* conclude the report with a list of all knowledge bases and resources that have been utilised during the elaboration of this assignment and acknowledges the efforts of external individuals and/or groups.

2 Business Information

This section elaborates on the business environment of the PASSLOCK system. First, the business vision gives a description of the envisaged business opportunities and unique selling points. Advantages and disadvantages with respect to the current situation through a SWOT-analysis will be given. Subsequently, the main product functions and their evolution over time are discussed. These are followed by an overview of the target audience, and a financial model is provided that covers the required investments and operational costs, and finally, this section finishes with an identification of potential competitors.

2.1 Business Vision

Firstly, we will describe the core goals that the PASSLOCK system needs to accomplish in order to be a successful product on the access control market, which are the following:

- **Networked access control:** The system consists of an array of locks that can be placed between clearly defined zones within/between buildings (such as rooms) through physical access points (such as doorways). Each of these locks is connected to a central, secure server where access data for individuals and data on the clearly defined zones is stored, and where the logic for access control is performed. This way, the access control structure of a building is determined from a single location in the network.

- **Remote management:** The administration staff of a building can remotely connect to the secure server over a secure connection to add/remove persons to the system, to add/remove methods of identification for a specific person in the system, or to change the logic with respect to the zones that are controlled by the locks.

- **Remote scheduling:** Rooms can be marked as flexible offices so that they may be rented by employees or tenants. There is a web service which is accessible by employees or tenants who have an associated account (i.e. which are registered in the secure system explained before). Individuals who are logged in on the web service can rent rooms which are available at certain time slots for themselves (and may have to pay for this), after which the system recognises that this individual should be let into that room on that specific time slot.

- **Monitor rooms in use:** Due to the fact that the central server processes and distributes all of the incoming data and acknowledgements sent to locks, a log can be kept of this information. This log may be viewed to gain information on whether there is currently one or more individuals in a specific room or zone within the system.

The extent to which the PASSLOCK system is able to excel in these primary goals and will be successful is partially dependent on internal factors such as strengths and weaknesses. Moreover, external factors, i.e. opportunities and threats, play a significant role in determining the success of the system. For uncovering this business related information, we utilise a so-called SWOT analysis (see [6] and [7]), which is a structural planning tool and methodology for the identification of the Strengths, Weaknesses, Opportunities and Threats of a product from a business perspective. The results of this table are depicted in Table 3, where the four categories of items are listed.

	Helpful (to achieve the objective)	Harmful (to achieve the objective)
Internal origin (product/company attributes)	**Strengths:** In-house experience with access control techniques such as key-code, key-card, NFC. Experience with setting up secure local networks on which most of the system will operate.	**Weaknesses:** Currently no influence on the market. No in-house experience with biometric access control techniques. No in-house production of sensors, so there is a dependency on external sources for the supply of hardware.
External origin (environment/market attributes)	**Opportunities:** High demand for variable/networked access control systems. Limited competition in access control for flexible office space. Existing systems offer a limited amount of (often only one) types of access control techniques or are not networked (data stored in on-device memory). No vendor lock-in due to high variety of stand-alone sensors being available.	**Threats:** Some access control techniques may be only available with decent quality in proprietary systems. Existing competition for traditional access control systems is very high and they may adjust their products to be networked/variable.

Table 3: SWOT-analysis

The unique selling points of the PASSLOCK system can be summarised as:

- **Highly adaptable access control:** The system has a high compatibility with various access control methods, such as keypads, key-card readers, RFID tag scanning or NFC communication through mobile phones, and also fingerprint and iris recognition as biometric access control techniques. Due to a flexible design of the locks, it is very easy to integrate existing physical locks (cylinders, turnstiles, etc) with any type(s) of access control sensors, so virtually any combination of physical lock and sensor(s) is possible. Furthermore, the access control can be maintained on an individual as well as room basis through an administrative secured web interface.

- **Flexible scheduling management:** Customers and tenants can view information on the building through a web interface and can rent or make a claim on office space. The information about rented or claimed office space is automatically available and enforced within the PASSLOCK access control system, which means that administrative staff has to spend less time on interaction with tenants and employees about office space, and that less administration has to be performed by hand. The administrative staff can designate which rooms are available for hire or claim and which are not, and can detail the available timeslots for these rooms.

- **Highly secure and available:** The locks communicate over a secure connection with a clustered server that stores all of the data regarding personnel/tenants, scheduling and room information. It is only possible to add or remove specific identification methods for specific individuals in the system, so it is not possible to obtain the data of an identification method (e.g. fingerprints, passwords, keypad codes). The secure server cluster and the servers that serve the web interface to employees/tenants is replicated and load balanced, so that we can offer high reliability and performance of the services that these servers provide.

- **Modelling room occupancy:** Due to the fact that the locking/unlocking of physical access points has to be verified by the secure cluster, information on the dates and times of when these access points were locked/unlocked can be logged. Subsequently, this logged information can be accessed through the admin interface, so that the system can determine semantically, based on these actions in the log, which rooms are currently occupied by one or more persons and which are not. Furthermore, this information can be provided through an interface to other systems to improve automatic lighting/HVAC functionality, by only allowing these to be turned on if the room is occupied.

2.2 Business Rationale

The mission statement of our company is to create distributed systems that will provide basic and advanced access control for buildings, office space in particular. We strive to design a system that is useful for large institutes such as enterprises and large government facilities and will scale well in those types of environments, but we also want our system to be useful for businesses who comprise of multiple smaller buildings such as local branch offices for banks or police stations, who also require proper access control for their facilities within and between these buildings.

The company is involved in the management of the hardware and software components that make up the PASSLOCK system. However, the company itself does not produce the hardware that is necessary for the system to function, but utilises hardware with both low-level and high-level interfaces to accomplish its task.

2.3 Target Audience

The target audience of the PASSLOCK system can be broadly divided in two groups, namely enterprises and governments who are in the possession of office buildings that require access control, and the management staff of flexible office spaces who want to enforce the scheduling of rented office space automatically through access control techniques. These groups of target customers will now be described separately.

First of all, the system provides access control for entryways in a large variety of buildings. Currently, most access control systems that are implemented in buildings work with static databases that are present in the locks themselves. When someone scans their key-card, code or some form of biometric data, the data of the request is compared with the data that is available on the database of the lock. When new users have to be added or old users have to be removed, the maintaining staff will have to go to the lock in person to modify this data, either by uploading it to the lock or by swiping a master key-card past

the lock which performs a set of instructions. Not only is the latter a threat to security (what if someone can reproduce such a master key-card?), but this also complicates the process of modifying access rights. Therefore, companies and governmental institutes that currently utilise such a system, or are inclined to purchasing one in the near future, will be interested in the system that we will offer. Our system provides a wide variety of locks and access points, is able to be managed via a remote interface, from which all access rights can be adjusted, and provides easy means for monitoring presence within the building through the data obtained from locks. Furthermore, the scheduling system could be implemented for enterprises and governmental institutes to incorporate the concept of flexible offices within these buildings, but then restricted to the employees of that company or governmental institute.

Secondly, our target audience also includes the management staff of buildings which implement the concept of flexible offices for the general public. In these buildings, tenants can rent office space or entire floors of office space for a certain amount of time, from which the building management profits. The building management is then responsible for the maintenance of the building. Our product is appealing for this type of customer audience, because it simplifies the task of scheduling, payment and access control for tenants to certain areas within the building. Tenants can register themselves online, rent office space online, and then use a code that will be provided to them to access the corresponding areas within the physical building when they arrive there. Little to no interaction is needed with the management staff, with the exception of when issues occur due to tenants violating their planned schedules or when a lock may not be operating correctly. This could save the management team of those buildings a large fraction of their work which consist of administration tasks.

2.4 Financial Model

(May be included in final deliverable)

2.4.1 Competitors

We will now elaborate on some of the most prominent competitors in the field of (distributed) access control and identify their key characteristics, and relate these to the PASSLOCK system to describe their similarities and differences.

2.4.2 Honeywell

The first and foremost of the competition is the Honeywell Managed Access Control system by HoneyWell. [9] Honeywell is a company that provides products in the markets of managed and unmanaged access control, intrusion detection, monitoring and various alarms system such as fire alarms, both for residential and commercial use. Their products are suitable for large enterprise office buildings, but products for smaller offices or personal use are also offered. The Honeywell Managed Access Control system consists of several sensors that scan badges or key-cards using NFC technology. The information that is needed to identify individuals is stored on the sensors themselves, and can be modified through a remote interface. The remote interface also offers the possibility to generate reports about access attempts and the system provides video surveillance as well.

Relating this system to the PASSLOCK system, we can see that the Honeywell system only supports one kind of access control, namely NFC badges and key-cards. Furthermore, the information that is needed to identify people is stored on the sensors themselves rather than a remote verification server, which can be considered as insecure. Honeywell's system is also limited to 2000 users for this reason.

2.4.3 Identicard

Another big competitor of the PASSLOCK system is the Identicard Access Control system by Identicard. [10] Identicard offers a similar system as Honeywell, but offers a wider variety of sensors and the solutions that are offered can be customised based on user needs. While this competitor has a lower market influence than Honeywell, we could consider this competitor to be more dangerous to our product than Honeywell, since Identicard offers very similar services to us.

Identicard also offers a mobile application that allows employees to unlock certain doors in the building for which they are authorised through pressing buttons in the application. The system is also able to be integrated with surveillance systems. However, Identicard does not offer any functionality which would be suitable for flexible offices, and similar to Honeywell the data which is needed for identification is replicated on the lock mechanisms themselves, which causes a limit to the amount of users for the entire system.

2.4.4 Matrix Systems

The Frontier Access Control system is developed by Matrix Systems. [11] This system provides access control through badges and photo ID cards. It also includes a wider variety of possible physical access points, such as elevators. Furthermore, this competitors provides the best biometric locks out of all competitors listed in this section, but the variety of sensors is still restricted to NFC and fingerprint readers.

2.4.5 Tyco Integrated Security

Next, we will discuss Security Access Control, a system developed by Tyco Integrated Security or TycoIS. [12] TycoIS provides physical, electric and managed access control through video surveillance, key-card scanners, fingerprint scanners and more. The software does however not grant administrative staff the possibility to generate logs of access attempts or to see access attempts for specific users. Similarly, the interface through which the access requirements of locks may be adjusted is very limited.

2.4.6 Protection 1

Finally, we mention the Protection 1 Access Control product by the Protection 1 company. [13] This system offers managed access control similar to the type that TycoIS provides, and also supports online video surveillance and a live chat with the supporting staff that manages the surveillance of the building. The sensors are limited to card readers and sensors, and the system does not support any kind of biometric sensors.

3 Requirements

This section describes the requirements of the PASSLOCK system and identifies the stakeholders and their concerns. The first paragraph details the architectural vision of the PASSLOCK system. Subsequently, we will define the concerns that play a role for stakeholders of the PASSLOCK system, and we will also identify all primary and secondary stakeholders of the system, and denote their concerns. The section then proceeds with a list of high-level requirements, followed by a list of use-cases, and is concluded with a description of the functional and non-functional requirements of the PASSLOCK system.

3.1 Stakeholders

The stakeholders of the system, as well as their corresponding concerns with respective to the PASSLOCK system, will now be identified. We will describe the concerns of the stakeholders using quality attributes as defined in the ISO 25010 [2] standard. The quality metrics that will be used are displayed in Table 4.

Standard	**Quality attributes**
ISO 25010	Usability, Security, Reliability, Performance (Efficiency), Compatibility, Maintainability, Portability

Table 4: Used definitions of quality attributes

Furthermore, we define the following quality attributes:

Profitability The ability for the manufacturer, distributor, or licensing authority of a product to create a profit through the sales of a product and the various maintenance activities for that product thereafter.

Affordability The ability for stakeholders to gain a net profit or other kind of significant advantage that weighs up against the installation and maintenance costs of the system.

Now that the appropriate quality attributes to describe the stakeholder concerns have been identified, we will give a description of all stakeholder types for the PASSLOCK system. A short description of their roles are given, as well as an overview of their concerns relative to the quality attributes.

3.1.1 Primary Stakeholders

End user (SH-1, high) The end users are the people that use the system to gain access to certain rooms of areas and to schedule rooms for meetings. This stakeholder consists of flex workers from start-ups, freelancers or just people on a journey that need a place to work for a few hours. Employees of a company where there is a high need for security, for example secret services or intelligence agencies, can also be end users of the system. They are using the system for access control to specific areas or rooms. Their main concerns are:

- **Usability:** The end users will use the system very often so operability should be high in order to avoid irritations.

- **Security:** End users expect that no one other than the authorised persons can access their private rooms.
- **Reliability:** End users should always be able to enter areas using their authentication method.
- **Performance efficiency:** End users should not have to wait an excessive amount of time before they are granted access.

Building management (SH-2, high) The building management is the stakeholder responsible for the daily management of a building or a group of buildings. They can authenticate new users on the system and update their security credentials such as iris scans or fingerprints. The systems should be available to the management at all time so that the management knows how many people there are in the building and where they are. We also expect that some customers use temporary employees for their concierge services so the system should be usable with minor knowledge of the system. Their main concerns are:

- **Usability:** The system should be easy to operate for multiple janitors, also the planning of available rooms should be an self-explaining process
- **Security:** Authorisation changes may not be made by people other than the building management.
- **Reliability:** Reliability is needed for good control of the building using HVAC and people management inside.
- **Compatibility:** The system should be interoperable with HVAC systems to reduce workload.

Customer (SH-3, medium) The customer wants a system that handles all access control within a building an also the the scheduling of rooms. The financial saving of implementing the system should outweigh the cost of the system in order to make it product of interest. As they give the full control of physical security to the system it speaks for itself that this is a major concern. Integration with other systems used in the companies could provide extra savings. Their main concerns are:

- **Affordability:** A high price can reduce the interest in the product and make competitors more attractive.
- **Security:** The customers relies on the system for access control but also logging to take effective countermeasures in the case of a breach.
- **Reliability:** A unreliable system is not interesting and can make the customer decide to switch to another supplier.
- **Compatibility:** Integration with other systems like ERP and HVAC can reduce costs and workload and so save money.

Product owner (SH-4, medium) The product owner provides the budget and invest in the development of the system. They want their investment back so profitability is important. The system should also be highly adaptable to different situations in order to create a large reachable consumer market. Development costs should be minimized in order to create a higher profit. Their main concerns are:

- **Profitability:** The system should generate revenue for the product owner in order to make profit.
- **Portability:** High portability makes it easier to deploy the system on different hardware and software and so create a bigger consuming market.
- **Maintainability:** The owner profits from an easy maintainable system because is decreases development costs.

Development team (SH-7, medium) This stakeholders is responsible for all development of the software. It consist of the software architects, the programmers and testers. In order to work as efficiently as possible the software should use modular components, reusable code and have great testability. Their main concern is:

- **Maintainability:** The system should be testable, modular and reusable in order to decrease work during development now as well in future version of the system.
- **Reliability:** Higher fault tolerance should reduce workload on developers.
- **Portability:** The development team benefits from a high portable system. Features should supporting coupling with other systems should be easier to implement.

Maintenance team (SH-5, low) The maintenance team is responsible for the system on site. It is tasked with the replacement of broken hardware, simple software problems and other problems that are not the fault of the producer. They also make small changes to the system when requested by the end users or the building management. Their main concerns are:

- **Maintainability:** Testability and modifiablity can increase the ease of fixing bugs and making adjustments to the system.
- **Reliability:** A reliable system reduces the needs for maintenance and dealing with downtime.
- **Portability:** Hardware parts should be easy replaceable in case of a defect. Installation should also not require specialised trained personnel.

3.1.2 External Stakeholders

Emergency services (SH-6, medium) The emergency services can use the system to gain information about users in the building. This can help during emergencies like fire, bomb threats and medical emergencies. Possible locations of end users can be retrieved from the system by the emergency services. Their main concerns are:

- **Performance efficiency:** During incidents time is a critical factor. The system should respond almost instantaneously on request.
- **Reliability:** Information about locations should be available and correct. Incorrect data could lead to dangerous situation for emergency personnel.
- **Portability:** Coupling with the systems of the emergency services can make the response time quicker.

External providers/suppliers (SH-8, low) This stakeholder is responsible for providing hardware for the system. This includes various locks, iris and fingerprints scanners, wires, servers and key cards. The are interested in supplying a wide variety of products so that the system can be implemented in different situations with different needs. Also to prevent the demand for long support of hardware a portable system is desired that does not depend on specific components.Their main concern is:

- **Portability:** Suppliers benefit from a portable system so that the system can be made to function using their hardware.

In Table 5 a stakeholder vs key-driver matrix is shown. This matrix functions as a tool to find the key-drivers of the system. Each stakeholder is given a certain number of points which can be freely spent on their concerns. The 3 quality attributes with the most points are the key-drives for this system. The points given to a stakeholder depends on its importance to the project. Stakeholders of high importance are given 100 points, medium 75 points and low 50 points.

		Concerns								
Stakeholders	**Points**	**Usability**	**Security**	**Reliability**	Performance	Compatibility	Maintainability	Affordability	Profitability	Portability
End user	100	35	20	30	15					
Building management	100	35	35	15		15				
Customer	75		20	15		10		30		
Product owner	75						15		35	25
Development team	50			10			25			15
Maintenance team	50			10			20			20
Total:	400	**70**	**75**	**80**	15	25	60	30	35	60

Table 5: Table containing point distribution and stakeholder concern matrix.

3.2 Key Drivers

Table 5 shows the 3 most important quality attributes. Those are briefly described below.

Security:
Security is the most important quality attribute in the PASSLOCK system. End users rely on the system to keep unauthorised people outside their rooms. Flex workers who use an office for multiple days may store valuable items or sensitive information inside their rented room that must be protected from intruders. Therefore the system should be built so security is guaranteed. User components, admin components and authentication should be strictly separated in the software so unauthorised use is prevented. Hardware used for storing sensitive data like fingerprints and iris-scans should be located in such a way that data theft is virtually impossible.

Usability: As there are many different users the system should be usable to all. Flexworkers and freelancers may only use the system once for a room so it should be self explanatory. As the building managers vary from temporary workers to experienced security staff, this part should also be easy to use.

Reliability: Security can only be assured when the system is up and running. Reliability of the system is therefore a key-driver. Failure could lead to locked in people and may cause emergency situations.

3.3 High-Level Requirements

The following requirements describe the high-level functionality of the system. They will be further refined through functional and non-functional requirements.

HL-1	**Must**	**Allow registration of new users**
The system must allow registration of new users by building management and optionally through a website.		

HL-2	**Must**	**Allow access to authorized users**
The system must allow access to areas that the user is authorized to enter whilst denying access to users that are not authorized.		

HL-3	**Must**	**Support multiple authentication methods**
The system must support multiple methods of authentication to verify the identity of users.		

HL-4	**Must**	**Allow booking of rooms on schedule**
The system must allow users to book rooms which have been made available depending on their authorization.		

HL-5	**Must**	**Show estimated occupancy of rooms**
The system must be able to show the estimated current occupancy of rooms based on the access- and locking history.		

HL-6	**Must**	**Show history of access**
The system must be able to show historical access data regarding rooms accessed by users and presence of users in rooms		

HL-7	**Must**	**Allow fail-safe exit**
The system should allow the users to exit the room (and by extension, the building) in the event of a failure of the system.		

HL-8	**Optional**	**Automated HVAC control**
The system should be able to control the HVAC of a room according to the presence and preference of its users.		

3.4 Use-cases

Figure 1 describes the system in terms of use cases. They are slightly elaborated in Table 6. The full description of these use-cases is provided in Appendix A1.

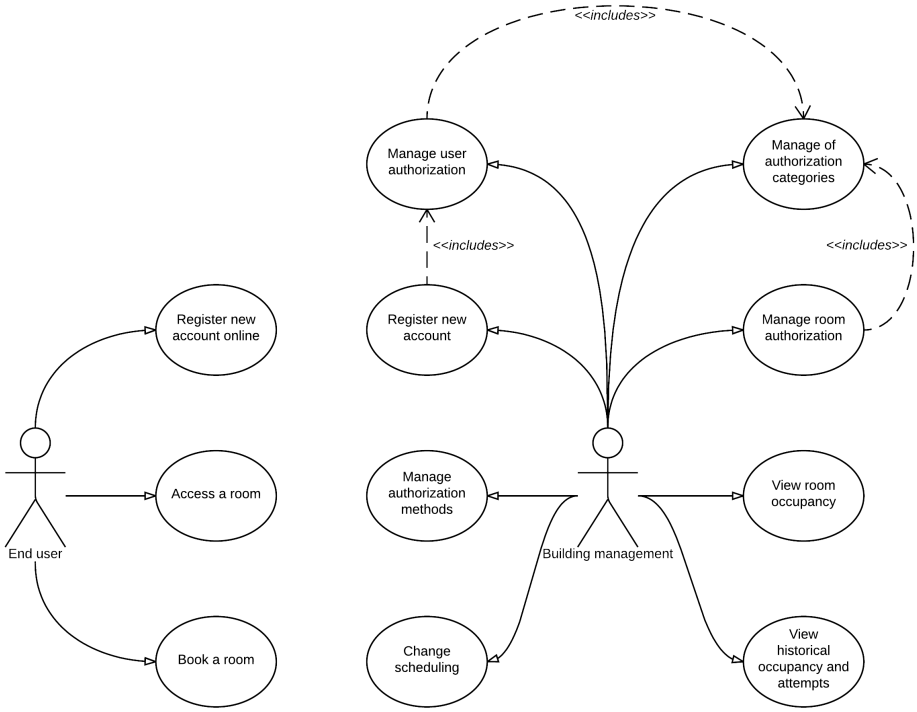

Figure 1: Use-case diagram

ID	Actor	Name	Description
UC-1.1	User	Register a new account online	Register a new unprivileged account on the website.
UC-1.2	Management	Register a new account	Register a new account on the system.
UC-1.3	Management	Manage user authorization	Manage the authorization categories for a user.
UC-2.1	User	Access a room	Access a secured room by providing authentication.
UC-2.2	Management	Manage room authorization	Add whitelisted and blacklisted categories to define authorized groups of users and set locking behaviour.
UC-2.3	Management	Manage authorization categories	Add or remove categories for access permission.
UC-3.1	Management	Manage authentication methods	Manage the forms of authentication for a user (such as fingerprints or passcodes).
UC-4.1	User	Book a room	Book a room available for scheduling for temporary access.
UC-4.2	Management	Change scheduling	Change room bookings and availability.
UC-5.1	Management	View room occupancy	View the estimated occupancy of rooms.
UC-6.1	Management	View historical occupancy	View the history of room occupancy and access (attempts) of rooms by users.

Table 6: A brief overview of the high-level use-cases.

3.5 Functional Requirements

In this section the functional requirements for the PASSLOCK system are presented. The requirements are derived from the high-level requirements. Their first number refers to the corresponding number of the high-level requirement.

FR-1.1	**Must**	The system shall allow new users to register on a website.
FR-1.2	**Must**	The system shall require a username, password, email address, and real name for user registration.
FR-1.3	**Must**	The system shall allow management to register a new user.
FR-1.4	**Must**	The system shall allow management to assign authorization categories to users.
FR-1.5	**Must**	The system shall allow management to enable or disable registration on the website.
FR-2.1	**Must**	The system shall require user authentication for access.

FR-2.2	**Must**	The system shall allow access to locked rooms by users who are authorized to enter.
FR-2.3	**Must**	The system shall deny access to locked rooms by users who are not authorized to enter.
FR-2.4	**Must**	The system shall support eager locking behaviour where rooms automatically lock when closed.
FR-2.5	**Must**	The system shall support lazy locking behaviour where once opened a room remains open until manually locked.
FR-2.6	**Must**	The system shall store access attempts and their result.
FR-2.7	**Must**	The system shall determine authorization by matching user authorization categories with room whitelist and blacklist categories.
FR-2.8	**Must**	The system shall allow management to change room whitelists and blacklists.
FR-2.9	**Must**	The system shall allow management to change room locking behaviour.
FR-3.1	**Must**	The system shall require at least one form of authentication per user.
FR-3.2	**Must**	The system shall allow rooms to be secured using multiple forms of authentication.
FR-3.3	**Must**	The system shall allow authentication using keycard readers.
FR-3.4	**Must**	The system shall allow authentication using fingerprints readers.
FR-3.5	**Must**	The system shall allow authentication using iris scans.
FR-3.6	**Must**	The system shall allow authentication using passcodes.
FR-3.7	**Optional**	The system shall allow authentication using scanned QR-codes on phones.
FR-3.8	**Optional**	The system shall allow authentication using scanned barcodes on phones.
FR-3.9	**Optional**	The system shall allow authentication using Near Field Communication on phones.
FR-4.1	**Must**	The system shall allow users to book rooms that are not occupied.
FR-4.2	**Must**	The system shall allow management to change the booking schedule.
FR-4.3	**Must**	The system shall allow management to change which rooms can be booked.

FR-4.4	**Must**	The system shall only allow booking of a room for which the user is authorized.
FR-4.5	**Must**	The system shall lock rooms with lazy locking behaviour if the booking period has ended.
FR-4.6	**must**	The system shall support multiple user per booking.
FR-4.7	**Must**	The system shall allow management to enable or disable room booking on the website.
FR-5.1	**Must**	The system shall allow management to view the estimated current occupancy by room.
FR-5.2	**Must**	The system shall determine occupancy based on the latest access history of a user.
FR-5.3	**Must**	The system shall determine occupancy by checking for unlocked rooms with lazy locking behaviour.
FR-6.1	**Must**	The system shall allow management to view a history of occupancy and entry attempts per room.
FR-6.2	**Must**	The system shall allow management to view a history of rooms visited and entry attempts by a user.
FR-7.1	**Must**	The system shall allow doors to be opened from the inside in case of a system failure or emergency situation.
FR-7.2	**Optional**	The system shall be able to adjust HVAC settings based on the presence of users in a room.
FR-7.3	**Optional**	The system shall be able to adjust HVAC settings based on the preferences of users in a room.

3.6 Non-Functional Requirements

This section describes the technical non-functional requirements of the system, based on the following categories: (1) security, (2) usability, (3) reliability and (4) performance. The first number of the identifiers refers to the category to which it belongs.

NFR-1.1	**Must**	Access to the web interfaces is only possible after log in
NFR-1.2	**Must**	Authorisation data can't be directly accessed through the user interface
NFR-1.3	**Must**	The system shall encrypt all data and messages
NFR-1.4	**Must**	Locks used are resistant to physical bypassing or hacking
NFR-2.1	**Must**	End user should not require any training in order to use the system
NFR-2.2	**Must**	Building management should be able to work with the web interfaces within 8 hours of training

NFR-3.1	**Must**	The system should be available 99.9% of the time.
NFR-3.2	**Must**	Authorisation failure should not occur in more than 0.1% of the attempts.
NFR-3.3	**Must**	The system should have a back-up function that can take over authorisation checks when the main system is down.
NFR-4.1	**Must**	The system should be able to operate a request from the user on the web interface within 2 seconds.
NFR-4.2	**Must**	The system should grant access to a room within 3 second after showing the right credentials

4 Analysis

This section gives an overview of the architectural decisions in terms of the pattern that are chosen to realise the system. It also provides an overview of all the risks that are associated with the system.

4.1 Architectural Decisions

In this section an overview of all of the architectural decisions is given. The decisions are based on the PDAP method, which is used to partition the system based on architectural patterns. For every pattern it is explained what issue is resolved by using the pattern, which assumptions and constraints are applicable, which other patterns were considered and a rationale explaining why the pattern was chosen instead of the other alternatives. Finally, the implications that are a result of using the chosen pattern are stated.

4.1.1 Broker

The project contains a few communication paths that need to be secure, as they contain information about the locks and the authentication methods of users. The broker pattern will help to create secure data streams.

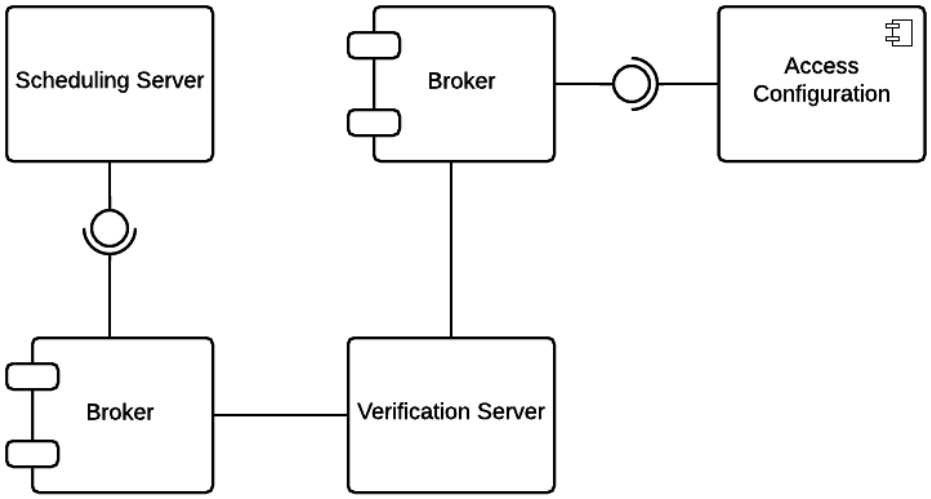

Figure 2: Broker

Source Architectural Patterns Revisited - A Pattern Language [3]

Issue The communication paths between the following components needs to be secure:

- Between the locks and the local communication server.
- Between the local communication server and the verification server.
- Between the access configuration UI and the verification server.

Assumptions/Constraints The different components are deployed on different networks, so that no other communication than the brokered communication can exist between the components.

Positions

- Broker
- Client-Server
- Remote Procedure Call

Decision The communication in the system with security sensitive information will be designed using the Broker pattern. The Broker pattern will hide the actual receiving and sending component, also in the broker security measurements can be taken to, for instance, restrict read or write access to certain parts of the system. This way when the locks are compromised, the communication will still need to go through the broker and reading biometric data from the verification server is impossible.

Argument

1. *Broker* isolates the communication by keeping all the communication related components in a separate layer. All the communication must go through the broker, which greatly increases the security of the system since all the security measures can be implemented by the broker. Using a broker also simplifies the rest of the system, because of the separation of concerns. As a result, it is also easier to add new components to the system, because these components can use the already existing broker for the communication.
2. *Client-Server* is much more general than broker and does not have the same security benefits as the broker pattern. Therefore, it is not used.
3. *Remote Procedure Call* encapsulates a call on a remote object and makes it seem like the call is made on a local object. Since this is not required for the system and it greatly complicates the system as a whole, it is not used.

Implications Using the broker pattern can lead to a less efficient system because of the communication overhead in the broker parts of the system. However, the security of the system increases due to the isolation that the broker pattern can give and the simplicity of the system will increase due to less complex communication parts in the software.

Related Requirements FR-2.2, FR-2.3,FR-2.7, FR-2.7, NFR-1.2, NFR-1.3

4.1.2 Shared Repository

The user information used to open a lock and the data for the scheduling of offices needs to be stored in a database. As it is not allowed to open locks with outdated information, strong consistency is needed for the database.

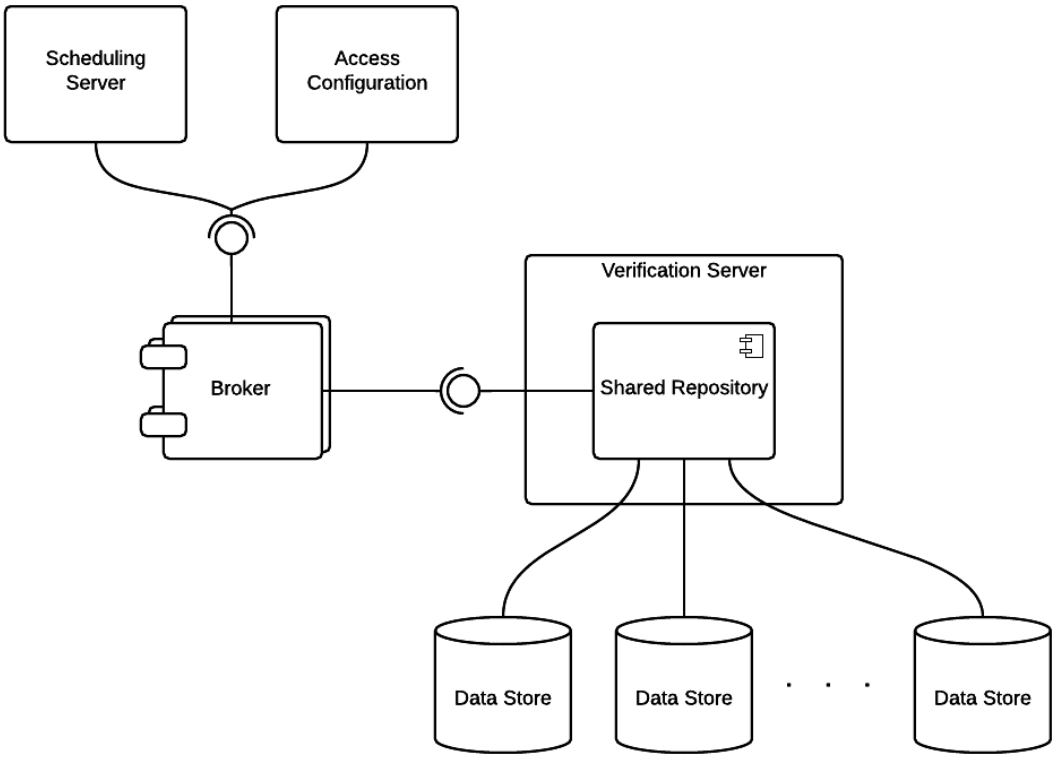

Figure 3: Shared Repository

Source Architectural Patterns Revisited - A Pattern Language [3]

Issue The information must be stored in one place, or at least can be accessed via a single interface. And the data store needs an access control manager that will prevent certain clients from accessing parts of the data store they don't have permission for.

Assumptions/Constraints All components that need to access data stores should be able to use the provided query language to be able to access the data stores.

Positions

- Shared Repository
- Active Repository
- Blackboard

Decision A shared repository will be used to give components access to data stores. The shared repository will be equipped with security measurements so that not all components are allowed to access all data stores.

Argument

1. *Shared Repository* allows all of the components to access the data from a centralised data store. This makes it possible to enforce strong consistency on the stored data and also increases the security of the stored data. The shared repository can implement security measures which can prevent access to certain parts of the data.

2. *Active Repository* is a shared repository that sends notification to subscribers based on certain events. Since it is not necessary that the system is notified by the repository this is not used.

3. *Blackboard* is used for dealing with non-deterministic problems. Because the system does not have such problems, this is not used.

Implications The security of the system can increase by using the shared repository, as it is relatively easy to secure the single interface of the shared repository. However, this also introduces a single point of entry, which makes the costs of having security flaws in the shared repository very high.

Related Requirements FR-2.2, FR-2.3, NFR-1.2, NFR-1.3, NFR-4.1, NFR-4.2

4.1.3 Trusted Subsystem

The biometric data and other privacy sensitive data must be shielded from direct access via web interfaces.

Source SOA Design Patterns [4]

Issue In case someone tries to circumvent the web interface to get direct access to the shared repository these attempts must be stopped to maintain the integrity of the data. If someone changes the fingerprint data of a person with a higher security clearance with their own fingerprint he could get access to areas he's not supposed to access.

Assumptions/Constraints The web interfaces are capable of sending the credentials of its current user to the system so that he can be authenticated by the system.

Positions

1. Trusted Subsystem

Decision The system will use a trusted subsystem to increase the security between the web interfaces and the system, and more specific the security of the privacy sensitive data in the system.

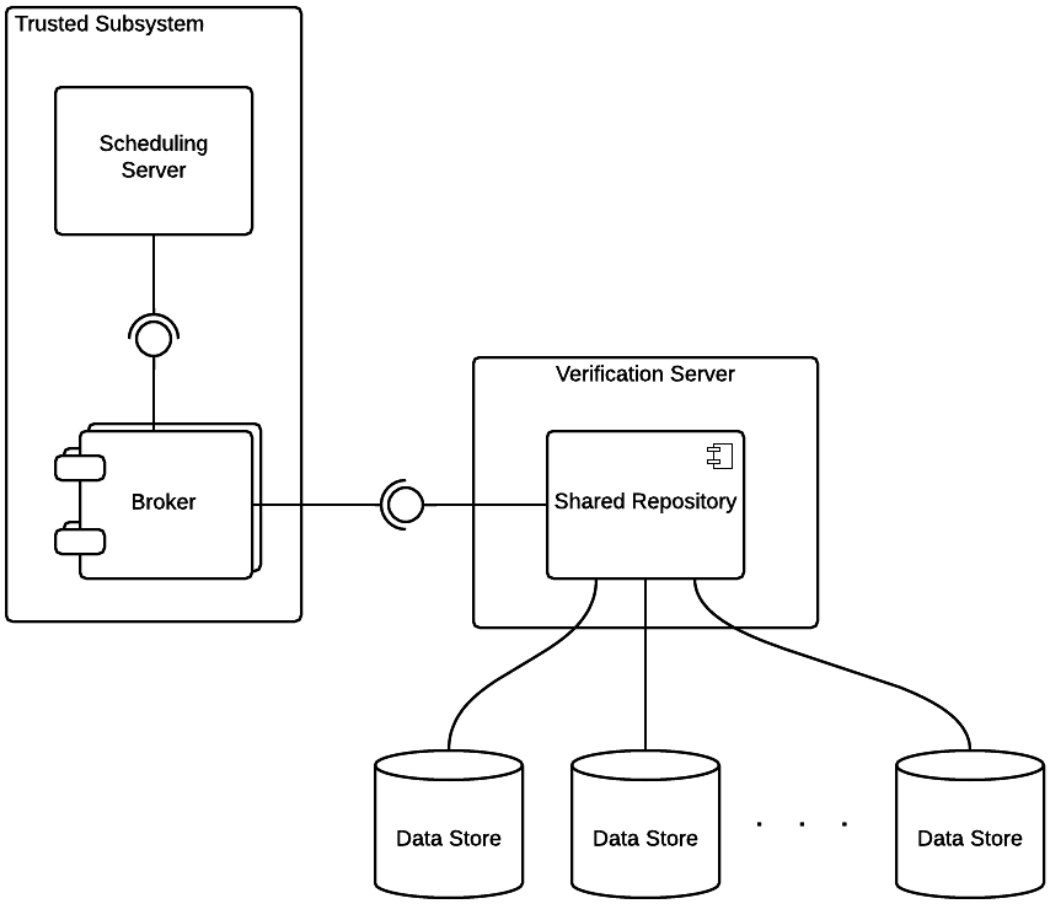

Figure 4: Trusted Subsystem

Argument

1. *Trusted Subsystem* enforces that the communication between the web interface and the shared data repository is authenticated by the trusted subsystem. This prevents other systems from directly accessing the shared data repository, which greatly increases the security of the system.

Implications The security of the system is increased by limiting the direct access to the shared repository. However, this again leads to a single component that can communicate with the shared repository, which will be the starting point for an attacker to attack the system. Therefore, it must be ensured that the trusted subsystem is very secure.

Related Requirements FR-3.4, FR-3.5, NFR-1.2, NFR-1.2, NFR-1.3

4.1.4 Layers

The scheduling and the access configuration will both be implemented as a web-service. The layers pattern will be used for the separation of concerns and will result in a loose coupling between the services on the different layers.

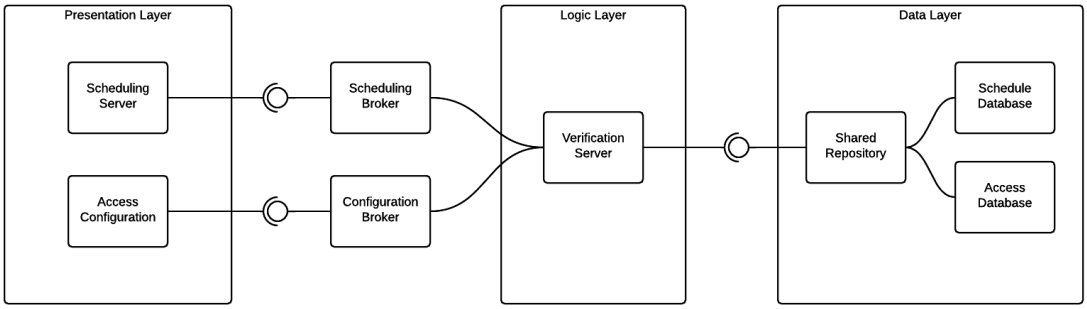

Figure 5: Layers

Source Architectural Patterns Revisited - A Pattern Language [3]

Issue The components that make up the web-services have different responsibilities and should be decoupled from each other based on their responsibilities. Communication between the different groups of components should still be possible, but not every group of components should be able to access all of the other groups.

Assumptions/Constraints Both the web-services are large components that require a decomposition into multiple layers.

Positions

1. Layers
2. Relaxed Layers

Decision The layers pattern is used to organise the system into multiple layers. The system is organised in such a way that a layer will only interact with the layer that is positioned right below that layer through a clearly defined interface. All of the components that reside on the same layer will operate at the same level of abstraction and can only communicate with components on the same layer or the layer right below it.

Argument

1. *Layers* separates the system into multiple layers which results in the separation of concerns within the system. This in turn enhances the maintainability and the flexibility of the system. The layers pattern also increases the security of the overall system, since it is possible to implement security related functionality at every layer.
2. *Relaxed Layers* allows some of the layers to communicate with a layer that is not directly below it. This has a great impact on the security of the system, which is one of the key-drivers of the system. Therefore, the relaxed layers pattern is not used.

Implications The introduction of layers hinders the performance of the system. Every request has to travel through multiple layers, which can slow down an operation significantly. However, it does increase the security of the system as well as the modifiability.

Related Requirements NFR-1.2, NFR-4.1, NFR-4.2

4.1.5 Model-View-Controller

Both of the user interfaces in the system use the layers patterns for the separation of concerns. For both these systems the presentation layer will be implemented using the model-view-controller pattern.

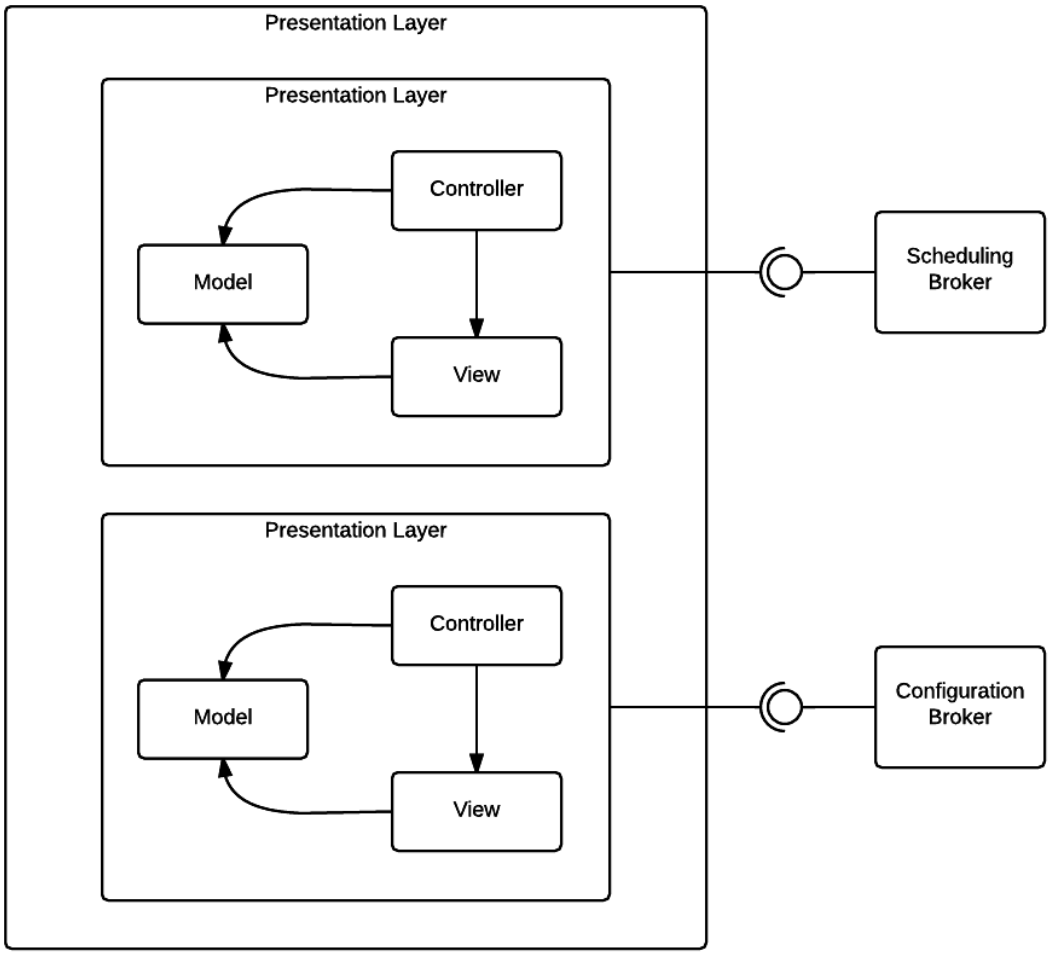

Figure 6: Model-View-Controller

Source Architectural Patterns Revisited - A Pattern Language [3]

Issue A clear separation of concerns is required between the user interface and the application logic and data, because the system offers multiple interfaces that display the same data in different ways.

Assumptions/Constraints The system offers multiple views of the same application data.

Positions

1. Model-View-Controller
2. Presentation-Abstraction-Control

Decision The model-view-controller pattern will be used to decouple the presentation from the application logic. The pattern will be used in the presentation layer of both the user interfaces of the system.

Argument

1. *Model-View-Controller* gives a clear separation of concerns, making it possible to use multiple views for the same model.
2. *Presentation-Abstraction-Control* is not suitable in this case, since it decomposes the system into multiple agents. Every agent is responsible for a specific functionality and maintains its own state.

Implications The use of the model-view-controller pattern gives a clear separation of concerns within the presentation layer. This separation of concerns gives better readability of the code and makes it possible to display the same data in different ways with the use of multiple views. However, it does increase the complexity of the system and also decreases the performance of the system.

Related Requirements HL-1, HL-4, HL-5, HL-6, FR-1.1

4.1.6 Load-Balanced Cluster

Both reliability and usability are key-drivers of the system. For this reason, it is very important that the system is available to the users and that the system has good performance. The Load-Balanced Cluster is used to improve upon these attributes.

Source Enterprise Solution Patterns Using Microsoft .NET [5]

Issue The scheduling server should be highly available. Therefore replication is used to deal with the failure of one of the scheduling servers. However, one of the scheduling servers in the cluster could also be receiving too much requests from the users of the system. Since this can also cause the scheduling server to fail, the load must be balanced equally across all of the scheduling servers in the cluster.

Assumptions/Constraints It should be possible to run the tasks of the scheduling server concurrently.

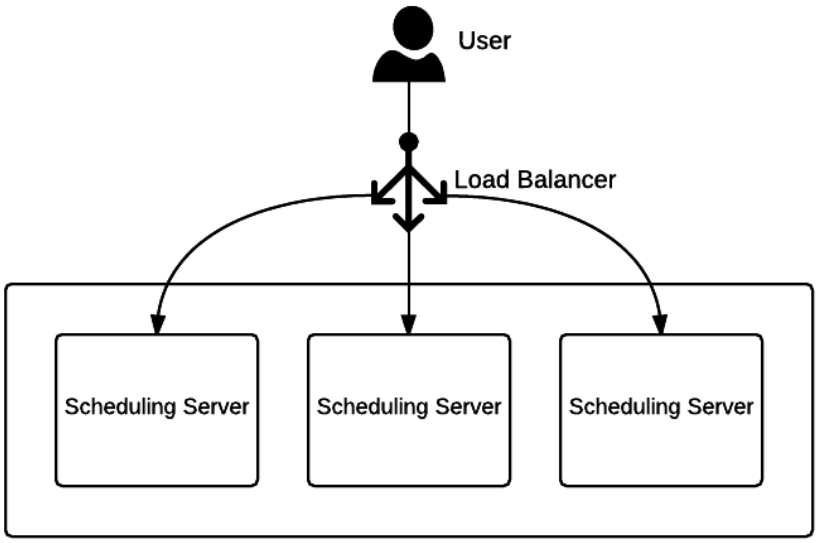

Figure 7: Load-Balanced Cluster

Positions

1. Load-Balanced Cluster
2. Failover Cluster

Decision The Load-Balanced Cluster will be used in the system to increase the availability of the scheduling server and to make the system better capable of handling large peaks in requests made to the scheduling server. The pattern also allows the system to scale when the cluster is not capable of handling the load. Extra scheduling servers can be added to the cluster to which traffic can be redirected without the user being aware of it.

Argument

1. *Load-Balanced Cluster* allows the system to be able to deal with failures of at least one of the servers in the cluster. Furthermore, it also allows the system to distribute load across all of the servers in the cluster.
2. *Failover Cluster* allows the system to deal with failures, but only one of the servers in the cluster is actively used.

Implications Using a Load-Balanced Cluster makes the system more complex, but gives an increased availability and reliability of the system. It should also be noted that the scheduling server cannot maintain any state without taking additional measures, since subsequent requests of a the same user are not always routed to the same server.

Related Requirements NFR-3.1, NFR-4.1, NFR-4.2

4.1.7 Failover Cluster

The access points send requests to the verification server along with the user credentials. The verification server then verifies the supplied credentials and if the credentials are valid, the verification server sends an unlock command to the access point. The access configuration and scheduling server also communicate with the verification server to fetch and store data in the data stores. This means that the verification server is a single point of failure in the system and the failover cluster is used to prevent the entire system from failing when the verification server fails.

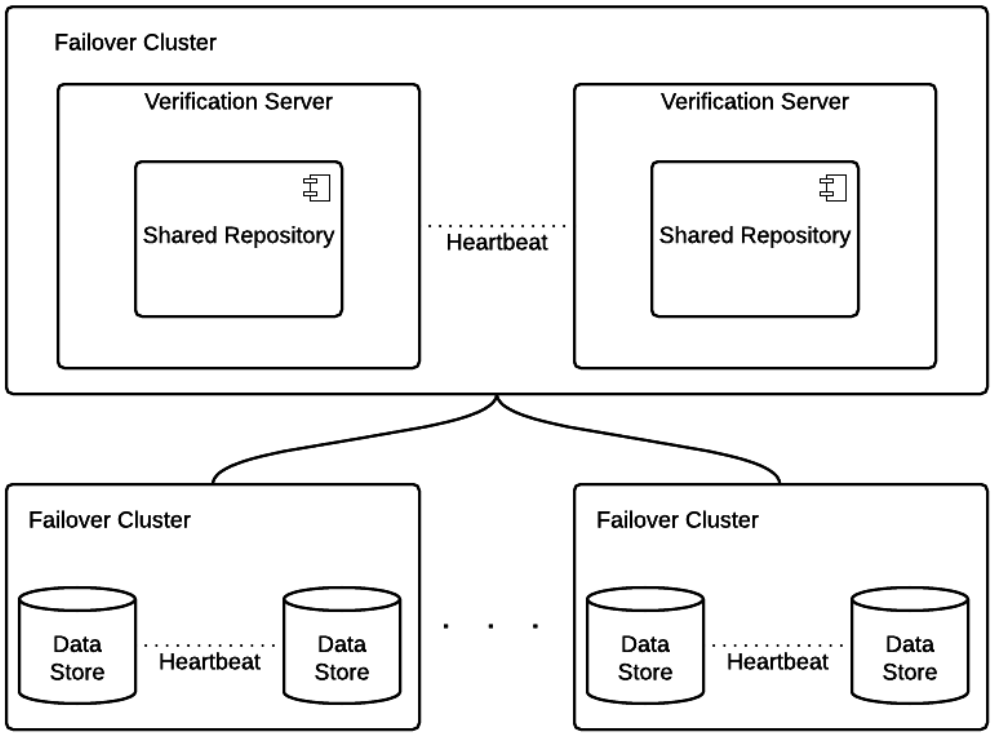

Figure 8: Failover Cluster

Source Enterprise Solution Patterns Using Microsoft .NET [5]

Issue The verification server handles all the requests from the locks and determines whether the lock should unlock. This makes the verification server a single point of failure within the system, which means that is no longer possible to use any of the locks if the server should fail. Additionally, the data stores that the verification server uses to store all the scheduling data and the access data are also a single point of failure.

Assumptions/Constraints If the verification server fails while it is processing one or more requests, the locks use a timeout and will notice that no reply is received.

Positions

1. Failover Cluster
2. Load-Balanced Cluster

Decision A failover cluster is used to increase the availability of the verification server. The server is replicated across multiple servers, but only one of those servers is actively processing requests at the time. When the active server fails, one of the other servers in the cluster can take over.

Argument

1. *Failover Cluster*
2. *Load-Balanced Cluster*

Implications

Related Requirements NFR-3.1, NFR-3.3

4.1.8 Indirection Layer

The system offers multiple payment methods to the users who rent an office. To decouple the interaction with the payment providers from the rest of the application and to allow new payment providers to be added to the system an indirection layer is added between the system and the payment providers.

Source Architectural Patterns Revisited - A Pattern Language [3]

Issue When a user rents an office through the web server the user has to pay for the time slot that is being requested. The system shall offer the user different payment methods, which should be decoupled from the rest of the system. It should also be possible to add additional payment methods in the future.

Assumptions/Constraints None

Positions

1. Indirection Layer
2. Interceptor

Decision An indirection layer is added between the system and the payment providers. The layer hides the implementation details for each of the individual payment providers and the system can interact with each of the payment providers in the same way. Because of the indirection layer it is also possible to add new payment providers in the future, without having the change the system itself.

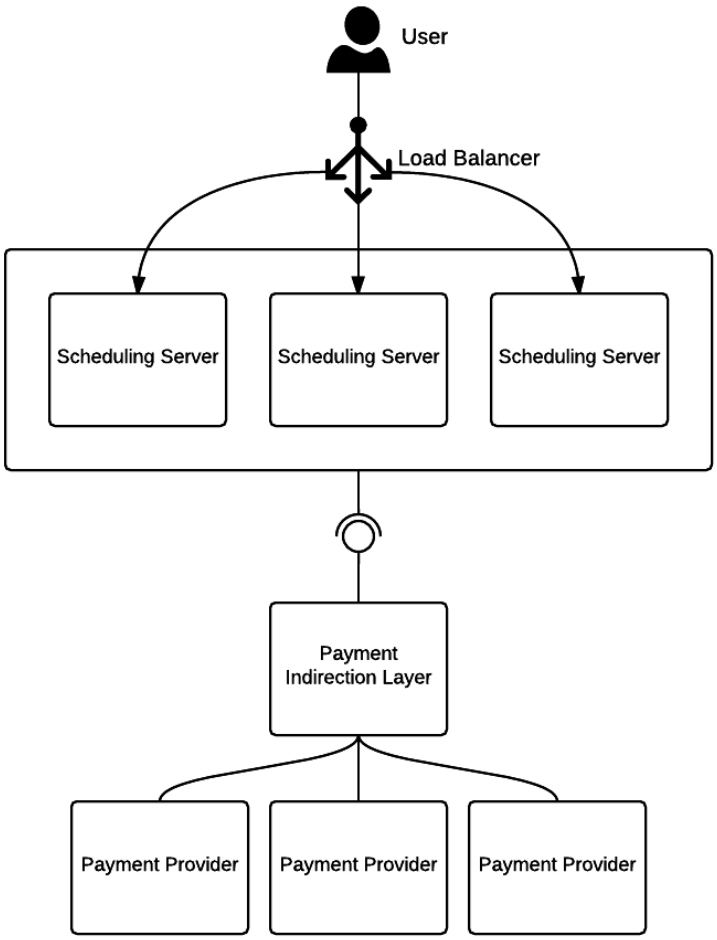

Figure 9: Indirection Layer

Argument

1. *Indirection Layer* adds a layer between the system and the modules that handle the interaction with the payment provider. This allows the system to interact with each of the payment providers in the same way and it gives a clear separation of concerns. It also allows new payment providers to be added to the system very easily.

2. *Interceptor* facilitates changes in the system in a transparent and automated way. However, adding additional payment services is not something that should be done in a transparent way.

Implications The indirection layer adds additional indirection to the system, which has a negative influence on the performance of the system. However, it is relatively easy to add new payment providers to the system with the indirection layer in place.

Related Requirements NFR-2.1, NFR-4.1

4.1.9 Message Queueing

Whenever someone uses an access point, the access point sends a request to the verification server along with the credentials the user has provided. The verification server will then verify the credentials and send a reply to the access point. Because there can be situations where many access points are used at the same time, some buffer mechanism is required, which is provided by Message Queueing.

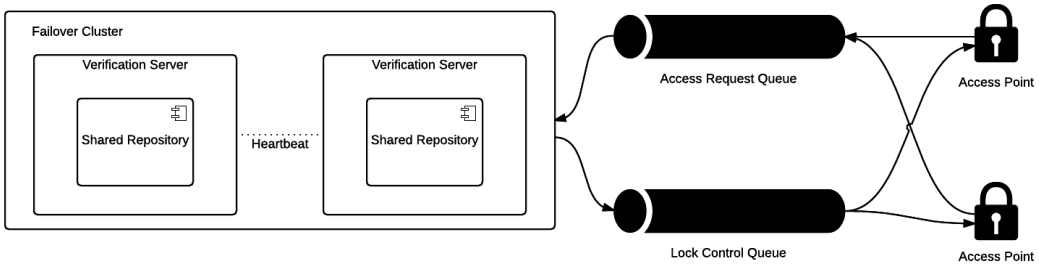

Figure 10: Message Queueing

Source Architectural Patterns Revisited - A Pattern Language [3]

Issue A user has to use some kind of credentials (pin code, fingerprint, etc.) when that user wants to gain access to a certain access point. The access point sends the request along with the credentials to the verification server. The verification server verifies the credentials and sends back a reply. However, the system should also be able to process a large number of requests when many access points are used at the same time. Furthermore, it should be possible to unlock a certain access point via the access configuration. For this to work, the access point must be able to receive an unlock command even when it has not send a request to the verification server.

Assumptions/Constraints The queue is large enough to store a large amount of requests in the case that many requests are coming from the access points and the access control is not able to process all of these requests immediately. The queue that is used to send replies to the access points must also be large enough to hold many replies in the case that some messages in the queue are not processed soon after they have been stored in the queue.

Positions

1. Message Queueing
2. Publish-Subscribe

Decision The requests from the access points and the replies to the access points are decoupled from each other. The access point pushes requests to a message queue from which the requests can be read by the verification server. When the verification server has verified the credentials, it pushes a reply to another queue.

Argument

1. *Message Queueing* decouples the sender and the receiver from each other and can act as a buffer between the two components. This allows the system to deal with a peak in the amount of requests that are sent. The separate queues also allows the verification server to unlock an access point when the access point has not sent a request to the verification server.

2. *Publish-Subscribe* also decouples the communication between the access points and the verification server. It has the same benefits as the Message Queues, but also allows multiple subscribers for the same events. However, since this is not used by the system and increases the complexity of the system, Publish-Subscribe is not used.

Implications The queue decouples the access point form the verification server and also prevents the verification server form sending a reply back to the access point. However, the queue also acts as a buffer between the two components, which allows the system to handle peaks in the amount of requests coming from the access points. When many requests are received, they are stored in the queue and the verification server can determine the rate at which it processes the requests. This does have a negative influence on the performance, since some requests are stored in the queue for some time.

Related Requirements FR-2.9, NFR-3.3

4.2 Risk Assessment

Table 12 in Appendix A2 shows the risks that are associated with the system. All of the risks are classified based on the severity of the risk, which is obtained by taking both the probability that the problem is encountered and the cost involved in mitigating the risk into account. Table 7 shows all the possible severity classifications that can be assigned to a risk.

	Probability		
Cost	− (Low)	∼ (Medium)	+ (High)
− (Low)	−− (Very Low)	− (Low)	∼ (Medium)
∼ (Medium)	− (Low)	∼ (Medium)	+ (High)
+ (High)	∼ (Medium)	+ (High)	++ (Critical)

Table 7: Risk severity derivation

5 System Architecture

This section will elaborate on the high level hardware and software components in the system. The system architecture is made visible in Figure 11. The rest of this section will clarify this figure.

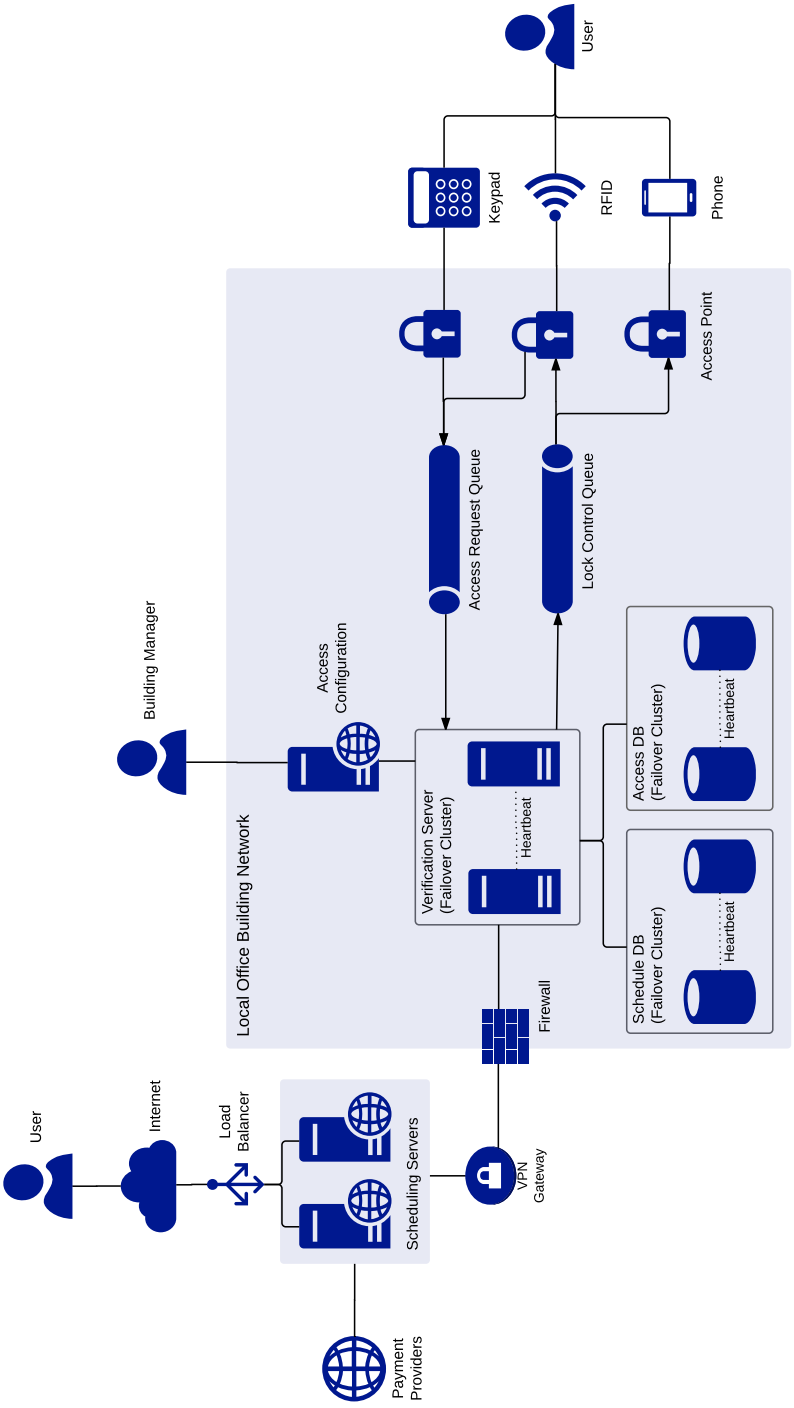

Figure 11: System Architecture

The system can be decomposed in two main parts: Access Control and Scheduling. The access control component is responsible for opening the correct lock at the correct moment, and the scheduling component is responsible for the office reservations for flex offices.

5.1 Access Control

Access control is responsible for opening locks, this includes all of the components that have to do with the physical locks but also the authentication and verification of users.

All of the access control components are located in the same local network of the office building. In case a customer has multiple buildings, where at least some of the employees or some of the flex workers needs access to multiple buildings, a shared local network can be created using VPN tunnels.

5.1.1 Access Point

The Access Points are the physical locks and controllers that are located at the doors to open and close the rooms. At each Access Point there is at least one authentication device coupled, with which data is sent to the verification server via the local network. The Access Points publish a message in the Access Request Queue when someone tries to authenticate himself, this way the queue acts as a buffer in case too many messages are sent by the Access Points. A time to life value will be set for the messages, as it is normal behaviour to re-enter your credentials in case the lock doesn't open fast enough.

Controllers inside the Access Points subscribe to messages from the Lock Control Queue with a routing key unique for each lock. Using this queue based communication locks can also be opened without requiring an Access Point to send a message first, this way building managers are capable of opening locks remotely.

5.1.2 Access Configuration

The Access Configuration component is the interface that the Building Mangers use to manage users and Access Points. When new physical locks are added they need to be inserted and configured in the system via the Access Configuration. This is also the interface where user information can be altered, when, for instance, an employee needs a higher or lower security level. Also, building managers can manually open locks via the interface.

This interface can only be accessed by devices inside the local network of a building, so building managers need to be physically present at the building to modify information about users or locks. Of course a VPN tunnel can be set up so that one person can manage multiple buildings.

5.1.3 Verification Server

In the Verification Server the computations are done to identify users given certain authentication information. It receives the request from the Access Points via messages in the Access Request Queue containing the authentication information and the lock identifier. The Verification Server will check first whether or not the user can be found with the authentication information and if that user has enough security clearance to open the lock. The Verification Server also checks if the user did reserve the room for the current time slot if appropriate. After performing all these checks the Verification Server will publish a message with a open or close command in the queue with a routing key belonging to the lock in the Lock Control Queue.
The Verification Server will use the failover cluster pattern to ensure the availability of its service.

5.1.4 Access Database

The Access Database is the data store that contains all the information about the users and locks in the system. For each user there is at least one authentication method stored, but this could grow so that users can authenticate themselves via multiple methods.

The Access Database will be deployed as a failover cluster, so that in case a server fails another one will take over the load without downtime. As server capacity won't likely be a bottleneck for the access database, the replicated servers won't be used for reading or writing.

5.2 Scheduling

The Scheduling component is responsible for giving users an interface to be able to make reservations for rooms. The Scheduling component has a large portion of subsystems that are not located inside the local network of the buildings. Therefore, a secure firewall must be placed between those subsystems.

Users can access the scheduling interface via the Internet so that users are able to view the availability of flex offices and to reserve a room. Since the scheduling is mainly created for flex offices rather than for traditional companies that only need access control, the Scheduling component can be turned off completely.

5.2.1 Scheduling Server

The Scheduling Server provides the web interface for users and connects with the Verification server via the VPN gateway and through the Firewall. The web interface enables the users to view the availability of rooms and to reserve rooms. Payments can be done directly during the reservation of a room. Furthermore, overviews can be generated so that users have a clear picture of their expenses.

As the Scheduling Server is accessed by users via the Internet and no well predefined user load can be expected, the Scheduling Server will be deployed as a Load-Balanced cluster.

5.2.2 Schedule Database

The Schedule Database is a data store containing all the information regarding reservations. This information includes the user identifier, the room identifier and timestamps.

5.2.3 Payment Providers

The Scheduling Servers use external payment providers to give users the ability to directly pay for a reserved room.

5.2.4 VPN

A virtual private network (VPN) is used to extend the local private network across several buildings. With this technology a secure network can be created while enhancing the usability of the system.

5.2.5 Firewall

The Firewall between the Scheduling Server and the Verification Server needs to be very strict. The kind of messages the Scheduling Server will send to the Verification Server are known in advance, so the Firewall should keep a white-list for these messages and block all other messages.

6 Hardware Architecture

This section describes the hardware architecture of the system. This section starts by giving a overview of the architecture, followed by a more detailed description of the hardware components.

6.1 Hardware overview

The overview of the hardware architecture is illustrated in figure 12. The PASSLOCK system can be divided into two main categories of components, the server components and the locking and authentication components.

At every access points contains a physical electronic lock, one or more authentication methods and a controller. the type of connection between this parts will depend on its specifications. Between the controller and the locks a switch will be implemented in order to put power on the lock. Between the controller and the authentication method USB or Ethernet is used.

The verification servers are placed locally in order to keep privacy sensitive data from being tapped or stored in uncontrolled environments. The scheduling servers are placed in data centres and cant be access through the Internet by users.

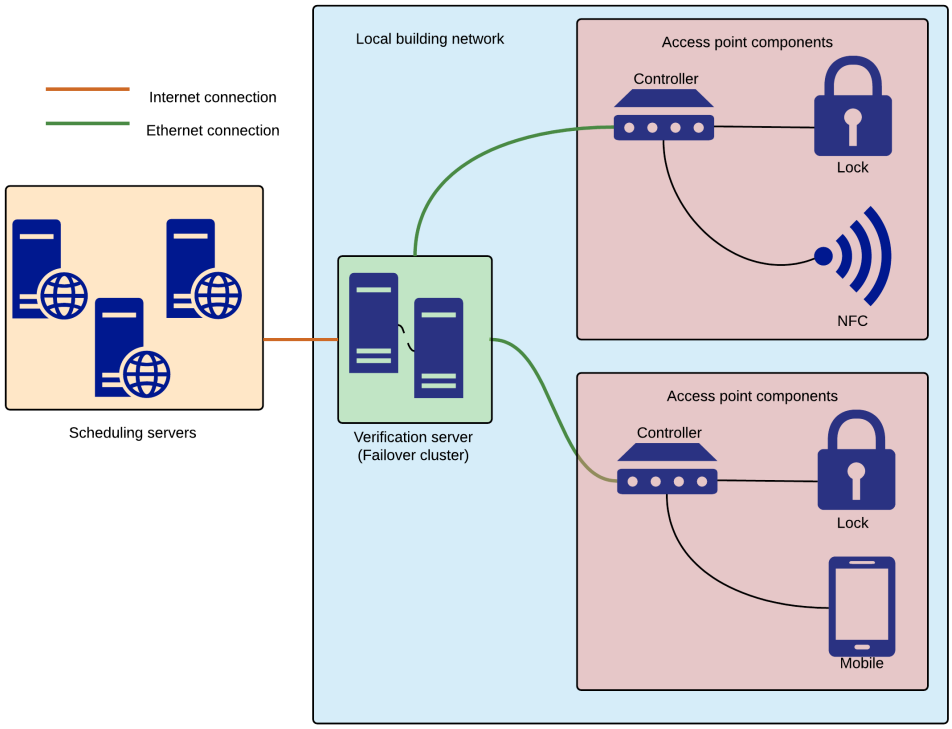

Figure 12: Hardware Architecture

6.2 Hardware descriptions

This section briefly describes the chosen hardware for the system. Hardware for optional authentication methods, like QR-codes, are not yet included.

6.2.1 Electronic lock

For the actual lock two different types are used. Depending on the level of security needed this can be a fail-safe lock or a fail secure lock. The difference is in the way a lock handles a power failure. The fail-safe lock needs power to be locked, the fail-secure lock needs power to unlock.

The locks can be opened from the inside at all times to comply with the EN79 standard about emergency exits. Furthermore the locks are equipped with a feedback signal to inform the system about the position of the lock. The locks work with 12V/DC.

Depending on the need for fail-safe or fail-secure the following locks are available:

- Mauer® 5492 E-AP
- Mauer® 5592 E-AP

6.2.2 Passcode/smartcard

For the authentication by smartcard, passcode or mobile phone a multifunctional scanner is used from HID global, the iCLASS SE RKPK40. This is a wall device that uses a keypad for passcodes, Bluetooth for Mobile IDs technique from HID and a smart-card reader for the authentication of user with a smart-card. The device can use wiegand for communication with the system. Specifications are displayed in table 8.

Component	**Specifications**
Manufacturer	HID global
Model	iCLASS SE RPK40
Read range Bluetooth	2 metre
Transmit capabilities	2.4GHz, 13.56MHz
Possible extension	Support for multi-factor authentication
Power	16V/DC
Product link	`http://www.hidglobal.com/sites/hidglobal.com/files/resource_files/hid-mobile-access-readers-nfc-ds-en.pdf`

Table 8: Specifications for the smart-card and NFC scanner with keypad

6.2.3 Fingerprint scanner

The fingerprint scanner comes from biometric systems. We use the EFIS121. This is a ethernet fingerprint scanner that provides Power over Ethernet to the scanner itself. Specifications are displayed in table 9. Included SDKs allow for integration within our system.

Component	**Specifications**
Manufacturer	ABS Applied Biometric Systems GmbH
Model	IFES121 PoE 802.3af
Image size	280 x 440 pixels
Image resolution	508dpi
Power	Power over Ethernet with external 9-35V/DC
Product link	`http://www.biometricsys.de/ethernet-fingerprint-scanner-efis121poe.html`

Table 9: Specifications for the fingerprint scanner

6.2.4 Iris scanner

The iris scanner used is the Eyelock nano NXT. This compact scanner provides great performance for single access points like doors. While by default the nano NXT does onboard iris matching using an internal database, the device also supports the complex integration with our system using an SDK. Specifications are displayed in table 10.

Component	**Specifications**
Manufacturer	Eyelock
Model	nano NXT
Capacity	up to 20 people per minute
Capture distance	Up to 12 inches
Possible extension	Support for multi-factor authentication with key-card
Power	Power of Ethernet or 24V/DC
Product link	`http://www.eyelock.com/index.php/products/nano-nxt`

Table 10: Specifications for the iris scanner

6.2.5 Communication server

The communication between the physical lock and its authentication method and the verification server will be done using a raspberry pi. This component is responsible for passing through the data from the authentication method and after response from the verification server the unlocking or locking of a lock. The raspberry pi offers a greatly flexible module that can be adjusted to the needs of the PASSLOCK system. The newest model, the raspberry pi 2 model B, is used. Specification are presented in table 11

Component	**Specifications**
Manufacturer	Raspberry pi foundation
Model	Raspberry pi 2 model B
CPU	900MHz quad-core ARM cortex A7
RAM	1GB
Inputs/Outputs	4 USB, 40 GPIO pins, Ethernet, HDMI, microSD, camera interface, display interface
Power	5V over USB
Product link	`https://www.raspberrypi.org/products/raspberry-pi-2-model-b/`

Table 11: Specifications for the communication server

7 Software Architecture

In the following section, the software architecture of the PASSLOCK system will be described. First of all, an overview of the initial model of the architecture will be given. Secondly, the subsystems of the entire system will be highlighted, including their architecture and their forces with respect to the key drivers. Finally, an elaborated model of the architecture will be given, in which the patterns as discussed in Section 4 will be highlighted and its implementation will be described in detail, as well as their forces.

7.1 Initial Model

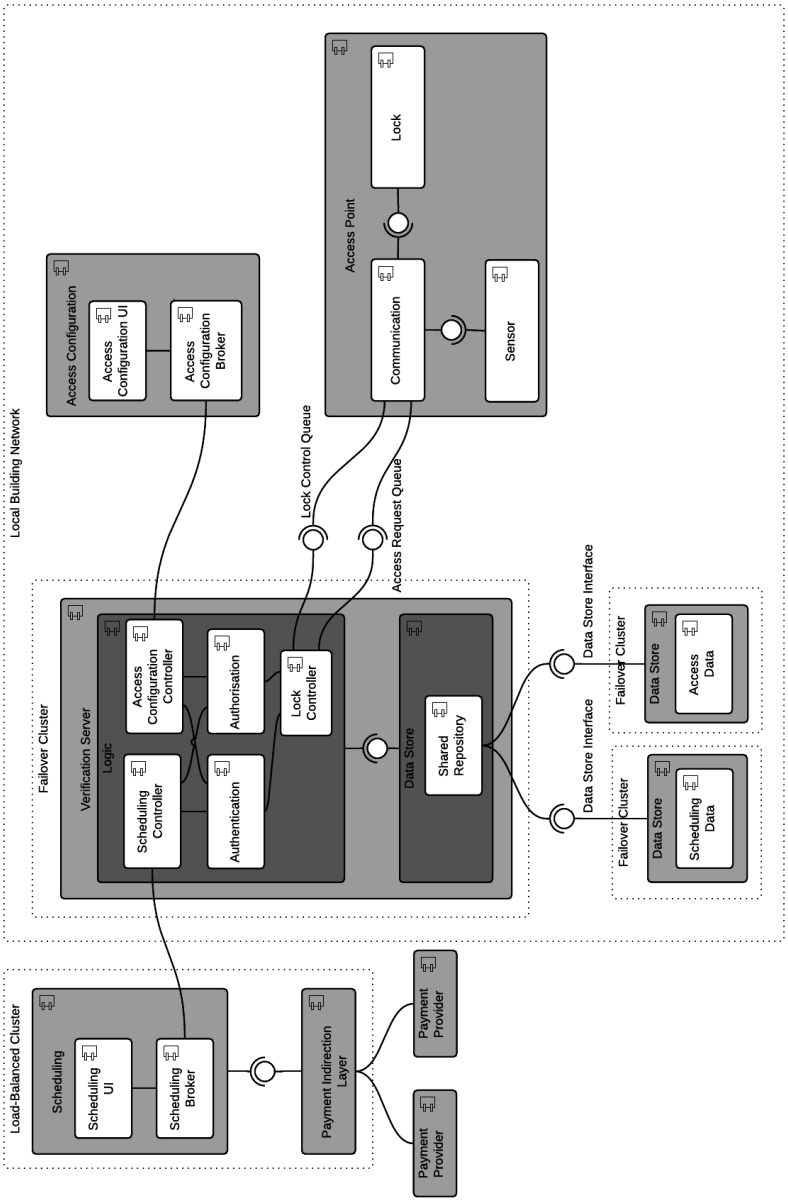

Figure 13: Initial Model

7.2 Subsystems

In the following sections, each subsystem of the PASSLOCK system will be identified and described individually, specifically what their impact is on the key drivers of the system: Security, Reliability and Usability.

7.2.1 Verification Server

The Verification Server acts as the core of the system. It provides mechanisms for authentication and authorisation and acts as a shared repository for the Scheduling and Access Configuration subsystems.

Access Points communicate with the `Lock Controller`. This component receives authentication data (such as fingerprint scans) from the Access Points through the `Lock Control Queue`. The data is used by the `Authentication` component to identify which user (if any) attempted to authenticate. The resulting user account and the access point identification sent with the request is used by the `Lock Controller` to query the `Authorization` component. When authorisation is successful, the controller sends an unlock command to the Access Point through the `Lock Control Queue`.

The Scheduling subsystem communicates with the `Scheduling Controller`, which performs user authentication and authorisation to determine if users are allowed to book certain rooms. The controller also manages data storage using the shared repository.

The Access Configuration subsystem communicates with the `Access Configuration Controller` to modify authentication, authorisation and configuration data in the shared repository.

Security The verification server is the only component that has direct access to both of the data stores. Since credentials such as fingerprints, iris scans and pin codes are stored here, the security of the verification server is very important. For this reason, the verification server is only accessible on the local network of a building or group of buildings, or via a VPN. It also enforces strong access control to ensure that other components can only access the data that is really required by that component.

Reliability Because the verification server is the only component that has direct access to the data stores, all the other components have to communicate with the verification server. This means that the verification server is a single point of failure in the system. To mitigate this problem, a fail-over cluster is used where multiple servers can act as verification server, but only one of those servers is active at any given moment. As soon as the active server fails, one of the other servers in the cluster can take over.

Usability The usability of the overall system is very important and one of the aspects of usability is the performance of the system. For a user of the system it is undesirable to wait too long before an access point is being unlocked. Therefore the request of an access point must be processed as quickly as possible. However, the strong security of the verification server has a negative impact on the performance of the system.

7.2.2 Access Point

The Access Points are the actual lock and authentication methods positioned near doors. The software for the `Access Point` is split up into three parts: a `Communication` component, a `Lock` component and a `Sensor` component.

The `Communication` component handles the communication between the `Verification Server` and the `Lock` and `Sensor`. It publishes the information received from the `Sensor` to the `Access Request Queue`, in the message it will include the sensor reading and the `Lock` identifier. The `Communication` component will also subscribe to messages in the `Lock Control Queue` that are published with a routing key corresponding to the `Lock` identifier. Routing keys can be composed as follows: `lockcontrol.CUSTOMERID.BUILDINGID.ROOMID.LOCKID`. When the communication component receives a message from the queue it will control the `Lock` given the contents of the message.

The `Lock` component is fairly simple, it is a bridge between the `Communication` part and the physical lock. It should forward the open and close commands sent from the `Communication` component. Besides that the `Lock` component should also be able close a lock after a specific timeout in case a customer wants authentication for everyone passing that lock.

The `Sensor` component acts as a bridge between the physical sensor and the `Communication` component. The `Sensor` component translates signals coming from the sensor and tags it with sensor type so that the `Verification Server` knows which kind of information it needs to process. Some pre-processing is also done in the `Sensor` component, examples of this pre-processing are: detect if the sensor sends the same signal more than once and detect if certain thresholds for sensor values are met (i.e. having a long enough PIN).

Security The access point does not verify the credentials that are provided by the user. Instead, it sends a request to the verification server via a queue. The verification server will then process this request locally and will send an unlock signal to the access point in the case that the credentials that were provided can indeed be used to unlock that access point. This makes unauthorised access to an area protected by an access point much harder, because to gain access the verification server has to be compromised or the lock itself has to be compromised.

Reliability The components that comprise the access point are chosen carefully such that they have a good reliability. However, no additional measures are taken to make the software components of the access points more reliable. In the case of a failure the access point should be rebooted or replaced.

Usability The most important aspect of the usability of the system with regard to the access points is the time it takes for the access point to unlock given valid credentials. Because the access point's request is sent to the verification server via a queue, the performance of this process is somewhat affected. However, since everything happens on a local network the performance impact should not be too high.

7.2.3 Access Configuration

The Access Configuration subsystem allows building management to change settings for scheduling and authorisation. This is done through the `Access Configuration UI`. This web portal lets the management to view the schedule and make changes to reservations. Also a management tool is available to change the authorisation methods for single users, changing white- and blacklist for rooms and put users on these lists. The `Access Configuration UI` communicates with the Verification server using the `Access Configuration Broker`.

Security The access configuration system needs to be very secure, since this system gives the possibility to alter access rights of a certain user, add or change the credentials of a certain user, and allows access points to be unlocked remotely. For this reason, the access configuration server can only access the verification server via the local network. Furthermore, the communication between these two system is done via a broker, which gives an increased level of security.

Reliability An access configuration server is located in every building, which are deployed independently from each other. If one of these servers fails, the other servers are not affected by this. However, there are no measures in place to mitigate the problem of one of the access configuration servers failing. Since it is a single server, the failure of such a server is not very likely and the impact of the access configuration server being down is not that much. The access configuration is only used to add new users to the system or alter the credentials of a user, which does not happen that often.

Usability The usability of the access configuration is impacted by the negative impact on the performance of the system by the additional security measures such as using a broker for the communication between the access configuration and the verification server. The performance is further impacted by the security measures that are in place at the verification server. But since only the building management has access to the system and the use of the system is limited, the usability of this system is of less importance than the usability of the scheduling server.

7.2.4 Scheduling

The Scheduling subsystem allows users to book rooms using the `Scheduling UI`. This website allows users to log in with an existing account or register a new account. Users are shown a list of rooms that are bookable by the user, as determined by scheduled occupancy and user authorization. Users can book these rooms for a chosen time period. The `Scheduling UI` uses the `Scheduling Broker` to communicate with the Verification Server and the database. When processing a booking, the user can be required to pay for the booking. This can be done trough several payment providers, which can be accessed uniformly using the `Payment Indirection Layer`.

Security The scheduling server needs to access the data store that stores information related to the scheduling of offices. To improve the security of the system, this data store is not directly accessible by the scheduling server, because if the scheduling server would be compromised it would be possible to alter the data that is being recorded by this data store. However, if the scheduling server would have unrestricted

access to the verification server, the security of the system could also be impacted. For this reason, the scheduling server acts as a trusted subsystem that has to authenticate itself to the verification server. The scheduling server is only able to communicate with the verification server when the correct credentials are provided and can only access certain parts of the data store through the verification in this way. This ensures that the scheduling server does not have access to information that is not required by the component and it limits the impact if the scheduling server were to be compromised.

Reliability The reliability of the scheduling server is very important, because the users have to use this system to rent offices and administrators can use this system to view information about a certain building or office. For this reason, a cluster of scheduling servers is used. This cluster will continue to function even when one of the servers in the cluster fails, which will greatly increase the reliability of the system.

Usability For the scheduling server, there are several important aspects to the usability of the scheduling server. The reliability and availability of the system, which was covered in the previous point, is very important, because the users have to be able to use this system. Another important aspect is the performance of the system. Because of the strong security, accessing information in the data store is somewhat slower. However, the requests of the users can be handled quite easily, because the requests of all of the users is evenly distributed among the servers in the cluster using a load balancer. This makes the scheduling server more responsive and also allows the scheduling server to continue to operate when a large peak in traffic is encountered.

7.3 Elaborated Model with Patterns

In the following sections, an elaborated model of the system will be described on the level of patterns as discussed in Section 4. Each pattern will be highlighted in the architecture, its implementation will be described, as well as their forces on the three key drivers of the system.

7.3.1 Model-View-Controller

7.3.2 Trusted Subsystem

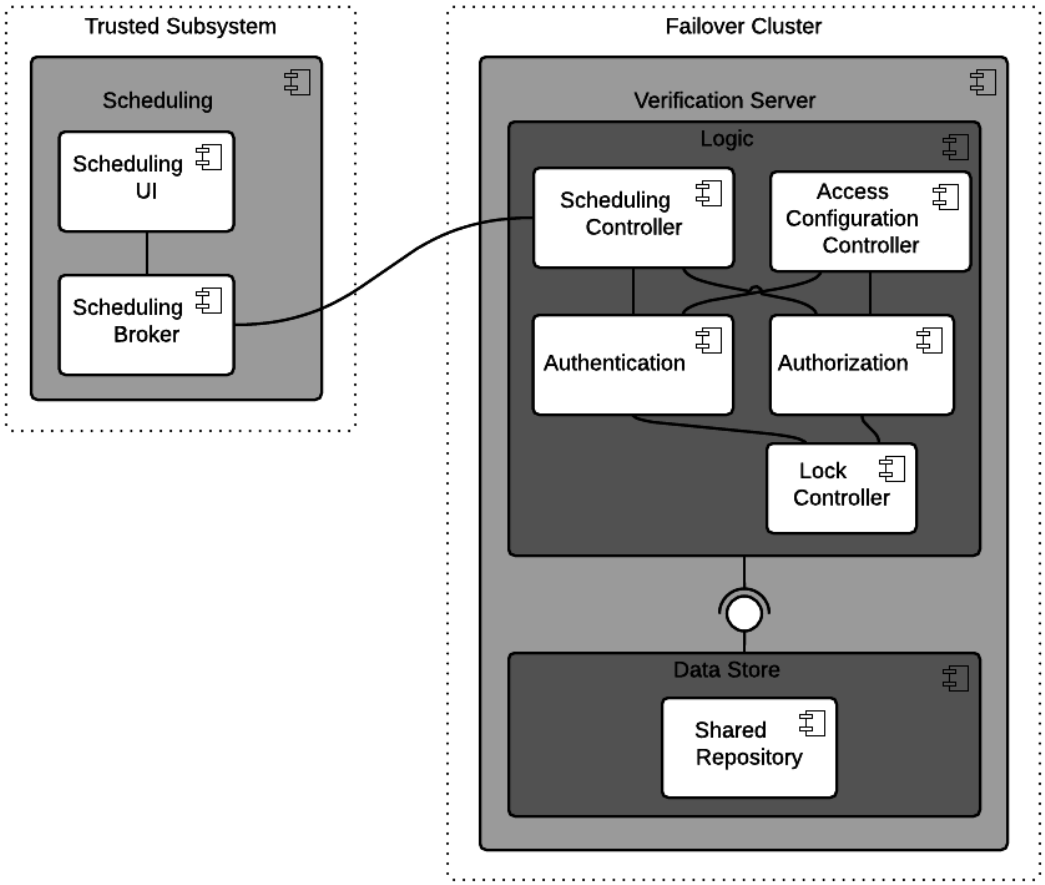

Figure 14: Trusted Subsystem

7.3.3 Layers

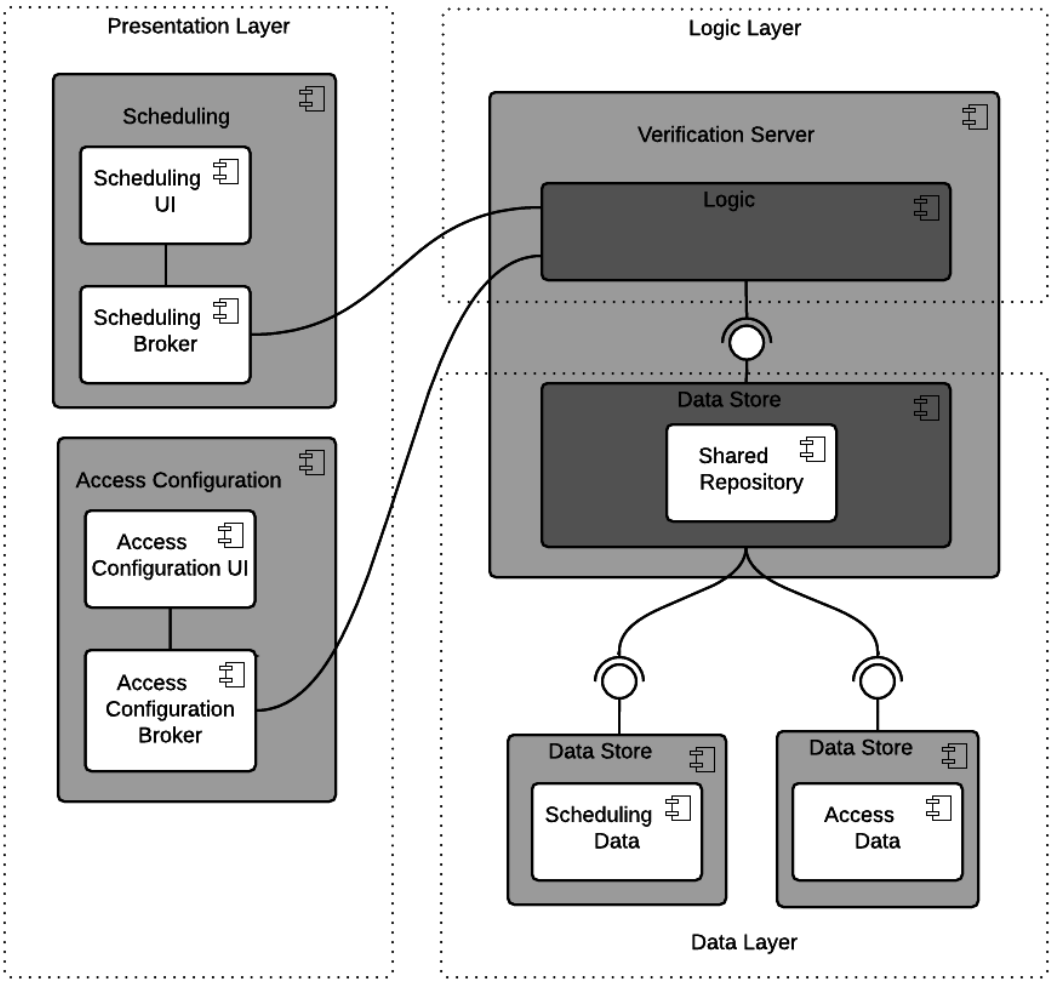

Figure 15: Layers

The Layers pattern is used to for separation of concerns in the system. The layers identified in the system are: the presentation layer, the logic layer and the data layer. By separating these groups of components a loosely coupled system is created.

The communication interface between the presentation layer and the logic layer is handled via brokers. And for the communication between the logic layer and the data layer a shared repository interface is used.

In Figure 16 the software components inside the layers are visible. The presentation layer consists of the scheduling interface to reserve rooms and the access configuration interface to manage users and locks. The logic layer holds the authentication and authorization components that decide which data is send to the presentation layer. The data layer contains the shared repository and the two data stores, the scheduling data and the access data. The shared repository decides which data store is used for read and

write operations.

7.3.4 Broker

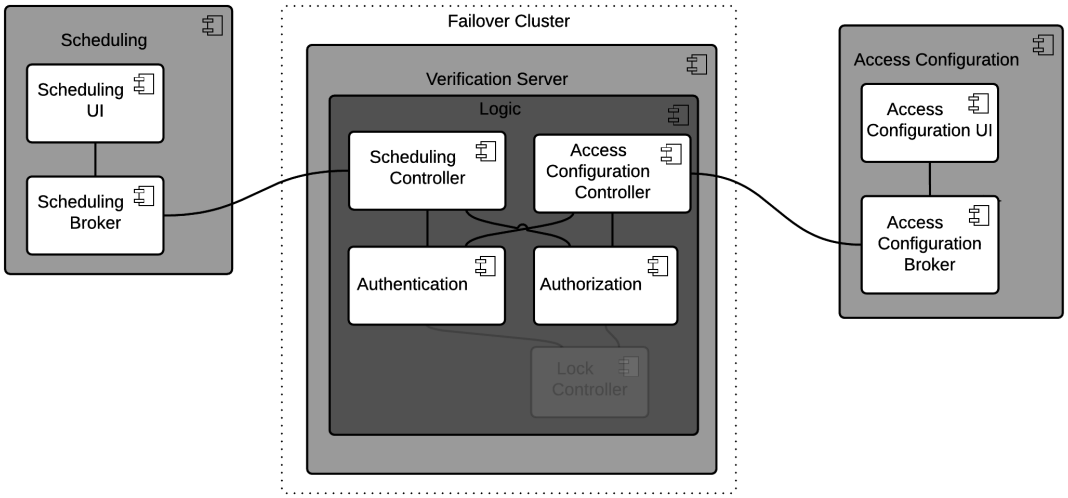

Figure 16: Layers

The system uses the Broker pattern to ensure safe communication of authentication and access data. The brokers are used to identify to which device the communication path needs to be set, and also to encrypt all of the communication.

7.3.5 Indirection Layer

8 Architecture Evaluation

(Will be included in the final deliverable)

9 System Evolution

In order to remain competitive and maintain its value for the customers, the system should keep evolving over time. This section describes the future evolution of the system.

(Will be finished in the final deliverable)

9.1 Automated Room Control

Office doors can be locked/unlocked (based on schedule for flex offices), the information about whether a room is locked/unlocked can be used to automatically turn off lighting, HVAC, etc in rooms that are locked.

9.2 Customised Room Control

With respect to personal offices or

9.3 Room Presence Detection

Use the access log to assist the detection of room presence.

9.4 Indoor Localisation

Use the access log to approximate the position of an individual within the building, or conclude that they are inside/outside the building.

9.5 Multi-Factor Authentication

When multiple sensors are present in a door, it currently means that the employees/tenants can use one of these sensors to request access to the room that the lock on the door controls. This could be expanded so that the computation unit within the lock requires that data from all sensors connected to it are obtained first and then send together to the server. The acknowledgement for unlocking the door will only be sent if all data provided is valid. This would be an implemention of Multi-Factor Authentication.

9.6 Emergency Services Integration

Once the system is implemented, the Room Presence Detection can be used to find out which rooms are the priority for emergency services who want to evacuate the building. Furthermore, Indoor Localisation can be used to give a general idea about the location of a specific person in a building.

10 References and Acknowledgements

We would like to extend our gratitude to our coach, Jorrit Idsardi, for reviewing our iterations and providing feedback on our drafts of this document, and to Paris Avgeriou for his lectures on Software Patterns.

We thank the students of Group 2 for reviewing version 0.3 of this document.
Furthermore, we refer to the following literature:

[1] Harrison, N. and Avgeriou, P. (2007). *Pattern-Driven Architectural Partitioning - Balancing Functional and Non-functional Requirements*. Retrieved from: http://www.cs.rug.nl/paris/papers/SARP07.pdf

[2] *ISO 25010*. Retrieved from Wikipedia: https://nl.wikipedia.org/wiki/ISO_25010

[3] Avgeriou, P. and Zdun, U (2005). *Architectural Patterns Revisited - A Pattern Language*. Retrieved from: http://www.cs.rug.nl/paris/papers/EPLOP05.pdf

[4] Erl, T. (2009). *SOA Design Patterns*. Prentice Hall. ISBN: 9780136135166

[5] *Enterprise Solution Patterns Using Microsoft .NET*. Retrieved from: :
https://msdn.microsoft.com/en-us/library/ff647095.aspx

[6] Ignasi U. (2014), *SWOT Table - how to convert to an article?* Retrieved from TeX - LaTeX Stack Exchange: http://tex.stackexchange.com/questions/177167/swot-table-how-to-convert-to-an-article

[7] *SWOT analysis*. Retrieved from Wikipedia:
https://en.wikipedia.org/wiki/SWOT_analysis

[8] *The London Business Footprint: The growth of serviced offices*, Deloitte Real Estate, London, 2015. Retrieved from:
https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/real-estate/deloitte-uk-london-business-footprint-serviced-offices-sector.pdf

[9] Honeywell Access Control System, retrieved from:
http://www.acdsecurity.com/commercial-a-industrial/managed-access-control.html

[10] Identicard Access Control System, retrieved from:
http://www.identicard.com/access-control/

[11] Frontier Access Control by Matrix Systems, retrieved from:
http://www.matrixsys.com/products/frontier.html

[12] Access Control by Tyco Integrated Security, retrieved from:
https://www.tycois.com/solutions-by-need/protect-my-business/access-control

[13] Access Control Security by Protection 1, retrieved from:
http://www.protection1.com/business/security-access-control-systems/

Appendices

A1: Use-case Elaboration

This appendix lists the detailed descriptions and stories of the use-cases that were introduced in Table 6 in Section 3.5.

UC-1.1 - Register a new account online

Primary Actor	User
Goal	Register a new unprivileged account on the website

Preconditions

The system management has enabled web-based user registration. The user does not yet have a user account.

Main success scenario

1. The user requests the website.
2. The user selects the register button.
3. The users fills in their name, username, password and email-address.
4. The user receives an email to verify their address and clicks the included verification link.
5. The user contacts the management to add an authentication method as per UC-3.1.

Extensions

3a. The user does not fill in every field.

3a-1. The user is prompted to complete the form.

Post conditions

The system has created a new account for the user. The new account does not have any privileges.

Related requirements	HL-1, FR-1.1, FR-1.2, FR-3.1

UC-1.2 - Register a new account

Primary Actor	Management
Goal	Register a new account on the system

Preconditions

The user for whom to create an account does not have one. Management has opened the user registration interface.

Main success scenario

1. The management fills in the name, username and email-address of the user.
2. The management allows the user to enter a password for their account.
3. The management prompts the user to add one or more authentication methods, as per UC-3.1.
4. The user receives an email to verify their email-address.

Extensions

1a. The management does not fill in every field.

1a-1. The management is prompted to complete the form.

1b. The user requires access privileges.

1b-1. The management manages user authorization as per UC-1.3.

Post conditions

The system has created a new account for the user, optionally with privileges and multiple authentication methods.

Related requirements	HL-1, FR-1.1, FR-1.3, FR-3.1

UC-1.3 - Manage user authorization

Primary Actor	Management
Goal	Manage the authorization categories for a user

Preconditions

The user has a user account. Management has opened the user management interface.

Main success scenario

1. The management selects categories for the user.
2. The management saves the changes to the system.

Extensions

1a. The user requires a new access category.

1a-1. The management adds a new category as per UC-2.3.

Post conditions

The system has changed the privileges of the user's account.

Related requirements	HL-2, FR-2.7

UC-2.1 - Access a room

Primary Actor	User
Goal	Access a room by providing authentication

Preconditions

The user has a user account. The user is present at the access point.

Main success scenario

1. The user presents the required forms of authentication.
2. The system checks that one of the user's categories is present in the room whitelist and none of the user's categories are present in the room blacklist.
3. The system checks that the room has not been booked for another user.
4. The system logs the access attempt and its result.
5. The access point unlocks.
6. The user opens the door.

Extensions

1a. The user does not possess the right forms of authentication.

1a-1. The user contacts the management and executes UC-3.1.

5a. The user does not have access privileges to the room.

5a-1. The access point does not unlock.

Post conditions

The has allowed the user entry if and only if they have access privileges and have presented the right forms of identification.

Related requirements	HL-2, FR-2.1, FR-2.2, FR-2.3, FR-2.7

UC-2.2 - Manage room authorization

Primary Actor	Management
Goal	Add whitelisted and blacklisted categories to define authorized groups of users and set locking behaviour

Preconditions

The management has opened the room management interface

Main success scenario

1. The management searches for and selects the room of which to manage authorization.
2. The management selects the categories for the whitelist of the room.
3. The management selects the categories for the blacklist of the room.
4. The management selects the locking behaviour of the room.
5. The management selects whether the room is bookable.
6. The management saves the changes to the system.

Extensions

2a. The user requires a new access category for the whitelist.

2a-1. The management adds a new category as per UC-2.3.

3a. The user requires a new access category for the blacklist.

3a-1. The management adds a new category as per UC-2.3.

Post conditions

The whitelist and blacklist categories of the room have been updated. The access point locking behaviour has been updated.

Related requirements	HL-2, FR-2.1, FR-2.2, FR-2.3, FR-2.4, FR-2.5, FR-2.8, FR-2.9, FR-4.3

UC-2.3 - Manage authorization categories

Primary Actor	Management
Goal	Add or remove categories for access permission

Preconditions

The management has opened the access management interface

Main success scenario

1. The system shows a list of categories.
2. The management enters a category name.
3. The system shows the category with a button to remove the category.

Extensions

3a. The category does not exist and the management wants to create it.

- 3a-1. The management presses 'create new category'.
- 3a-2. The system creates the new category.

3b. The management wants to remove the category.

- 3b-1. The management presses 'remove category'.
- 3b-2. The system removes the category from all users.
- 3b-3. The system removes the category from all room whitelists and blacklists.
- 3b-4. The system removes the category.

Post conditions

The category of a given name has been created, kept or removed.

Related requirements	HL-2, FR-1.4, FR-2.7, FR-2.8

UC-3.1 - Manage authentication methods

Primary Actor	Management
Goal	Manage the forms of authentication for a user (such as fingerprints or passcodes.

Preconditions

The user has an account. The management has opened the user management interface. The user is physically present and has physical identification.

Main success scenario

1. The management checks the user's physical identification (such as a passport or employee badge).
2. The management selects the user's account.
3. The management selects an authentication method.
4. The management prompts the user to set the new authentication method.

Extensions

1a. The physical identification is not correct.

1a-1. The management does not proceed with adding an authentication method.

1a-2. The management contacts security services.

4a. The authentication method is a new keycard.

4a-1. The management does not prompt the user to enter details.

4a-2. The management provides the user with a new keycard.

Post conditions

The user's account contains the new authentication method if the physical identification was successful.

Related requirements	HL-3, FR-3.2, FR-3.3

UC-4.1 - Book a room

Primary Actor	User
Goal	Book an available room for temporary access.

Preconditions

The user has a user account. The user has opened the end user interface.

Main success scenario

1. The user selects the book-a-room function in the interface.
2. The user selects a date.
3. The system shows available rooms according to the room schedule and bookability, as well as the authorization categories of the user.
4. The user selects a room.

Extensions

4a. A user want to book a room for a group of users.

4a-1. The user selects other users that will also make use of the reservation.

Post conditions

The system has reserved a room for a specific user or a group of users.

Related requirements	HL-4, FR-2.7, FR-4.1, FR-4.3, FR-4.4, FR-4.6

UC-4.2 - Change scheduling

Primary Actor	Management
Goal	Change room bookings and availability.

Preconditions

The management has opened the user management interface.

Main success scenario

1. The management selects the scheduling management function in the interface.
2. The management wants to adjust the schedule.

Extensions

2a. The management wants add a booking.

2a-1. The management select the desired room for the booking.

2a.2. The management selects a date and time.

2a.3. The management selects users for the booking.

2b. The management wants to remove a booking.

2b-1. The management select the booking.

2b-2. The management removes the booking.

2c. The management wants turn on or off the booking feature.

2b-1. The management switches on or off the booking module.

Post conditions

The management building has adjusted the booking schedule.

Related requirements	HL-4, FR-4.1, FR-4.2, FR-4.3, FR-4.6, FR-4.7

UC-5.1 - View room occupancy

Primary Actor	Management
Goal	View the estimated occupancy of rooms

Preconditions

The management has opened the user management interface.

Main success scenario

1. The management selects the occupancy overview in the interface.

Extensions

1a. The management wants the estimated occupancy of a specific room.

1a-1. The management selects the desired room from the overview.

Post conditions

The system shows a overview of the estimated occupancy per room.

Related requirements	HL-5, FR-2.6, FR-5.1, FR-5.2, FR-5.3

UC-6.1 - View historical occupancy

Primary Actor	Management
Goal	View the estimated occupancy of rooms

Preconditions

The management has opened the user management interface.

Main success scenario

1. The management selects a room or a user.
2. The management selects a date to view.

Extensions

1a. The management only want to view failed access attempts.

1a-1. The management selects a user or a room and checks a filter filtering out successful attempts .

Post conditions

The system shows a overview of historical occupancy of rooms or access attempts and their results per user.

Related requirements	HL-6, FR-2.6,FR-6.1, FR-6.2

A2: Risk Assessment

	Risk	P*	I†	S‡	Owner	Indicator	Prevention	Reaction
Business								
RB-1	Lack of customer interest	∼	+	+	Product owner	Too few orders in a fiscal year to maintain profitability	Market research (MR)	Think of other uses for product
RB-2	Too much competition to maintain profitability	∼	+	+	Product owner	Too few orders in a fiscal year to maintain profitability	MR	Increase unique features and product quality
RB-3	Too large project scope	−	∼	−	Architect	Inability to reach milestones	MR to find key features	Cut features to core requirements
Technology								
RT-1	The system incorrectly identifies a user	−	+	∼	Architect	Breaches of security after misidentification, penetration testing	Multi-factor authentication depending on security requirement	More accurate biometric scanners and/or longer passcodes
RT-2	Social engineering attack	∼	+	+	Product owner	Breaches of security after social engineering, penetration testing	Passport/ID check when managing user authentication methods, staff training	Staff training

RT-3	Remote intrusion	∼	+	+	Architect, Developer	Breaches of security after intrusion, penetration testing, intrusion detection systems	Separation of components, trusted subsystem pattern, using brokers for communication, code auditing	Code auditing, intrusion prevention systems
RT-4	User loses or forgets authentication method	+	−	∼	Product owner	User is unable to provide authentication	Reset through management after identity check	None
Implementation								
RI-1	Incompatible hardware components	∼	+	+	Architect	Components can not work together	Research compatible components	Change hardware components for compatibility
RI-2	Wrong time estimation for development	∼	∼	∼	Project manager	Inability to reach milestones	Spend time on planning, limit project or milestone scope	Cut features to reduce workload
RI-3	Change of requirements	+	∼	+	Project manager	Inability to complete the system	Spend time on planning, define deadlines for project and set milestones	Delay launch, investigate impact of a requirement change
Operational								
RO-1	System servers go down	∼	+	+	Architect	Communication failures to servers	System redundancy	Further redundancy, code and infrastructure auditing

RO-2	Power outage	–	∼	–	Product owner	Lock mechanisms no longer functioning	Use fail-secure locking to prevent unauthorised entry	None

∗) Probability †) Impact ‡) Severity

Table 12: Risk severity

A3: Time Tracking

Week 1

ID	Date	Task	Hours	Sum
TH	11/11/15	Setting up LaTeX environment and elaborate template document	3	**3.5**
	11/15/15	Idea generation and checking text	0.5	
TA	15/11/15	Idea generation and checking text	1	**1**
MM	15/11/15	Idea generation and checking text	0.5	**0.5**
JM	14/11/15	First idea generation	1	**1**
			Total:	**6**

Week 2

ID	Date	Task	Hours	Sum
TH	18/11/15	Brainstorming	3	**15**
	19/11/15	Small modifications to template	0.5	
	20/11/15	Research on competitor systems	1	
	21/11/15	Discussion	0.5	
	21/11/15	Research on types of access points and interfaces	1.5	
	21/11/15	Reading and reviewing sections 1-4	1.5	
	21/11/15	Writing section 1-2	1.5	
	22/11/15	Reading PDAP literature	1	
	22/11/15	Writing section 1-2	1	
	22/11/15	Refining section 3-4	3	
	22/11/15	Discussion on document structure and system scope	0.5	
MK	16/11/15	Feedback and Q&A	1	**9**
	18/11/15	Brainstorming	3	
	19/11/15	Identifying applicable patterns	2	
	20/11/15	Writing first 3 architectural decisions	3	
TA	16/11/15	Feedback and Q&A	1	**14**
	18/11/15	Brainstorming	3	
	21/11/15	Proofreading, suggestions and corrections. Writing sections 3.4 high-level requirements and 3.5 use-cases	5	
	22/11/15	Reading PDAP literature, writing section 3.6 functional requirements, bibliography, reworking section 3.5 use-cases, proofreading and correcting sections 1 and 4.1	5	
MM	16/11/15	Feedback and Q&A	1	**16**
	18/11/15	Brainstorming	3	
	19/11/15	Identifying applicable patterns	2	

	21/11/15	Literature Study	2	
	21/11/15	Analysis section	4	
	22/11/15	Literature Study	1	
	22/11/15	Analysis section	2	
	22/11/15	Reviewing	1	
JM	16/11/15	Feedback and Q&A	1	**13**
	18/11/15	Brainstorming	3	
	18/11/15	Literature study	1	
	19/11/15	Stakeholders and key-drivers section	3	
	21/11/15	Reviewing and corrections chapter 1,3	2	
	22/11/15	Non-functional requirements, reviewing and correction chapter 4	3	
			Total:	**67**

Week 3

ID	Date	Task	Hours	Sum
TH	23/11/15	Feedback and Q&A	0.5	**18**
	24/11/15	Meeting	2	
	25/11/15	Meeting	4	
	28/11/15	Refining section 2	2.5	
	28/11/15	Reviewing/refining section 3	2.5	
	28/11/15	Discussion	0.5	
	29/11/15	Discussion	1.5	
	29/11/15	Research on competition, sensor types and computation unit	1	
	29/11/15	Reviewing/refining section 4	1.5	
	29/11/15	Reviewing section 5-6	0.5	
	29/11/15	Reviewing/refining/writing section 7	1	
	29/11/15	Small draft of evolution ideas in section 9	0.5	
MK	24/11/15	Meeting	2	**16**
	25/11/15	Meeting	5	
	28/11/15	Writing section 5.1	2	
	28/11/15	Writing section 5.2	1	
	29/11/15	Research on authentication methods	1	
	29/11/15	Updating section 5	3	
	29/11/15	Working on section 7	2	
TA	23/11/15	Feedback and Q&A	0.5	**18**
	24/11/15	Update requirements regarding presence detection, start elaborating use-cases	2	
	25/11/15	Work on A1 use-cases, 3.6 functional requirements	5.5	
	26/11/15	Work on A2 risk assessment	2	
	28/11/15	Add more related requirements to Patterns in 4.1 Analysis, amend section 9, start presentation	3	

	29/11/15	Work on section 7 software architecture, proof-reading sections 5 and 6	5	
MM	23/11/15	Feedback and Q&A	0.5	**18**
	24/11/15	Meeting	2	
	25/11/15	Meeting	5	
	27/11/15	System Architecture	1.5	
	29/11/15	Analysis	3	
	29/11/15	System Architecture	1	
	29/11/15	Software Architecture	4	
	29/11/15	Review	1	
JM	23/11/15	Feedback and Q&A	0.5	**16.5**
	24/11/15	Meeting, key-drivers and SH corrections	2	
	25/11/15	Work on use-cases and corrections	4.5	
	28/11/15	Review risks and adding RI-3	0.5	
	29/11/15	Writing section 6 and 7.2.3	6	
	29/11/15	Reviewing and corrections section 1-7	2	
	29/11/15	Adding related requirements in section 4	1	
			Total:	**86.5**

Week 4

ID	Date	Task	Hours	Sum
TH	00/00/15	...	...	
	00/00/15	...	...	
MK	00/00/15	...	...	
	00/00/15	...	...	
TA	00/00/15	...	...	
	00/00/15	...	...	
MM	00/00/15	...	...	
	00/00/15	...	...	
JM	00/00/15	...	...	
	00/00/15	...	...	
			Total:	**...**

Total logged time

ID	Author	Hours
TH	Thomas Hoeksema	36.5
MK	Maarten Kollenstart	25
TA	Toon Albers	33
MM	Michel Medema	34.5
JM	Jasper Mohlmann	30.5
	Total:	**159.5**