DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Information Systems

Designing a data access control concept for the Knowledge4Retail platform

Kilian Dresse

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Information Systems

Designing a data access control concept for the Knowledge4Retail platform

Entwicklung eines Datenzugriffskontrollkonzepts für die Knowledge4Retail-Plattform

Author:	Kilian Dresse
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Tim Schopf
Submission Date:	16.05.2022

I confirm that this bachelor’s thesis in information systems is my own work and I have documented all sources and material used.

Kilian Dresse

Munich, 16.05.2022 Kilian Dresse

Acknowledgments

To begin, I would like to show great appreciation to my advisor Tim Schopf who always supported me throughout my bachelor's thesis.

I also want to thank Prof. Dr. Florian Matthes for giving me the opportunity to write my thesis at his chair Software Engineering for Business Information Systems (SEBIS).

Furthermore, I would like to thank all my interview partners for their time without whom this research would not have been possible.

Last but not least, I want to thank my family and friends who supported me during the past five months.

Abstract

Knowledge4Retail (K4R) is an interdisciplinary research project set out to digitalize the stationary retail business. A number of different partners are collaborating to optimize two key factors of retail. First, the optimization of product placement inside the store to accelerate turnover and create a more personal shopping experience. Secondly, the process of optimizing and automating store internal logistics, from the arrival of new deliveries to the refilling of shelves. To achieve these goals various use cases were created, which are developed by the partners of the project.

The platform operates around a digital twin, which is a digital replica of a physical object and in this case a retail store. This dynamic database needs to be accessible by the use cases and components to fulfill their objective. The multitude of components accessing the diverse resources has to be controlled to assure information confidentiality and integrity. To enhance the platform's security, data access control needs to be implemented.

This research analyzes the data flows of the various components of the system to create a foundation for a data access control concept for the K4R platform. The research's insights are largely based on expert interviews, which were conducted to analyze the exact definitions of every component of the system as well as what the individual components of the project need to access. Furthermore, the focus lies on how employees of the supermarket should be able to access the digital twin's information. This research uses role-based access control to define access roles for each component in the system based on their activity and what they should be allowed to access.

In the process, a UML component diagram is created to resemble the platform's data flows as well as a UML class diagram to analyze the different resources of the digital twin. Combining the results from the diagrams, the roles are established, creating the desired elements of a data access control concept. The proposed elements build the foundation of a future implementation for access control on the K4R platform. Through this, a potential methodology for the creation of a role-based access control system for a complex interdisciplinary platform such as K4R is established.

In addition, the expert interviews are used to explore different hosting styles for the K4R system as well as the possibility of external parties accessing information of a supermarket's K4R instance.

Kurzfassung

Knowledge4Retail (K4R) ist ein interdisziplinäres Forschungsprojekt, das den stationären Handel weiter digitalisieren will. Verschiedene Partner arbeiten zusammen, um zwei Erfolgsfaktoren des Einzelhandels zu optimieren. Einerseits geht es um die Optimierung der Produktplatzierung im Laden, um den Umsatz zu steigern und ein kundenorientierteres Einkaufserlebnis zu schaffen. Andererseits gilt es die Intralogistik, von der Ankunft neuer Lieferungen bis zum Auffüllen der Regale, zu optimieren und zu automatisieren. Um diese Ziele zu erreichen, wurden verschiedene Anwendungsfälle geschaffen, die von den Partnern des Projekts entwickelt werden.

Die Plattform basiert auf einem digitalen Zwilling, der ein virtuelles Abbild eines physischen Objekts ist, in diesem Fall eines Einzelhandelsgeschäfts. Auf diese dynamische Datenbank müssen die Anwendungsfälle und Komponenten zugreifen können, um ihre Aufgaben zu erfüllen. Die Vielzahl der Komponenten, die auf die verschiedenen Ressourcen zugreifen, muss kontrolliert und die Vertrauenswürdigkeit und Integrität der Informationen müssen gewährleistet werden. Zudem ist eine Datenzugriffskontrolle zu implementieren, um die Sicherheit der Plattform zu erhöhen.

Diese Studie analysiert die Datenflüsse der verschiedenen Komponenten des Systems und hat das Ziel, die Grundlage für ein Datenzugriffskontrollkonzept für die K4R-Plattform zu schaffen. Dabei wird die Studie unterstützt durch Experteninterviews, die durchgeführt wurden, um die genauen Definitionen jeder Komponente des Systems zu analysieren und zu ermitteln, worauf die einzelnen Komponenten des Projekts zugreifen müssen. Zusätzlich liegt der Fokus darauf, wie die Mitarbeiter des Supermarktes auf die Informationen des digitalen Zwillings zugreifen können. Hierfür wird eine rollenbasierte Zugriffskontrolle verwendet, damit die Rollen für jede Komponente im System definiert sind.

In der Arbeit wird ein UML-Komponentendiagramm erstellt, um die Datenflüsse der Plattform darzustellen, sowie ein UML-Klassendiagramm, um die verschiedenen Ressourcen des digitalen Zwillings zu analysieren. Durch die Kombination der Ergebnisse aus den Diagrammen werden die Rollen festgelegt, wodurch die gewünschten Elemente eines Datenzugriffskontrollkonzepts entstehen. Die vorgeschlagenen Elemente bilden die Grundlage für eine zukünftige Implementierung der Zugriffskontrolle auf der K4R-Plattform. Dadurch entsteht außerdem eine mögliche Methodik für die Erstellung eines rollenbasierten Zugriffskontrollsystems für eine komplexe interdisziplinäre Plattform wie K4R.

Zusätzlich werden die Experteninterviews genutzt, um verschiedene Hosting-Optionen für das K4R-System zu untersuchen, sowie die Möglichkeit des Zugriffs externer Parteien auf Informationen der K4R-Instanz eines Supermarktes zu analysieren.

Contents

1. Introduction

1.1. Motivation

The revenue and popularity of online retail are continuously rising, while stationary retail is left behind [Hei17]. Due to changed customer behavior, online retailers are experiencing growth rates of up to 9.5%, while stationary retailers stand at 1.5% [Böt+21]. Knowledge4Retail (K4R) is a project "made to revolutionize the stationary retail business" [K4R20]. A number of partners from different areas of business joined forces to form the project. To "revolutionize" stationary retail, merchants need to evaluate new ways to improve value propositions and expand their services [KT+19; Böt+21].

Together, the goal is to create an open-source platform, which uses artificial intelligence to connect the digital world with stationary retail. The interdisciplinary research project focuses mainly on two success factors with the help of use cases to develop new and modern applications for retail. First, it concentrates on an optimized placement of products in a store through the use case of **strategic retail marketing**, which is supported by artificial intelligence. Secondly, the focus lies on the improved availability of products in a shelf, by improving the internal logistics of a store, from the arrival of new deliveries to the refilling of shelves. This is done with the help of the use case of **intelligent intralogistic**. The improved management of logistics is complemented by real-time tracking of item take-out with the use case of **intelligent refrigerators**. In addition, the use cases are enhanced by **service robotics**, which provide various assistance for both the customer side and the corporate side of retail.

The project has funding of 13 million euros through the German Federal Ministry of Economic Affairs and Climate Action [Ger22] as well as the leading partner: team neusta, which is a software service provider specialized in the development of complex software, mobile, and e-commerce solutions [Gmb22b]. Relevant partners include the Technical University of Munich, the German Research Center for Artificial Intelligence (DFKI), and the retail partner "dm-drogerie markt" (dm) with his subsidiary company dmTECH, as well as the German IT service provider, nagarro.

At the heart of Knowledge4Retail lies a digital twin. A digital replication of a physical object in the real world [GV17]. In the case of K4R, the digital twin replicates an instance of a supermarket facility, storing amongst other things, the precise locations of all placed products in their respective shelves. It needs to communicate with the various use cases along with other components of the system, such as the ERP[1] software or staff members.

At the moment, the communication between the various actors of the K4R platform is not secured in any way. Creating a secure platform is crucial to ensure that companies and their

[1] Enterprise Resource Planning

assets are protected [Eck18]. A high-security standard is partially derived from controlling the access to the data of the digital twin to ensure information integrity and confidentiality [SK07]. Therefore, to enhance the platform's security, data access control needs to be implemented.

Access control serves as the administrative and automated process to determine what kind of system operations on which system resources, a user or system is allowed to make [Hu+15]. For the K4R project, this translates to what all the different components of the system must be authorized to access on the centralized database, the digital twin.

The interdisciplinary research project, K4R is made up of a vast number of different organizations. Accordingly, the fundamental process of this research lies in the introduction of access control to manage the needed access from all different parties. To accomplish this, it needs to be established what each partner expects and desires from the platform. Furthermore, apart from analyzing how the different use cases and components interact with the system, we also examine how employees of the supermarket should be able to access the digital twin's information.

1.2. Research Design

This research touches on the conceptualization of data access control for Knowledge4Retail, providing the groundwork for future research and implementation. To begin, the data traffic between the different players of the system has to be elaborated. A crucial factor to differentiate who may communicate with whom and how. Hence, the goal of this research lies in the creation of a data access control concept, from the analysis of the information flow to the definition and assignment of access rights. Accordingly, the research is separated into three main research questions that need to be addressed during the course of this research.

RQ1 How to model data flows within the Knowledge4Retail platform for associated organizations and roles?

The first of three research questions focuses on understanding the data traffic between the various components involved in the K4R platform. This includes the previously described organizations taking part in the K4R project as well as the use cases they develop. For all of these organizations, roles need to be declared. These roles define their activities on the K4R platform.

To model the data flows between these organizations and roles, we first need to clearly define the different organizations and roles of the system. After which, an overview of the system may be created, illustrating the relationship between each respective party. The model is created by using the foundations of the project from the documentation of K4R, provided by Confluence[2]. The overview is then displayed using a component diagram. The created model is further enhanced by the insights coming from the second research objective and its corresponding research question.

RQ2 What are requirements of retailers and partner organizations for a Knowledge4Retail data access control concept?

[2] A workspace for teams to share and collaborate on documentations

With the second research question, we concentrate on the requirements of the involved parties for the final concept regarding to how they would like their respective components (like the various presented use cases) to be integrated into the system.

In this part of the research semi-structured interviews are used. They function to determine the needs and nonessentials of partners for the platform. The interviews focus on the development of requirements in the access control concept for roles and organizations. This includes a revision of the component architecture created through **RQ1** and suggestions on the final list of roles as well. Additionally, the research objective tackles the question of how the platform could be hosted, specifically whether on-premise or cloud is the best suited as well as how third-party organizations might be able to play a role on the platform.

With the collected insights from the semi-structured interviews, a concept for data access control can be created.

RQ3 How to design elements of a data access control concept for the Knowledge4Retail platform?

There are various techniques and models of access control. For this kind of system, role-based access control (RBAC) is the most sufficient type. The last third of the thesis combines the collected information from the previous two research objectives to create various elements of a role-based access control concept. It focuses on the establishment of roles in the concept as well as propositions for the access rights of the individual roles.

Lastly, the created concept needs to be evaluated, resulting in another conduction of semi-constructed interviews. The interviewees of the second round use their expertise to validate the final results of this research.

1.3. Structure

In this section, we provide an overview of the structure of the thesis. Figure 1.1 showcases how the different research questions intertwine to result in the creation of the data access control concept for the Knowledge4Retail platform.

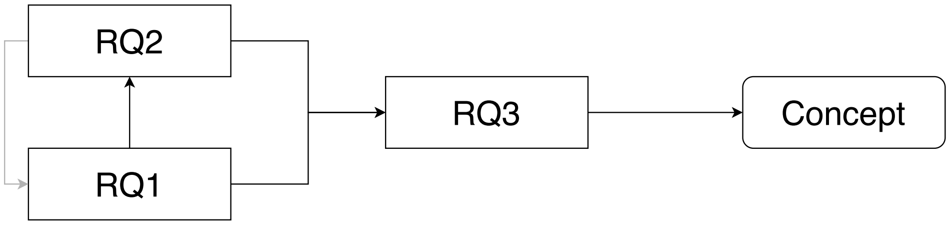

Figure 1.1.: Flow of knowledge between the research questions

While initial results from **RQ1**'s data flow and role definitions provide the base for **RQ2**'s interviews, only with the results from said interviews the **RQ1**'s research objective can be finalized. With **RQ1** and **RQ2** resolved, their knowledge is used to answer **RQ3** and as a result create a concept for data access control.

To start off, the foundations and related works necessary for the thesis are introduced in chapter 2. It describes all necessary background for the research regarding IT security and access control, but also the digital twin and the K4R project. Marking the beginning of the main part, we take a deep dive into the methodology in chapter 3. While the research design gives a quick glance at the structure of the thesis, this Chapter focuses on the used methodology to create the access control concept. This also includes the first approach at **RQ1** to prepare material for the interviews. In the following, **RQ2** is answered by presenting the results from the interviews in chapter 4, split up into the respective sections. Subsequently, in chapter 5, the results from the interviews are used to finalize **RQ1** by deciding on the list of actors and defining the data flow in the system. After which, elements of an access control concept are created. Finally, we take a critical look at the created concept in the discussion in chapter 6. During this, we debate the problems and limitations of the approach with the help of the outcome of the evaluation interviews. Last but not least, we conclude the thesis by giving an overview of its results and providing suggestions for future work in chapter 7.

2. Foundations and Related Work

To provide a basis for this research, this chapter introduces the necessary background needed. This process is divided into four sections. We first establish the necessary terminology for access control, followed by presenting the relevant access control models for the research. Furthermore, foundations on digital twins and the K4R project itself are described.

2.1. Terminology of an Access Control Model

To design a data access control concept for the Knowledge4Retail platform we, first of all, need to define what makes such a concept. For that, we take a look at IT-Security as a whole and define the terminology regarding an access control model.

2.1.1. IT Security

IT security is essential to ensure that companies and their assets are protected, preventing economic damage that can result from breaches of confidentiality, manipulation, or even disruptions to the availability of company services [Eck18]. In IT security we need to differentiate between safety and security for an IT system. It is safe when it can provide its expected functionality, which means that the actual functionality is in line with the expected. An IT system is secure if it can protect its saved and managed information from illegitimate access by unauthorized users [Eck18]. In general, an IT system is a computer, software, or any sort of system that can process data or communicate as a functional unit on which security measures need to be implemented to ensure its integrity [Die04]. These systems are usually made up of multiple functional units, hence more IT systems.

According to [Die04], an IT system has two complementary views for security. The first of which is defined as dependability and is applicable *inside* the system. It describes the state where one can be sure that his/her processed data is not restricted in its availability in any way. The second view, controllability, ensures that a user *in front* of a system is never restricted from accessing the IT system. This concept by Dierstein is called the "dual security" concept.

There are several areas regarding the security of an IT system that each have a specific protection goal through which the security in that area can be realized. In practice, there will always remain a risk. The entire security of an IT system cannot be guaranteed with regards to the works of a protection goal, since humans as a functional unit of an IT system cannot be classified as reliable, but are rather considered to be uncontrollable [Die04].

Furthermore, an IT system is categorized into five protection goals. The protection goals are confidentiality, integrity, availability, non-repudiation, and authenticity [Eck18; SK07]:

- Confidentiality demands the inability of unauthorized users to access confidential information and therefore prevents read access to the system.
- Integrity is met when the *write* access on a system, still assures that stored information of a system can be identified as legitimate and is protected.
- Availability is present when authenticated and authorized users cannot be restricted in their allowed entitlements in any way (through for example performance issues of the server).
- Non-repudiation requires that a user cannot deny his/her actions after its performance (mostly needed in e-commerce).
- Authenticity means for a functional unit to be real and credible as well as verifiable based on unique characteristic properties such as a login.

The most relevant of the five protection goals is the confidentiality and integrity for the research around data access control. As [Die04] defined these two goals confidentiality and integrity fall under dependability alongside the earlier mentioned availability. In the following Figure 2.1, we can see an overview of the presented areas in IT security.

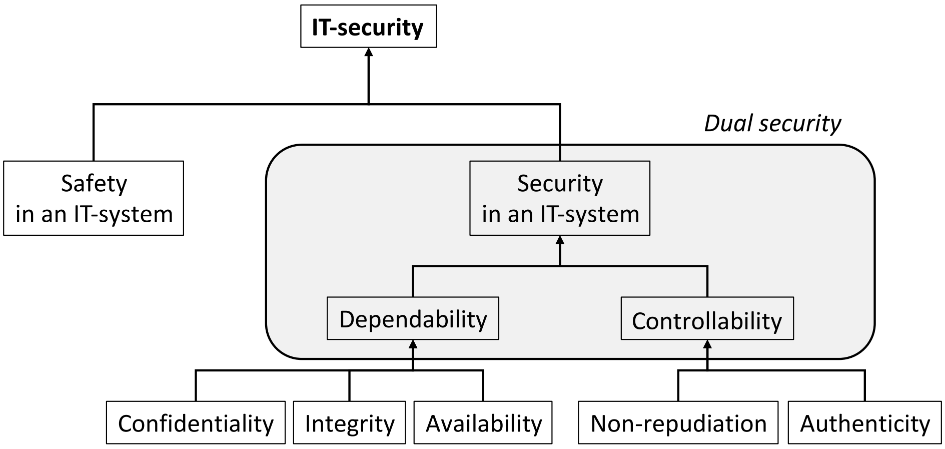

Figure 2.1.: Overview of the different areas in IT security

The Knowledge4Retail project is based primarily on application systems that can be accessed through interfaces by other functional units or components (subsection 2.4.2). These serve as gateway points to confidential data. An application system does not only represent the application itself, it is built upon numerous information system layers: hardware, operating system, middleware, and potentially pre-existing infrastructures. A layering system is built so that a layer may access methods and interfaces only from the layer beneath to fulfill its task, which makes it a strictly layered architecture to ensure easy maintenance for secure access to the system [Bus+02].

To create a data access control concept for K4R, the most important *basic security function* is access control (subsection 2.1.2), which goes hand in hand with permission management (subsection 2.1.3). Together they meet the protection goals for confidentiality and integrity.

2.1.2. Access Control

Access control serves as the administrative and automated process to determine what kind of system operations on which system resources, a principal is allowed to make. The goal of access control lies in the implementation of policies as rule sets, made to guard the resources of the system [Hu+15]. First of all, we need to define the following, different terms used in access control: principal, resource, and access.

A principal is an entity of the system and has a distinct identity. With this identity, it can get the correct access rights assigned to it [Leh07]. A principal can be a person, but also a process, machine, or any communication tool [And20]. However, the correct terminology of the principal is imprecise in literature. Lampson is referring to it as the subject [Lam69]. Gollmann however, defines the principal as the acting role, while the subject relates to the process which is accessing the resources [Gol10]. Going forward, we will refer to Lampson's definition since there is no need for further differentiation in this thesis. Hence subject as well as the acting principal will be referred to as a subject.

The resources accessed by subjects can be defined as any entity of the IT system that can be operated on. Another term for a resource is an object. However, some objects may also act as subjects, since a resource can be an executable program of the system which also accesses other objects (or resources) of the system for example [Leh07].

Accessing an object can be either reading or writing. A reading access does not change the system's state, whereas a writing access does just that. Writing can be anything from editing an object, deleting it entirely, or creating a new one [Leh07].

Nowadays access control is implemented all across the IT infrastructure and its different layers. It is used in operating systems for file and folder protection, in applications to restrict access to confidential information like payroll processing or health benefits management, and in databases to manage access to tables, records, and fields (like the semantic digital twin, subsection 2.3.1) [Hu+15].

When talking about access control, according to Gollmann, the protection goal can be split into three main tasks [Gol10].

- The **authentication** can be interpreted as the right to enter through a request. This is usually done by the subject, asking to access information. It describes the process of a subject's identity being validated (or authenticated) [Leh07].
- The **authorization** serves as the evaluation of the request from the subject, checking whether the implemented policies are not violated.
- And the **setting** of security policies.

To fully understand the concept of security policies and how to establish them properly, additional focus is set on permission management.

2.1.3. Permission Management

Permission management is about assigning access rights for every subject to every access-controlled resource of the system [Leh07]. Additionally, it is used to ensure the integrity of the system (subsection 2.1.1).

An operation system, for example, defines read and write permissions on every file. Permission management assigns and alters access rights as well as defines under what circumstances those rights may be used. For example at what time of day these rights may be used or depending on which authentication method they used to authorize their identity [Leh07].

Saltzer et al. defined a few essential principles for permission management [SS75]. First is the **complete mediation** principle, stating that every single access request by a subject needs to be validated on its permission status. Following this principle can become quite overwhelming, which is why typically one of the complementary principles is implemented. These are either the whitelist or blacklist principles.

A whitelist describes a list where all access is prohibited unless it is explicitly allowed and a blacklist is the opposite.

The blacklist principle is easier to implement since every single access right does not have to be defined but it has dangerous design flaws. When the blacklist is not designed thoroughly a subject may end up with more access rights than intended.

With a whitelist, on the other hand, there can only be too few rights given, which can easily be fixed without causing a security issue, making it the more secure choice, also known as the **fail-safe default** principle [SS75].

The next principle goes hand in hand with the fail-safe default principle. The **least privilege** principle also known as the *need to know* principle, which demands that a subject should always have the bare minimum amount of access rights needed to fulfill its task [SS75].

Lastly, we have the **separation of privilege** also known as the **separation of duty** principle. It defines that where needed permission should be separated between different subjects, to minimize the amount of control or power a user should have. One example of this is the concept of a banker. While he/she has the right to transfer money to others, he/she should not be able to do so with his/her account [Leh07; SS75]. Hence the restriction is needed to avoid a subject's fraud by carrying out conflicting activities [Jos+05].

In conclusion, permission management aims to let subjects perform their work properly without having overly harsh restrictions, but also not the opposite, which could create harm through intentionally or inadvertently comprising resources of the system [Leh07].

2.2. Access Control Models for Business Application Systems

The permission management in an access control model can be implemented through several different models, made for different platforms in the IT landscape. Through the course of this section, we work our way theoretically and historically towards the foundations of role-based access control. In 1969, Lampson introduced the first permission models as an access control matrix [Lam69]. For these matrixes, different options of access rights may be used. Table 2.1 demonstrates the different rights that will be used in the following models [Eck18].

Read-only	Solely allows reading access
Append	Allows adding data to an existing object
Execute	Allows the execution of executional objects
Read-write	Allows full access to a resource, hence both read and write permissions
Control	Allows the forwarding as well as withdrawal of rights for other subjects

Table 2.1.: Access rights used in different models [Eck18]

Then the access control matrixes can be implemented into access control lists (ACL) with an access control matrix on a column and its corresponding object [And20]. An example of this can be seen in Table 2.2. It displays what kind of access each subject is given, with the simplest use of access rights: "r" standing for *read-only* and "rw", for *read-write*.

User	Accounting Data
Sam	rw
Alice	rw
Bob	r

Table 2.2.: Example of an access control list [And20]

This model is referred to as identity-based access control (IBAC) since the access rights are defined according to the principal's identity [Mol08]. Because of its lack of complexity, IBAC is easy to grasp and understand. It is mainly used in environments, where principals need to manage their own file security [And20]. However, when it comes to more complex systems like an intertwined corporate landscape, identity-based access control is easily overwhelmed.

In IBAC, every object needs its own ACL and every subject would need its own identifier in each list [Mol08]. It is a very static approach that is not open to changes, when a subject would enter a different *role* in the company, every ACL would need to be altered to its new access rights. Therefore, to properly manage access rights in a project like Knowledge4Retail, we need more robust forms of access control.

2.2.1. Bell-LaPulda Model

The Bell and LaPulda model was designed for the military and is used to classify information depending on its level of security clearance. The model is based on a dynamic access control matrix with the access rights defined as the universal rights apart from just read and write, but using all of the options defined in Table 2.1: *read-only, append, execute, read-write, and control* [Eck18; BL73].

On top of the dynamic access control matrix, each subject and object is given a security level and only a subject with the same or higher level of security clearance is able to read the information, referred to as the *no-read-up* rule [Hu+15; BL73].

2.2.2. Discretionary Access Control

Discretionary access control (DAC) lets the subject, and in this case, normally a physical user, decide on the assignment of access rights and therefore has full *control* over the object. The main aspect of the model is the ownership of an object. While IT system's global security settings may not be altered, security settings involving objects owned by the user may be. It can be determined without oversight by other policies or administrative subjects [Eck18; Hu+15]. In general, the involved access rights only include *read-only* and *read-write*, but other access rights may be included like *control* to delegate the ownership to other subjects or *execute*, which is added in the most commonly used instance of DAC, the operating system UNIX [Leh07].

Discretionary access control is one of the simplest approaches for access control of personal data, which finds its uses on lower levels in business applications.

2.2.3. Role-based Access Control

RBAC is the most commonly used access control model in business applications [Leh07]. It is used in almost any ERP or content management system to provide access control because of its good performance in distributed systems [Eck18].

The concept of role-based access control (RBAC) was first used in the 1970s and 1980s by various institutions, but it was not until 1992 that a model was formerly generalized by Ferraiolo and Kuhn [Hu+15; FKC03]. In RBAC, access rights are defined as roles and each role describes a specific job or field of activity in a company, that holds a set of permissions. Unlike other access control models, RBAC does not focus on the subjects of the system, but instead on the specific jobs [Eck18]. Hence, access rights are assigned to the roles, depending on and tailored towards the field of activity. Each subject is then assigned to at least one role in the system to perform a specific job [FKC03]. In the year 2000, Sandhu built on the work of Ferraiolo and Kuhn to develop a standard for RBAC [SFK+00], which resulted in its addition to the NIST[1]. The resulting NIST RBAC model consists of four components [SFK+00; Jos+05].

- A finite set of `Users`, which is equal to our definition of subjects, including both the principal and any acting process
- A finite set of `Rules`, with one rule including a set of permissions
- A finite set of `Permissions`, equal to the universal access rights defined in Table 2.1
- And a finite set of `Sessions`, where one session refers to a subject's current interaction with the system

[1] National Institute of Standards and Technology

The NIST RBAC is separated into four levels of capabilities, in order to only use the capabilities needed for the IT system, one should always take the lowest level that satisfies one's interest. The first level is the Flat RBAC, which includes the basic functionality showcased above.

Secondly, we have the hierarchical RBAC. At this level, the different jobs are arranged in a hierarchical manner similar to corporate organizations. This adds the ability for senior roles to inherit the set of permission from junior roles beneath it. An example of a hierarchical RBAC can be seen in Figure 2.2. The example by Eckert showcases the relationship between different employees in the banking sector as well as the separated role of a customer. We can see how, for example, the branch manager inherits all the roles and therefore has all available permissions in the model, and every employee in the branch is given a general permission set in form of the `Employee`, regardless of their job title.

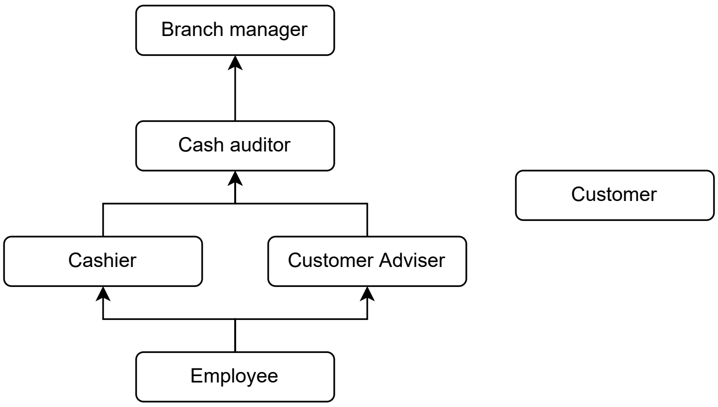

Figure 2.2.: Example of an arrangement of roles in hierarchical RCAB in the banking sector [Eck18]

However, the hierarchical approach can become problematic when a senior role has *all* the rights of a *junior* role, and hence having access to objects that it does not need for its current task, which brings us to the third level of RBAC.

Constrained RBAC keeps a role's capabilities under control. The constraints are introduced to conform to the separation of duty (subsection 2.1.3), meant to prevent a subject from executing two contradicting tasks. This is known as the static separation of duty. Continuing with the banking example, a reasonable constraint would be to make it impossible for an employee to take the role of a customer and therefore full control over his bank account [Eck18]. The dynamic separation of duty would also consider session changes, meaning that it is not strictly forbidden that a subject has two opposing roles, just not in the same session, therefore adding the time constraint to the evaluation of the subject's access rights.

Lastly, the fourth level is defined as the symmetric RBAC, which adds permission-role reviews. This addition is not relevant to the means of this thesis, which is why we will not go into more detail.

Those four levels of role-based access control are added on top of each other, thus concerning this thesis, the previously described decision at which level all needs of a system would be satisfied is required.

2.2.4. Attribute-based Access Control

Role-based access control is the standard for incorporating access control in a company [Leh07; Eck18]. However, nowadays IT systems and their enterprises have become more interconnected and distributed, which is why Hu et al. presented, once more a NIST-standard for the "latest development in an evolution of authorization processes" in 2013 with the attribute-based access control (ABAC) [Hu+13; Hu+15].

Instead of declaring roles that need to include all subjects of a system, ABAC declares attributes, which every subject and object of a system holds alongside environmental factors that may affect the permission assignment. A subject of the system may have attributes describing its job title, task description, or seniority level, whereas objects hold attributes for their type, creator, and confidentiality level of the object. Environmental factors may include from where or when a subject is requesting access.

On behalf of the attributes with which a subject requests access to an object, the system dynamically determines its individual set of access rights. In consequence, this makes the system a lot more dynamic, having individually designed access rights instead of a static set of roles in an RBAC model [Hu+15].

However, the model has a few problems, which make it difficult to incorporate it into the thesis' scope. Primarily, there are not enough examples of it being incorporated into large corporations, as well as some oversight problems that come with the extremely dynamic approach. These come from the lack of clear hierarchical options or clear audibility, on who is accessing what information [SO17]. In literature, those problems were explored by creating hybrid attribute-based access control models, where the attributes are used to simply assign the roles for an RBAC model [Cru+08; WSS10; SO17].

2.3. Digital Twin

The concept of a digital twin first emerged in the year 2003, when Micheal Grieves and John Vickers presented the idea that a virtual model could provide the foundations for product life-cycle management [GV17]. At the time, however, the technology still lacked the needed technological advancements. Recent advancements in related technology fields such as the Internet of Things (IoT), big data, Industry 4.0, real-time sensors, and sensor networks [Tao+19; Jon+20] have enabled derivation leading to the increased recognition of the digital twin concept. This eventually put the digital twin on Gartner's key strategic technology trends for the years 2017 to 2019 [Gar18]. Consequently, the concept has become an attractive subject for academia and industry to research and develop [Jon+20].

A digital twin can be defined as a digital representation of a product that consists of information about said product. Products in this context can be tangible or intangible describing products, systems, services, or processes. The goal of a digital twin usually is to simulate the behavior of the physical opposition to optimize its performance [Eig20]. Hence a digital twin can be split into three parts: The physical product in the real world, its digital replication in the virtual world, and the information that binds them together [GV17].

In order to fully grasp Grieves' concept, we have to look at a few more constructs of digital twins: A digital twin is either a digital twin prototype (DTP) or a digital twin instance (DTI). A DTP replicates the "prototypical physical artifact", which holds the data needed to produce a physical version of the digital counterpart. A DTI, on the other hand, replicates a "specific corresponding physical product that an individual Digital Twin remains linked to throughout the life of that physical product" [GV17].

A DTI has to be convergent, which means that it is "self-evolutionary", implying that the digital twin changes during its lifetime, especially synchronized to its physical copy. Therefore, a digital twin instance needs to be able to keep track of changes to the physical product, either automatically, semi-automatically, or manually [Eig20]. Hence, making the digital twin instance, the type of digital twin that needs to be and is used on the K4R platform.

These DTIs and DTPs operate on a digital twin environment (DTE), which serves as the integrated, application space of a digital twin [GV17]. In terms of K4R, this is the space that integrates external information into the system such as retail product information from the ERP system or any of the other external sources, that can have an impact on the behavior of the digital twin.

2.3.1. Semantic Digital Twin

In the scope of this thesis, we will be often referring to the digital replica of the supermarket as Kumpel et al. defined the *semantic* digital twin (semDT). It serves as a specific layer of a digital twin model. The semantic layer of a digital twin solely represents the concrete representation of the store, which makes it the one capable of analyzing the current state of the digital twin. It creates a semantic connection between two components. The symbolic knowledge base holds the information about products or shelves such as their taxonomy or 3D models and the scene graph representing the relative locations of products to shelves or layers of shelves for example [KMB21].

It is the "symbolic representation of robots, human beings, and their environment" [KMB21]. The semantic digital twin is then used to store, interpret and inquire data from different objects such as the ERP system as a knowledge base, but also other subjects of the system. An overview of a semantic digital twin can be seen in Figure 2.3.

As seen in the graph, ontologies represent the knowledge taken in from other subjects through interfaces but also the internal knowledge bases. An ontology in the retail store example could be some information about a product that can be combined with other ontologies combining more specific information such as the product type or its current position in the store. In other words, it creates meaning out of separate facts and entities about an object [SS10].

The communication between subjects of the IT system and the semDT is handled through different interfaces handling the requests of a subject. Typically in an IT system, these are referred to as application programming interfaces (API).

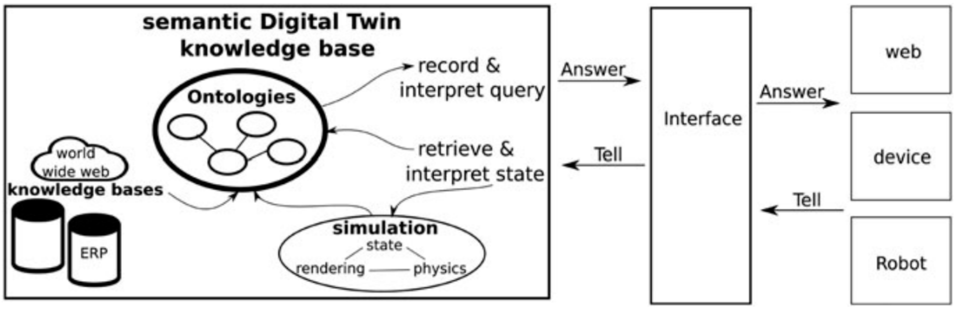

Figure 2.3.: Semantic digital twin and its involved components [KMB21]

2.3.2. KnowRob

KnowRob was first introduced in the year 2009 by Tenorth and Beetz [TB09] and later built upon it in 2018 [Bee+18]. It serves as a knowledge processing framework set to bridge the gap between vaguely described tasks and the accurate information needed for a robot to fulfill its task. An example lies in the correct interpretation of the command to "clean up". The framework can be used to integrate information from various sources, not only for robots but suitable for a semDT as well. Among the various actions it can take, it is able to combine information from robot sensor input, common-sense semantic knowledge, or task descriptions as well as provide classifying and clustering methods or query interfaces and visualization tools [KMB21; HB19; Bee+18].

2.4. K4R Platform Components

In this section, we describe the different parts of the system and what each component is meant to achieve. The information about the use cases and what they are capable of are provided by the projects website [K4R20] as well as its project-internal Confluence pages.

2.4.1. The Retail Store

The physical store has a digital representation known as a digital twin (section 2.3). To fully understand what the different use cases are responsible for, additional focus is set on what a store represents.

A retail store generally consists of multiple aisles, which each consists of a number of shelves, consisting of multiple layers or levels that hold a number of items of a product. This generally sums up the customer viewable area of a store apart from the cash registers. Behind the scenes, the digital twin also represents the warehouse, where deliveries from suppliers are received via pallets of products. We take a more in-depth analysis of a store's different parts in section 5.2.

2.4.2. The Core API

Knowledge4Retail's retail store communicates with other components through Core API, built on REST (Representational State Transfer), "an architectural style for designing networked applications" [Zho+14]. The REST API is stateless, meaning that a message sent includes all information needed for the counterpart of the communication to understand it. The application layer of a RESTful API is built upon the methods of the Hypertext Transfer Protocol (HTTP).

As this thesis presents a concept for access control without an implementation approach, further depth into this is not required. Nevertheless, it is important to note that the Core API provides various forms of requests through HTTP. It enables subjects to read (`GET`), create (`POST`), append (`PUT`), update (`PATCH`) and delete (`DELETE`) objects of the database, amongst other operations [BFF96; DS10]. An exemplar of such an HTTP request can be seen in Figure 2.4, which returns the store instance with the specified `storeId`.

```
GET k4r-core/api/v0/stores/{storeId}
```

Figure 2.4.: A conceptual example of an HTTP request header on the semDT

2.4.3. External Components

Alongside the four use cases, we have a few external sources of information that the digital twin takes into account. The most important external component is the enterprise resource planning (ERP) system of the supermarket, which delivers all the information surrounding the products used inside a store. This includes all kinds of legacy data, like its pricing, size, or weight. Typical ERP applications used on the market include SAP ERP[2] and Microsoft Dynamics[3]. In order to take all the different ERP applications into account, an ERP Adapter was implemented to create a universal communication with the K4R platform. That way the system is not limited to a singular ERP software but can communicate indirectly with any of them. Apart from the ERP integration, there are three smaller external sources that influence the platform.

First of all, we have the **Hetida Designer** which is "a graphical composition tool for analytical workflows based on the Python data science stack" [ins22]. It is solely used in the two use cases, intelligent intralogistics (subsection 2.4.4) and strategic retail marketing (subsection 2.4.5).

Secondly, we need to list **Ubica Robotics**, which records information about a store for the Knowledge4Retail platform, by creating a digital image of the store. This information is used to monitor a store and is further discussed in subsection 2.4.6.

And lastly, we have the **Kaptura** Product Scans, which are used for the digitalization of the products by creating digital 3D models of all retail products, along with their metrics

[2] https://www.sap.com/products/enterprise-management-erp.html

[3] https://dynamics.microsoft.com/de-de/

of weight and size [Gmb22a]. These models are stored in the semantic digital twin for the corresponding retail store products, for robots and employees to recognize in shelves and on deliveries.

2.4.4. Intelligent Intralogistics

The first of the four use cases is about optimizing the logistics inside the store. Intralogistics is the most expensive step in the supply chain. It takes up 48 percent of the total costs of logistics [K4R20]. To optimize the logistical process the use case takes information from the semDT to make filling shelves and managing the warehouse more efficient. This process can be split up into four separate functional units:

1. The **Support for Branch Commissioning** which uses the information from the digital twin and calculates the shortest route for the items with an optimally arranged commission order through an AI-powered Hetida Workflow.

2. This information is passed on to the **Generic Tour Planner** which creates the optimal sequence of aisles to visit using the data from the workflow.

3. When the tour is created the **Presorting** process may begin. It takes the delivered pallets and tells the warehouse worker on which trolley, which box needs to be placed to minimize the stocking time.

4. Finally, while an employee is filling the shelves on a tour, he/she can take advantage of the **Optimized Stocking Strategy**. With the help of a robot, the employee is shown where to put each item through a light beam, pointing at the exact spot in the shelf that the item needs to be placed.

2.4.5. Strategic Retail Marketing

In order to keep up with online retail, retailers have to create more personalized experiences for their customers [Hei17]. To achieve this, retailers can make the assortment of a store more specialized on the region, the season of the year, or simply the size of the store. Furthermore, the visibility of a product directly affects its sales, vertically, as products on eye-level height perform better as well as horizontally since products at corners of aisles have higher sales [RU10]. According to Drèze et al., horizontal and vertical positioning affects the difference in sales of a product by 59%, from worst to best positioning [DHP94; RU10]. To enhance the assortment, the use case focuses on creating planograms, models that visualize a single shelf and how the different products need to be placed inside it. It visualizes how many facings each of the products should have on a shelf. An example can be seen in Figure 2.5.

Just like the **Intelligent Intralogistics**, the process can be split into four separate functional units:

1. **Sales data** is used to calculate a weighted index on the revenue over the past 60 weeks depending on the item and its position in the store.

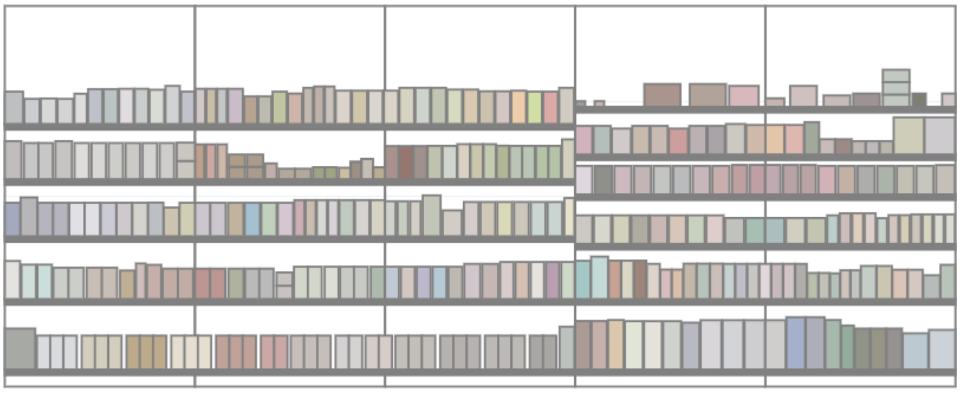

Figure 2.5.: Example of a planogram, showcasing a single shelf. Source: Confluence

2. A **List layout** translates the visual planogram into a computer-readable list.
3. The **Customer Application** customizes the list layout depending on the difference in aisle shelf sizes for individual stores. It changes the number of facings of each product depending on the sales data of the item.
4. Lastly, we have the **Optimization in Marketing**, which uses the real positions of products and their sales data to systematically place products to maximize revenue. This process is done by an AI-powered Hetida workflow and serves as the start and endpoint of this subsystem.

2.4.6. Service Robotics

Apart from artificial intelligence integration, the other technology Knowledge4Retail uses, is the integration of robotics to assist and support the employees in their everyday tasks. Each robot is equipped with a set of sensors that enables it to navigate freely inside a store. The functional units are developed and tested at the DFKI.

The Knowledge4Retail project includes four separate robotic use cases, designated for a specific task in the store:

- The **Autonomous Transport of Goods** is designed to move the sorted pallets to the aisles where an employee can refill the shelves.
- The **Store Monitoring** robot drives around every part of a store to create a digital representation of the store and sends the data to the digital twin to create realograms, a planogram of the current state of each shelf in the store. This robot is currently outsourced and maintained by the previously mentioned company, Ubica Robotics.
- The **Pick & Place** robot is designed to drive through the store and place items in the correct shelf without needing the help of a human employee. It does that with the help of a robotic arm.

- The **Pepper Assistant** is a customer-facing mobile robot, which is meant to drive around the store and help clients find their way around the store, providing additional information about products as well as showing them where specific items are located [KSB+19].

In terms of research for these use cases, there have been numerous pieces of literature published to use the robots in practice. Kazhoyan et al. studied the scalability and robustness of mobile robots in a real-world environment [Kaz+20]. For the concept of a Pick & Place robot, research has tested the machine with real objects, and whether it is able to correctly identify and pick up said objects [MB19; Tha+21; Cos+20].

2.4.7. Intelligent Refrigerator

The last of the four use cases is dealing with the intelligent refrigerator, which can be seen as a smart shelf. It is equipped with sensors to know the specific inventory of that refrigerator. It can keep track of its take-out and restock on its own. Additionally, it can serve as an additional Point-Of-Sale where customers can seemingly take out items and pay for them automatically over the store's ERP System. This use case was created to showcase "the future of Automated Autonomous Retail" [K4R20].

The American role model for this concept is Amazon Go, which builds small stores entirely around sensors to track a customer's movement and product takeout. Resulting in processing a purchase without needing a regular cash registry checkout. With a big tech company like Amazon on the market, they threaten to undermine the regular retailers, creating the eminent need for an open-source approach [PB18].

3. Methodology

This chapter illustrates the methods used for the following chapters of this thesis Starting off with presenting the process of conducting expert interviews in section 3.1, including the initial results of the data flows. Followed by the methodology behind creating the access control concept in section 3.2 and the evaluation process of the created concept in section 3.3.

3.1. Expert Interviews

With the theoretical background in mind as well as the foundational overview of the Knowledge4Retail platform, we conducted semi-structured interviews. We first go into detail about how the interviews were conducted and why. For the purpose of conducting the interviews, we first defined the reasoning behind the interviews. Furthermore, we present the process of how the interviews were held as well as with whom.

3.1.1. Motivation

So far we were able to get a general understanding of the Knowledge4Retail platform through the documentation of the project on its Confluence website and by participating in project internal meetings. In order to gain a broader vision of the platform and where it is ultimately headed, we conduct the interviews to get a better understanding of the system's various functionality. In particular, this process is split up into three main parts.

The first of three is concerning the general architecture of a supermarket chain. To provide access roles for each and every subject of a system, we need to establish what the different jobs and practices are that are handled in a supermarket branch but also a supermarket chain as a whole and more specifically what they are supposed to do and what not.

Secondly, with insights into the humanoid fraction of the system's subjects, we need to take a look at the digital side. Similarly, we need to establish a better understanding of what the different processes, defined in section 2.4, are performing in detail. Resulting in a definition of what each of the use cases (and external components) of the project need access to and whatnot. However, this also involves the understanding of the architecture of the IT system of an instance of the semDT in a store, regarding the data flows between the different components and the digital twin.

Lastly, we want to analyze the bigger picture of the project with the general architecture in mind, which might also influence the way access control is planned out. For that, we need to understand how the system is eventually distributed. This can be done either in a local scenario where each store has its own individual Knowledge4Retail platform integrated or by installing the service centralized in an entire supermarket chain or even outside the company with each of the companies, having their own individual store instance. This ultimately comes down to the question of whether an on-premise or cloud version is more sufficient. There are a few differences between those options that need to be discussed to find the optimal solution.

Additionally, we want to discuss the possibility for external companies to make use of the platform. By providing suppliers with the chance to gain insights into the performance of their products in a store from a supermarket chain. This could include the products' placement in the store or their current pricing. This would not only showcase a potentially interesting analytics tool for the companies but also establish how total outsiders of the system might be able to interact with it.

3.1.2. Interview partners

For the selection of the interviewees, we wanted to choose experts from all the different parts of the Knowledge4Retail project. Hence we chose partners from almost every major branch, notably without the use case of the intelligent refrigerator, which is compensated by information provided by partners with a broader understanding of the project. The distribution of the six interview partners in their area of expertise, as well as their company, can be seen in Table 3.1. To preserve the interviewee's anonymity, we are referencing them with alphabetical letters.

Interviewee	Area of Expertise	Company
A	Project Management	dmTech
B	Intelligent Intralogistics	dm
C	Strategic Retail Marketing	dm
D	Service Robotics	Uni Bremen
E	Service Robotics	DFKI
F	ERP Integration	Nagarro

Table 3.1.: Interview partners and their area of expertise

With the project's primary retail partner being dm, this is where we retrieve the most intel, resulting in the major dm-related selection for the interviews. Additionally, the latter three partners provide more technical insight. While interviewees D and E are both active in the service robotics use case, D is working more on a practical note at the University of Bremen, whereas E is working at the DFKI, which focuses on the AI applications of the robots. At last, interviewee F is working with a multitude of supermarket chains on the integration of ERP systems. This lets him provide a broad overview of how different supermarkets function.

3.1.3. Guiding presentation

The partners come from various companies and professions with different levels of expertise in access control and computer science as a whole. Hence, to properly guide the interviewee through the displayed topics, we create a visual presentation to help the interview partner grasp the presented concepts.

Roles of a supermarket store

For the first topic, the different roles of a supermarket, we first outline the concept of RBAC (subsection 2.2.3) and showcase a basic hierarchal example of roles in a supermarket facility. It displays an admin standing at the top and below a branch manager and on the lowest level: cashier, shelf filler, and warehouse worker. On the basis of this visual presentation, we ask the interviewee about the presented roles and what they do, and whether the chosen hierarchy is sufficient or open for improvement. For the lowest level of our proposed hierarchy, we are especially interested in the separation of the three jobs of a supermarket, building a potential foundation of a separation of duty (subsection 2.1.3) for these roles. Furthermore, we want to explore the role of the cashier, due to the fact that it might not need access to any of the methods of the K4R platform, because of its reliance on the cash registry and nothing else.

Roles of the platform

After the first section of the interview is completed, we move on to the digital components of the project. For the architectural overview of the platform, we display a simplified UML component diagram of the entire system, created through our present understanding of Knowledge4Retail. It is important to mention that the placed interface connections are not meant to be entirely correct, but simply used to illustrate where we have a flow of data between different components.

The result can be seen in Figure 3.1. Each component of the diagram was colored differently to make it easier for the interviewee to differentiate the components during the conduction of an interview. As previously described in section 2.4, we placed the digital twin in the center with all of the use cases and external components surrounding it. Inside the component, we provided the database of the retail store in the form of nine entities. The entities shown represent the initial pilot entities of the project and therefore give a simplistic understanding of the semantic digital twin for the interviewee. The definition of each is described in the following, using our current knowledge ahead of the interviews:

- `Product` (Pr): Holds the master data of an item, like its metrics or normal pricing
- `Article` (A): An individual instance of a product
- `Layout Piece` (LP): The physical object of a shelf or any fixed piece of store layout
- `Shelf` (Sh): The inside of a shelf holding all of its items and individual facings
- `Store` (St): Information about the store as a whole
- `Picking Order` (PO): Some form of commissioning order (for example: Click&Collect)

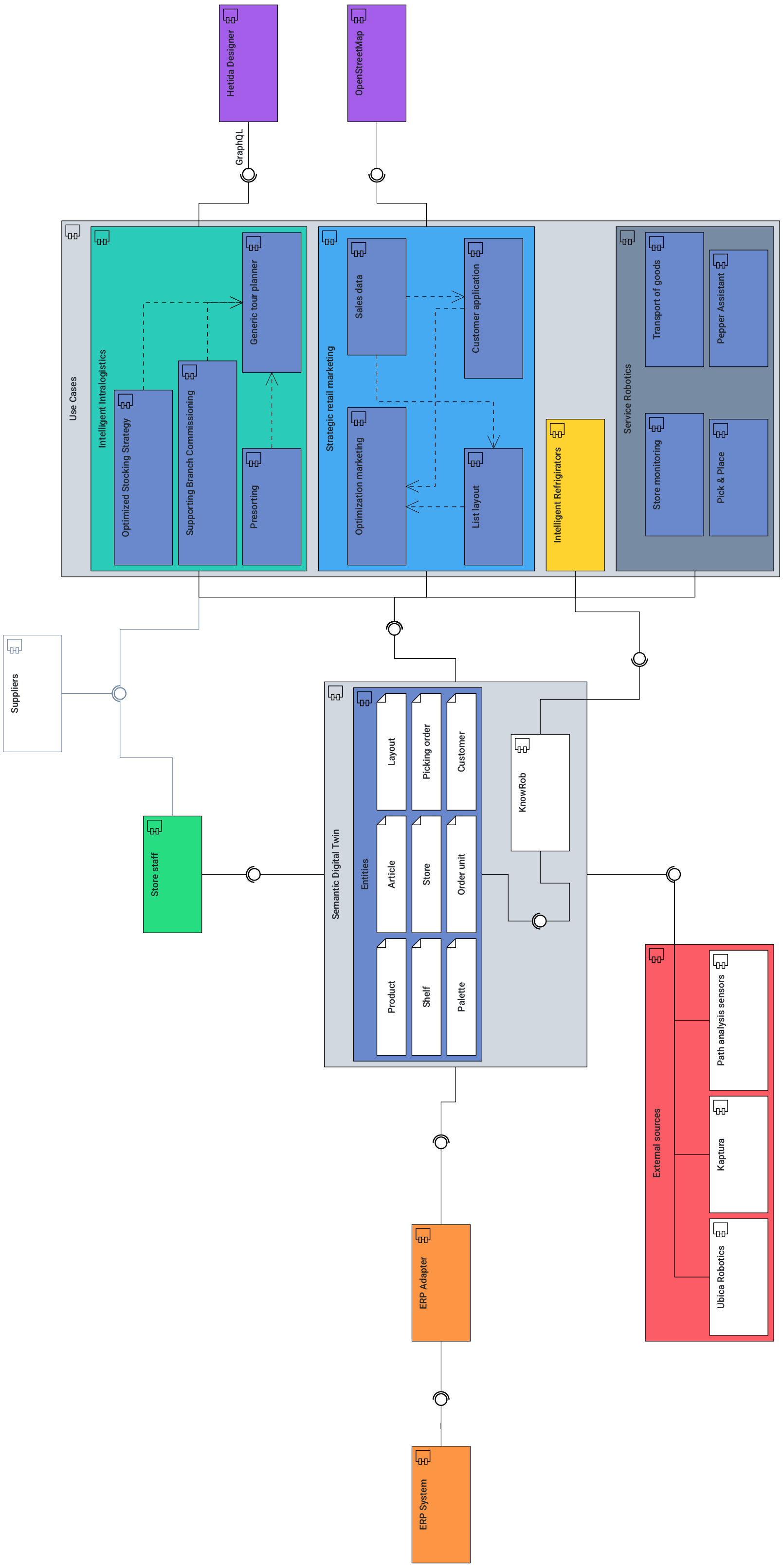

Figure 3.1.: Component Diagram before the interviews

- `Palette` (Pa): A palette in the warehouse that is also carried through the store by the robots for autonomous transport of goods
- `Order Unit` (OU): A delivery order to the store
- `Customer` C: Information about a customer

Additionally, we added the component of the knowledge processing framework, KnowRob (subsection 2.3.2) inside the digital twin component, due to their close internal relationship.

Around the semDT, we positioned four areas. On the left and at the bottom we have the external software providers of the system with the ERP system and its translation adapter on the left. The other smaller external components are encapsulated below the semDT. At the top, we find the store staff interacting with the digital twin. Above it, in grey, we added the suppliers as a potential addition to the structural overview.

However, the biggest part of this diagram is on the left, where we attempt to visualize the relationship between the four different use cases and the digital twin. For this, we split the different use case components into smaller subcomponents and made a hunch about their relationship with one another, which is reviewed by the different interviewees. In the intelligent intralogistics (subsection 2.4.4), we decided on unidirectional dependencies towards the tour planner. The processes of the **optimized stocking strategy**, **support branch commissioning** and **presorting** all present data to the **generic tour planner**, which uses the information to create the optimal route through the store. For the strategic retail marketing (subsection 2.4.5), the dependencies are more entangled. With the help of the **sales data**, the **customer application** and **list layout** are created which are then both used in the AI workflows of the **optimization marketing** to improve the positioning of the individual facings. To the right of the two use cases, we placed the Hetida Designer and the OpenStreetMap database. According to the documentation, the Hetida Workflows were directly influencing the intelligent intralogistics and the map database was used to help optimize the planograms geographically. The intelligent refrigerator is not split into further subcomponents, according to its documentation. Nonetheless, the service robotics use case holds four individual robotic use cases, which each individually communicates with the semDT, eliminating the need for dependencies between each of them.

After asking the interviewee about the accuracy of the presented diagram, we take a more thorough look at the four use cases individually, asking the interviewee depending on their expertise about each of them. For this, we prepared separate tables for the four different use cases. Each table showcases an access control list (section 2.2), where each of the subjects is one of the subcomponents of a use case and the objects are the nine chosen entities of the retail store. The tables can be seen in Tables Table 3.2-3.5, where the entities are abbreviated. For an effortless understanding in an interview, we chose only *read-only* and *read-write* as access rights for the respective subjects, coloring each cell, depending on whether the subject should have no access, read-only or read-write. In the table below, we abbreviated the two respective access rights with "r" and "rw". Upon the provided documentation of each of the use cases, we filled in the tables according to our current understanding, to create comprehensible interviews without the need to go through every cell of the table, but focus on those that are unclear or false.

Intelligent intralogistics	Pr	A	LP	Sh	St	PO	Pa	OU	C
Optimized stocking strategy	r	rw	r	r	r	rw			
Generic tour planner	r	r	r	r	r				
Presorting	r	rw	r	r	r		rw	rw	
Support branch commissioning	r	r	r	r	r	rw			

Table 3.2.: Initial access control list for the intelligent intralogistics

Strategic retail marketing	Pr	A	LP	Sh	St	PO	Pa	OU	C
Sales data		r	r		rw				
List layout	r		r	r	r				
Customer application			rw	rw	rw				
Optimization marketing	r	rw	rw	rw	rw	rw	rw		rw

Table 3.3.: Initial access control list for strategic retail marketing

	Pr	A	LP	Sh	St	PO	Pa	OU	C
Intelligent refrigerator	r	rw			r	rw		rw	

Table 3.4.: Initial access control list for the intelligent refrigerator

On-premise & Manufacturer integration

For the two minor parts of the interviews, we provide more guiding visual support. For the questioning of on-premise or cloud hosting, we simply feature the previously described three options on a slide (on-premise, company-centered on-premise, cloud hosting) to help the interviewee, find the fitting solution according to their situation or company. And lastly, for our last section of the interview, the proposition of an analytics tool for product manufacturers and suppliers, we provide some bullet points with a basic explanation of the idea (previously described in subsection 3.1.1).

3.1.4. Questions

Consequential, we present the list of questions, asked alongside the guiding presentation and the various explanations that came with it. All interviews are held in German, hence the

Service robotics	Pr	A	LP	Sh	St	PO	Pa	OU	C
Autonomous transport of goods		rw	r		r		rw		
Store monitoring	r		r	r	r				
Pick&Place	r	rw	r	r	r				
Pepper assistant	r		r		r				rw

Table 3.5.: Initial access control list for service robotics

presented interview questions are translated.

1. Roles of a supermarket
   a) What are the typical work areas in a supermarket branch in your opinion and how would you separate them from each other?
   b) What do you think they should be allowed to do on the platform?
      Example: Warehouse workers interact with the system only through wearables, but the store manager can access data about the system through a separate application.
   c) What is the highest level of authority in a supermarket?
      i. What is the branch manager allowed to do?
      ii. What is determined from higher up?
   d) How do cashiers interact with the system?
2. Roles of the platform
   a) Do you think the structure as it has been set up is reasonable?
      Note: Regarding the UML component diagram
   b) Are there any subcomponents of the use cases that you are missing?
   c) What access rights do the following use cases need?
      Note: Presenting different ACLs (Table 3.2-3.5)
3. On-premise or cloud hosting
   a) Does each store get its own instance or is the program centralized in a company like for example dm?
   b) Do you think a universal cloud application is more suitable?
      Explanation: Cloud version would result in less effort on the customer side and enable further training of the AI tools.
4. Supplier integration

a) Do you think this is a reasonable addition to the system?

Note: The idea to give outsiders like manufacturers or suppliers of the products, insight into the system.

5. Closing question

a) Do you have any additional questions?

3.1.5. Analysis

The interview questions were answered by the six interviewees. All the interviews were transcribed with the help of software for automatic transcription of audio files, provided by `sonix.ai`. The software's output was then corrected and validated. It is important to mention that the answers cannot be evaluated empirically. Rather every comment and response needs to be evaluated and put into perspective, regarding the other provided answers. Therefore, the finalized transcribed documents were encoded using the software, MAXQDA2022[1]. Each code represents a text segment, which serves as a categorization of the interviewee's answer. For the thesis, we have divided the different categories into the four different question blocks of the interview.

- Roles in a supermarket
- Roles of the platform
- On-Premise
- Manufacturer integration

For the roles of the platform, in particular, we further divided the block into its various parts. On the one side the component diagram itself and the ERP integration and on the other side the four different use cases. The corresponding figure can be found in Figure 3.2.

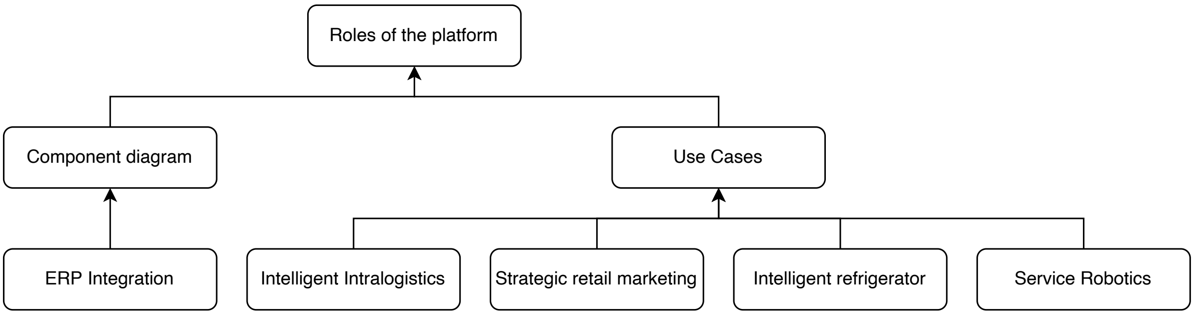

Figure 3.2.: Interview categories of the roles on the platform

Under each of the categories are various codes, determined during the process of encodement (with MAXQDA2022). These codes create the basis for the presentation of the interview results in chapter 4.

[1] www.maxqda.de

3.2. Data Access Control Concept

After having conducted the expert interviews, we use the received new insights to create the access control concept for the Knowledge4Retail platform.

To begin, it needs to be established which access control model has been chosen. As previously hinted at, we will be taking advantage of the RBAC model. section 2.2 presented various access control models. The **Bell-LaPula Model** is focused on restricting the access depending on the subject's level of security clearance, expressing a strictly linear hierarchy of access rights. Similarly, the **DAC** lacks the needed complexity for an entire platform, as its primary application lies in computer operating systems. However, both models build the basis for the **role-based access control** through the early adoption of hierarchical classification and the establishment of various access rights.

Role-based access control is the perfect fit for the K4R platform, due to its business-focused approach. The classification of subjects into their activities instead of solely their identity is only one of the key factors to use RBAC in a business environment (more found in subsection 2.2.3). **Attribute-based access control** on the other hand is too complex. In terms of the Knowledge4Retail platform, due to its static appearance, we can have predefined roles for each and every component. Hence, there is not enough variance in the permission rights to experiment with the ABAC model. A version of the RBAC model is sufficient.

The concept creation can be split into three layers. First, the new information is used to enhance and correct the component diagram. Upon that the digital twin which was previously represented through the nine entities used in the early development of K4R is finalized to create a concrete system to access. And finally, roles are defined, both for the different data flows, described in the component diagram as well as for the individual humanoid subjects of the system. For each of these roles, a set of access rights needs to be established. The results of the concept are presented in chapter 5.

3.2.1. Component diagram

The methodology behind the finalized architecture of the entire system is fairly straightforward. Previously, the documentation of the project was used to create an overview of the system through the creation of a UML component diagram. Using the better understanding of all the different use cases and components from the conducted interviews, the flow of information is altered to create a more realistic conceptualization of the architecture. Additionally, the construction and definition of various components of the system are changed to better serve the intention of the concept, creating access roles for every single component. Therefore, in the process of creating the final version for the platform's architecture, components are clearly defined in what information it needs and with whom it communicates to perform its objective properly.

The different data flows that are defined on this layer of the access control model serve as the base for the component's access control list. In the UML component diagram, the data flows are marked through interfaces for each communication endpoint. The one end provides something, provided through the interface, and the other end requires it to fulfill its task. These interfaces are labeled regarding their purpose. To serve the purpose of the interviews, the number of interfaces and relationships between different components was previously minimized to improve its presentability. In its final instance, the correctness of the diagram stands in the foreground, which is creating a more complex illustration than the previous iteration. In the general sense of K4R, the digital twin provides information while other components of the system require it to function, which brings us to the second layer of the concept.

3.2.2. Semantic Digital Twin UML

Previously, during the conduction of the interviews, nine entities were used to resemble the digital twin (or retail store). The nine entities were used to create a comprehensible overview of what kind of entities need to be accessed. They made the discussion of access rights on the system more approachable. For the final concept, the entities lack the necessary depth, which is why a UML diagram is introduced. Hence, we use the results from the interviews to create a better understanding of the digital twin.

The retail store as a digital twin is displayed using a conceptual UML class diagram. From the Confluence documentation of the project, an implementation-based UML diagram is used. The documentation diagram's empathy lies in creating a perfectly accurate representation of the implementation, it has an overwhelming complexity and needs to be simplified for the sake of the concept. Since each class resembles an object in the access control concept, it is important that the result stays tangible. Therefore, the goal of this diagram is to find the middle ground between the plain nine entities and the overwhelming implementation UML diagram.

Due to the fact that our diagram does not need to resemble any implementation functionality, we solely analyze the different classes *necessary* for the access control concept and how they are integrated into one another. In addition, we provide conceptually based attributes of each class to achieve an appropriate understanding of what kind of information each class holds.

3.2.3. Role Definition

With the subjects and objects of the system defined through the prior two layers, we can define the different roles. This process is split up into two parts. Similarly to the course of the interviews, we will look at the digital and human roles separately.

For the digital side, we already boiled down the different components to best serve as roles through the **component diagram**. In terms of the human side, the results from the interviews from subsubsection 3.1.3 are used to define all the necessary roles that need to interact with the K4R platform in one way or another. After reasonable roles for both sections are defined, a role hierarchy for each is created (concept explained around Figure 2.2), based on the collected insights from the interviews. With all roles defined in their respective hierarchy, we connect the dots and create the access control matrix in the form of ACLs.

3.2.4. Access control list

As described in subsection 2.2.3, after defining the `User` and `Roles`, the set of `Permissions` has to be defined. For that, we introduced the concept of ACLs where each row represents a subject's access rights to all the objects of the system. Each cell can be filled with one of the predefined access rights (*read-only*, *append*, *execute*, *read-write*, and *control*). Equivalently, we substitute the subjects with our predefined roles. Just like the roles are separated into two sections, we create two separate tables for the human access rights and the rights of the components.

Ahead of the interviews, the documentation already provided a decent overview of the information that the components need. Therefore, the results from the presented ACLs from the interviews are used to better interpret the comments of the interviewees, regarding access rights. Furthermore, the newly formed set of roles and objects needs to be integrated (replacing the nine initial entities), forming the final access control list. The updated ACLs using the initial nine entities were also updated, but since they only duplicate the explanations for the final ACL, they can be found in section A.1.

In the case of the staff roles, there was no prior knowledge regarding the needed activities on the K4R platform. Accordingly, the interpretation of the interview results is the exclusive source for the definition of the access rights of the roles. This marks the finalization of the access control concept.

3.2.5. Additional findings

In terms of the interviews, two additional topics were introduced. The integration of manufacturers into the system and the hosting style of the platform. The first of which serves as an additional proposal for the access control concept. The latter takes a look at the broader picture of the system, and how individual instances of the system should interact with one another. Similarly, through the results from the interviews, a proposition is made regarding how a given hosting influences access control and which one is the most intriguing for different types of retail companies.

3.3. Evaluation Interviews

In order to validate the concept for the Knowledge4Retail project, two additional interviews took place. Just like the access control is split into two parts, the interviewees come from their respective areas of expertise in the project. Interviewee A, who was already part of the expert interviews, and who is in charge of the project management from dm is necessary to validate the side of the staff roles. On the other hand, interviewee G has not been part of this research yet, he is involved in the implementation of K4R. His knowledge of the implementation of the project is needed to properly validate the different architectural concepts created, as well as the access rights on the component side. Accordingly, with each interviewee, the focus was set on their respective areas.

Each interview lasted between 20-30 minutes. During the course of the interview, each interviewee was presented the different visual illustrations of the concept, discussed in section 3.2. Each topic was followed by an open discussion regarding its plausibility and correctness.

The results from the evaluation will be embedded in the thesis' discussion (chapter 6). Since the concept was already finished beforehand, we did not alter the elements of the created concept in regards to the suggestions from the evaluation interviews. The results are solely presented with brief elaboration on a possible integration in the future.

4. Interview Results

In this section, we give an overview of the most important statements taken from the interviews by examining the various codes that were set. Each interview was set to last 30 minutes. In the end, the interview time lasted between approximately 24 and 50 minutes. For the presentation of the results from all the interviews, we take a look at the nine categories individually, though it needs to be said that the number of statements and codes per category strongly varies, due to the fact that with each interviewee, the discussed topics depended on their level of insight in each of the categories. This led to the most discussed category being the *roles in the supermarket* with all of the interviewees being able to give their take on the issue but resulting in contradictory comments on some topics. On the other hand, the *ERP integration* for example could only be discussed with experts in the field, and hence we have fewer responses on certain topics therefore of higher quality. In general, the codes that have the most segments are the ones that were specifically asked upon from the interview questionnaire, other codes appeared from the general flow of dialogue. In the following, we go over every category and its respective codes.

4.1. Roles in a Supermarket

For the first category, all roles that were discussed were assigned to a code. The following presents the results from every code. The list of codes includes: the role of a **store employee**, **branch manager** and all related roles from a retailer's **headquarters**.

4.1.1. Store Employee

The first role we need to take a look at is the regular store employee. The main topic of this role was regarding its set of jobs. Five out of the six interviewees have pointed out that there is no possibility for the separation of the three activities (cash registry, shelf filling, and warehouse) in a normally sized supermarket branch, like in a dm or Rewe. The reasoning is always similar, through the small nature of the stores, a worker does what is currently needed the most, which results in rapid changes in their activities throughout the day.

> The employees who work for us in the stores. We don't have such huge stores. They always have all the roles, [...]. If a cash register is needed, then they open a third cash register also. And when cash register 3 is closed again, they refill the shelf. Their role changes fluidly. (Interviewee A, dmTech; translated)

The frequently used example has been regarding the interchangeability between filling shelves and doing cash registry. A change into warehouse work is not so fluid and might be more time-consuming, opening the possibility for separation of duty. However, it is also noteworthy that in the case of dm, a warehouse does not necessarily exist in a supermarket branch.

> In other words, our checkout staff are also shelf replenishers, and at best we don't have a warehouse. [...] [The staff member] just stays somewhere at the front in the checkout area, which means theoretically he doesn't do anything in the warehouse. (Interviewee C, dm; translated)

Furthermore, with the potential exclusion of the warehouse role in some cases, we also need to take a look at the cashier. As interviewee D argues an employee has no impact on the K4R ecosystem while sitting at the cash register.

> [...] this runs via the SAP system, because we get the data from the SAP system and accordingly, I don't know whether the cash register really needs its own user, at least in the Knowledge4Retail platform.
>
> (Interviewee D, Uni Bremen; translated)

Apart from the question about the cashier's lack of influence, it was pointed out by interviewee B that a normal employee should be able to see the planograms of the store. An employee needs to compare the target shelf facings with the actual current placements. Interviewee C took this proposition one step further.

> [The employees] can only operate a simple user interface [...]. On the one hand, it is a changing occupancy and on the other hand, there is also simply a lack of know-how. (Interviewee C, dm; translated)

Additionally, interviewee D has provided a suggestion for how an employee should interact with the system apart from reading information. He explains that it is not the user who is interacting with the system, but some sort of wearable that can have writing access on the system.

It was also mentioned multiple times how a branch of dm has a very flat hierarchy. A branch manager sitting at the top and the regular employees are below. However, interviewee F argues that there are certainly larger hierarchies in place at other supermarkets. Interviewees A, E, and F mention the use case of contract workers.

> In really large supermarkets, this issue of shelf replenishment is often outsourced as a contract for work. Another company comes in and fills the shelves.
>
> (Interviewee A, dmTech; translated)

This results in a "company border" as interviewee A puts it and argues that these workers should have the freedom of doing what they are told without personnel from the supermarket itself interfering. Interviewee E also mentions the possibility of mini-jobbers doing the inventory in a store, going in and out just to count and scan items. Similarly, interviewee F talks about the outsourcing of the warehouse in "Do-It-Yourself stores", such as Hagebaumarkt or Obi, where the warehouses are completely separated from the rest of the store. It is called a market distribution center with its own market distribution workers.

4.1.2. Branch manager

The branch manager is the head of a branch. The different interviewees had various takes on what differentiates his/her position from the ordinary employee. A manager should have the option to `control` access rights to assign jobs to new employees but according to the interview partner from nagarro, it ends there.

> A branch manager [...] has almost no more rights than normal branch employees. [...] Temporary workers, such as shelf replenishers, [...] he can give them authorization to use the system. But there is not much more.
>
> (Interviewee F, nagarro; translated)

dm's own interviewee C also mentioned the need for the `control` of access rights but argues that the manager should have further reading options on the system. She points out that a manager should have overseeing reading access on "dashboards on some key performance indicators (KPI)". Interviewee A adds that specifically for dm, the managers always have the last saying and basically full control over what goes in and out of the store if they wanted to.

> Every day, [dm headquarters] calculates order proposals for the stores, stating what additional goods, [branches] would need according to the data. [...] [Managers] can say: I know tomorrow is [. . .] summer party from the kindergarten [...] and therefore I set here and there the stock higher. In fact, they have full authority to intervene, to order, and to edit. They also see sales, they see everything that happens to the balance sheet in their store. Only suggestions ever come from headquarters, and the branch manager is the master of his land.
>
> (Interviewee A, dmTech; translated)

This goes along with the previously mentioned flat hierarchy that dm has, but again for other supermarket chains more hierarchical layers are typically in place. In general, the interviewees individually agreed that a branch manager should only be able to have access to their own store. In a more hierarchal company, this would translate into regional and maybe even national managers also being in place. Managers who can oversee a number of stores depending on their level, according to interviewee D. Instead of the dm specific KPIs regarding a single store, these could analyze data from their specified set of stores as well as provide specific pricing or restocking depending. Just like the branch manager could but on a higher level, affecting more stores.

4.1.3. Headquarters

At the top of every supermarket chain hierarchy lies a headquarter, which pulls the strings. In this section, we go over all the codes linked to the center of operations. All information regarding a headquarter of a supermarket chain is retrieved from the dm internal interviewees, A, B, and C. However, all presented roles and activities presented by the three interviewees should be applicable to almost any supermarket infrastructure, that plans on using the K4R platform. The presented roles, mostly revolve around analytical workflows. Interviewee C presents three roles: assortment manager, category manager, and data scientist. The assortment manager evaluates the item placements in a branch according to its needs. The manager needs to be an expert in a product segment of a supermarket, knowing what product sales how well.

> There are guidelines that can be optimized and better adapted to the needs of the branch. [...] This means that if I have a nutrition department, I have an assortment manager who is particularly well versed in nutrition articles and can therefore judge particularly well what these placements mean and how well each article sells. (Interviewee C, dm; translated)

The category manager uses the results from the assortment manager's analysis to create the planograms for the stores. The data scientist "prepares and understands data" to create KPIs. According to B, they need to be able to read a lot of data to do so. Hierarchically speaking, because of this, they should be placed above the two types of managers, as B points out. Hence, overall these three jobs provide the branches with suggestions regarding their product placements, pricing as well as overall analytics, which can then be analyzed by the corresponding branch (or regional) manager.

Another topic that was suggested by interviewees tackles the activity of installing a new branch. Interviewee A suggests the integration of the construction of a supermarket where the store is planned out, in terms of where a shelf should be placed. In addition, interviewee E from DFKI focuses on the software installation of the store, more specifically the digital twin and its surrounding components. He further illustrates that it would serve similarly to a system administrator of a facility.

> An installation engineer [. . .] who installs the applications in the supermarket and the maintainer of the application. [...] perhaps a bit more than a system admin, who not only does IT, but installs [...] any AI application or robotics application [...]. (Interviewee E, DFKI; translated)

Speaking of administrative operations, interviewee D explicitly talked about the specific role of a system admin. D mentions how the administration should also be separated hierarchically. Apart from the ultimate administrator at the top, he mentions lower-level admins that focus on data management and especially focus on supporting the branches with their problems and questions. In the context of the lower-level administrators, we can find parallels between interviewee E's installation and maintenance manager, but the exact definition and distinction of roles will not be further discussed in this section (but in section 5.3).

4.2. Component Diagram

After presenting all the different roles described by the interviewee, we move on to the second question block. The first category of the section deals with our created component diagram (Figure 3.1).

4.2.1. Digital Twin

First, we need to take a look at its center, the semantic digital twin, and the nine presented entities. The center point of discussion revolved around the definition of a `Pallet`. Interviewee F (expert of ERP integration) defined a `Pallet` as part of a delivery.

> Deliveries mean a central warehouse delivers to a store, 20 pallets, and on the pallets are the articles. And this is done on a certain date, at a certain time.
>
> (Interviewee F, Nagarro; translated)

However, as later mentioned by F, the Delivery does not need its own entity, since it is represented by the pallets together with the picking order. Initially, we defined the pallet as a storing unit that is also moved across a store to refill shelves. For this specific purpose, B argues that there is a difference between a trolley, which can be moved around the store to refill shelves, and the `Pallet` which is sitting inside the warehouse. The separation into two objects becomes more apparent in the intelligent intralogistic (B's area of expertise). This would be in line with F's definition of a `Pallet`, strictly corresponding to the storage place of delivered items. Furthermore, interviewee F also explained a basic concept for the digital twin's database and how there is a need to differentiate between the two types of data, which is especially vital for ERP integration.

> There is master data, i.e. data that is maintained, and then it is stable for a longer period of time. Then there's moving data, transaction data it's called like delivery documents.
>
> (Interviewee F, nagarro; translated)

Additionally to the `Pallet`'s definition the `Order Unit` was a topic in need of discussion. Initially, we interpreted a `Picking Order` to be an entire deliver unit, however, interviewee A brought to our attention that it resembles a single box, which is called a kollo. A kollo is loaded with items and delivered to a branch along with many other kollos. Interviewee A explains that one should never walk up to a shelf with a single item to restock, but with an entire kollo, that hopefully fits entirely in the product's available facings. Apart from that, the abstraction of the semantic digital twin through the nine entities was not challenged any further.

Moving over to the use cases, interviewee C, from the strategic retail marketing, commented that the `OpenStreetMap` integration is obsolete and not used by her use case. Similarly, she argued that the `Hetida Designer` should be connected to both the `Intelligent Intralogistic`, but also the `Strategic Retail Marketing` use case because the Hetida workflows are heavily used by the program to create the optimal planograms for the store. Interviewee D added that each of the use cases should be strictly consuming data without returning any information to the digital twin, but also agrees that it would make a lot more sense if the use cases would also write back information to the system. He also mentions that he does not believe in a need for a `Store Staff` component since they would only interact with the system through wearables to execute use cases. However, as we know from the previously discussed role of the store employee that is not the case since they also need to read current planograms and more through a "simple user interface". In the same sense, it was pointed out by A that the `Store Staff` component should be renamed to just `Staff`, as it not only represents an access point for store employees but also access from higher up and especially the headquarter. Lastly, it was pointed out that it is reasonable to have all the external components at the bottom, but also the `ERP Adapter` to mostly provide information for the semantic digital twin.

4.2.2. ERP integration

In this section, we take a closer look at the comments specifically about the integration of ERP to the system, where we have our expert, interviewee F. Consequently, we will focus mostly on his statements for understanding the integration. The basic concept, he introduces, concentrates on how the stock in the ERP system must always replicate the actual stock of a retail store.

> Roughly speaking, the ERP system is the inventory management system. This means that the transactions must always be carried out there. [...] The ERP system represents the stock in a physical store [...]. This should also correspond to the actual stock.
> (Interviewee F, Nagarro; translated)

Through this principle, F argues that the communication between ERP and the semDT goes both ways. The ERP system provides the master data for all products as well as providing product deliveries and the semDT sends transactions to the ERP software. A traditional transaction only really exists with regard to the `Intelligent Refrigerator` in the system. Though these are also the only point-of-sales, apart from the ERP integrated cash registries in the system. Therefore, all that is needed to assure the replication of the actual inventory and the ERP inventory.

Additionally, we also took a look at the possible access rights, specifically for the ERP system on the semantic digital twin. According to F, the ERP system should have mostly writing access (*read-write*). Evidently, it should have writing access on the `Product` entities to provide master data, which includes its unit of measure. Furthermore, interviewee F would add writing access on `Layout Pieces`, `Shelfs` and the `Store` itself. Since we have not created an ACL beforehand for ERP, the corresponding access control list can be seen in Table 4.1 for which we specifically use the ERP Adapter, since it resembles the component reaching out to the system.

	Pr	A	LP	Sh	St	PO	Pa	OU	C
ERP Adapter	rw		rw	rw	rw				

Table 4.1.: Access control list for the ERP Adapter

4.3. Use Cases

After discussing the component diagram as a whole and the smaller components, we move on to the more complex components, the use cases. For each use case, we present the most important results. To recite, the four use cases that were touched upon are: **intelligent intralogistics**, **strategic retail marketing**, **service robotics** and the **intelligent refrigerator**. However, we need to clarify that due to the various areas of expertise, we mostly focus on the answers from the respective experts as other interviewees partly contradicted the statements from the use case's expert.

4.3.1. Intelligent Intralogistics

In order to fully understand what the use case is about, we first need to take a critical look at the information flow created in the component diagram (Figure 3.1). According to our initial diagram, all information flows into the **generic tour planner** to create a route. However, as interviewee A (project manager) mentions, it should be the other way around, at least for the act of **presorting**.

> The other way around, from the digital twin, tours are calculated, which then are used in presorting as input parameters. [...] I do the presorting on the basis of the route. [...] [The generic tour planner] serves as a basis and not a result.
>
> (Interviewee A, dmTech; translated)

Furthermore, initially, we understood the use case, `optimized stocking strategy` as a moving trolley (or `Pallet`) moving through the aisles of a store. Interviewee B, our expert on **intelligent intralogistics**, explained that its use case is more stationary. As it is solely used to help the employee fill the shelves through the help of a light beam. Hence, the `optimized stocking strategy` has no need for a tour, whatsoever.

> This is one of those spotlights that virtually locates itself in the store and, after scanning a barcode, displays the item's position on the shelf.
>
> (Interviewee B, dm; translated)

After discussing the dependencies between the subcomponents, we took a look at the use case's access control list, where we looked at each of the subcomponents and how they should interact with the semDT in detail.

Optimized Stocking Strategy

The optimized stocking strategy needs to be considered independently from any of the other subcomponents as touched upon earlier. Due to the contradicting definition of the subcomponent, its access right in the presented Table 3.2 needs some changing. According to B, the component is solely a visualization tool and hence does not need any writing access to the system.

> Even if we look at it independently of its technology and move away from the spotlight solution towards smart glasses or something like that, it's purely a visualization tool. [...] As it is implemented at the moment, no feedback is given on whether something was positioned somewhere, but only what **should** be positioned **where**.
>
> (Interviewee B, dm; translated)

Hence, B suggests that we have no form of feedback from the system or any notion of confirmation that a product is placed correctly in a shelf. Accordingly, the row regarding the optimized stocking strategy of the access control list presented to the interviewees solely needs changes in its type of access right.

Generic Tour Planner

For the generic tour planner, we have discussed most of the important suggestions by the interviewees before with the changes in its dependencies. The only other comment that was made by interviewee B is the fact that the planner should also have reading access on the `Picking Order`. He argues that it is needed to align a route not only with refilling but also with commission orders around the store area. Apart from that, both the project manager, A, and the expert, B agreed with its access control row in Table 3.2.

Presorting

The center of discussion regarding the act of presorting was the exact definition of the `Pallet`, which we already touched on (section 4.2). Our expert on the subject, interviewee B, gave us some further insights into how the process of presorting is carried out.

> Not all pallets are sorted. Rather, it is calculated [...], whether it is useful for a pallet to be sorted, and then, in the next step, a ranking is created [...], for which pallet is the most important. [...] An average store gets between 20 and 30 pallets a week. If you were to presort them onto trolleys, you'd be in the region of a hundred trolleys that you'd stock. [...] It's just completely unrealistic to sort everything, besides that it would be a huge time commitment.
>
> (Interviewee B; dm; translated)

Therefore in the process of presorting, it is first decided which pallets are optimized and presorted onto trolleys. Taking a look at its access control list row, only the access right on the pallet was questioned. Interviewee B elaborates that the trolley's load is *written* onto the trolley object in the **presorting** and then read by the previously explored **optimized stocking strategy**, but argues that the delivered pallet is *read-only*. He argues that another entity for the trolley should be added to properly differentiate between the two. However, it also needs to be mentioned that, due to the limited amount of pallets that can be sorted, regular pallets are also used for transporting goods inside the store.

Support Branch Commissioning

The **support branch commissioning** works very closely with the **generic tour planner**. Therefore, our expert argues that the most important aspect of this subcomponent is to match the needed picking orders with the calculated tour. Hence, it needs *read-write* access for the `Picking Order` to do so. For the other entities, there was a common agreement regarding their access rights.

4.3.2. Strategic Retail Marketing

The **strategic retail marketing** is made up of four subcomponents, just like the **intelligent intralogistics**. Therefore, we also need to take a look at our presented dependencies between the four components, just like before. In this case, we had fewer friction points. Interviewee A only mentioned that after the workflow from the `optimization marketing` subcomponent is finished, a new iteration would begin by sending the results to the `list layout`. Hence, the current illustration only showcases a single iteration and not the continuous circuit between the subcomponents. Apart from the existing four subcomponents, interviewee C, our expert for this use case, explains that there exists a fifth subcomponent.

> What you have now is the optimization of planograms. And that's all true. [...] But we also have [...] the Realogram Analytics. [...] [The planogram] is the current target and [the realogram] shows how it's been realized.
>
> (Interviewee C, dm; translated)

Interviewee C further explains that the entire use cases do not only involve the planograms that showcase how the shelves *should* be filled, but also a secondary tool on how they are *actually* filled. To explain the concept of the **Realogram Analytics**, C showed us a few pictures showcasing the difference between a planogram and a realogram. A realogram, which can be seen in Figure 4.1, highlights misplaced items. Red means that the item should not have been placed in this shelf at all and yellow that it has been misplaced inside the shelf.

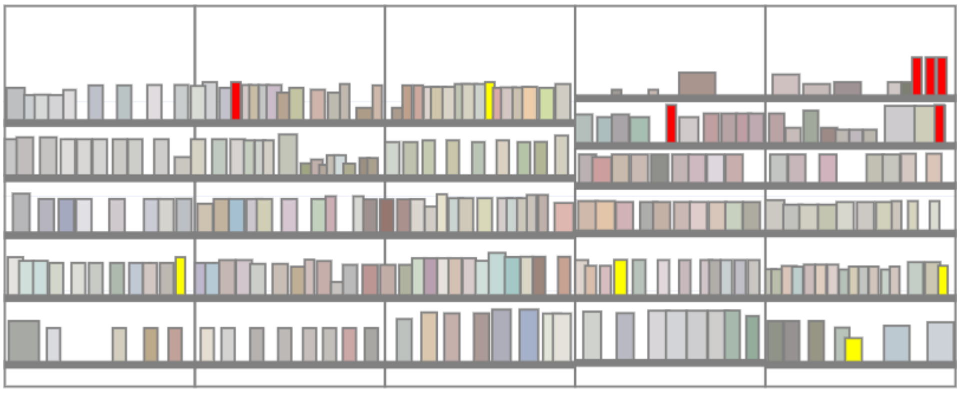

Figure 4.1.: Example of a realogram, showcasing a single shelf. Source: Interview partner C

The realograms are created through the `Store Monitoring` use case and accessed by normal employees. They can use the figures to correct the item placement in the shelves. The `realogram analytics` dependencies with other subcomponents are unclear from the interviews, though the matching with the `list layout` seems apparent. However, the expert on the subject suggested giving the component the same access rights as the list layout.

After taking a look at the dependencies, we take a look at the initial four subcomponents. However, for both the `list layout` and the `customized application`, the involved interviewees all agreed with the presented access control list and had no further comments, which is why we only highlight the responses from the other two.

Sales Data

The sales data is a fairly closed concept and did not find any room for discussion. Our expert on this use case solely touched on its need to also be able to read `Product` and `Shelf`.

Optimization Marketing

For the access rights of this subcomponent, contradictive comments were made. Interviewee C argues that the program does not need as much *read-write* access, excluding `Picking Order`, `Palette` and `Customer`, since those are part of the logistics, designed in the **intelligent intralogistics** use case. The project manager, interviewee A, however, thinks it is reasonable to give the AI program as much material to work with as possible for its workflow. In the end, we need to find a reasonable solution between the two interpretations, just like with the integration of a potential fifth use case with the `realogram analytics` (discussed in subsection 5.4.1).

4.3.3. Intelligent Refrigerator

As discussed in its background (subsection 2.4.7), the intelligent refrigerator serves as a smart shelf and independent point-of-sale. The interaction of the use case with the others was yet unclear. However, it was mentioned that the component has little communication with the rest of the system.

> [The intelligent refrigerator] is a one-shelf store, which means, it writes itself.
>
> (Interviewee D, Uni Bremen; translated)

Therefore, we cannot identify any reasonable dependencies apart from the communication with the ERP Adapter to handle purchases (as mentioned in the ERP subsection 4.2.2). Due to its limited integration in the system, we chose not to integrate a specific expert on the subject, but since it is closely integrated with the ERP system, interviewee F's input was plenty. In terms of its ACL, interviewees D and F argued that the use case has no interest in `Picking Order`, as it has no interconnection with the rest of the product lineup in the store. Furthermore, they mentioned that `Shelf` and `Layout Piece` should be *read-write* to establish the smart shelf capabilities. This would enable a shelf where the planogram can always be up to date due to its sensorial capabilities, making it a smart shelf.

Additionally, the importance of the `Customer` entity was highlighted. This could be used to create a user profile for a payment checkout of the customer or collect user data. However, interviewee F said that none of the `Customer` applications are currently implemented. Checkout cannot be done with a login and no user data can be collected. Still, he mentioned that it could spin up to be an interesting addition.

4.3.4. Service Robotics

The four subcomponents of the **service robotics** each serve their own individual purpose, which is why we will go through the comments on each individually. Nevertheless, there was one general concern that affects all four of them. Interviewee D, one of our experts on the robots, mentioned the concept of a map.

> We have a map of the supermarket. [The different robots] will all use it. That means map creation is something that each of the robots needs. [...] If the robot is on the road and bumps five times accidentally in the same sign, they can just update the map for all robots. (Interviewee D, Uni Bremen; translated)

Hence, each of them needs to have access to a map of some sort that can be updated by any of the robots when they notice changes in the store. Regarding the ACL, this also implies that they all need writing access to the `Layout Piece` to update the store's layout. In general, it also needs to be mentioned that overall the robotic applications are not yet connected to other components and are only used in a *sandbox* environment to test their basic functionality. None of them are meant to be used in practice anytime soon, which made the conversations about their integration into the system fairly hypothetical.

Pick & Place

> [pick & place] is still a utopian idea. If the head of research at Kuka Robotics [...] says: "I don't see that happening in the next 20 years." Then I think the subject is dead. (Interviewee A, dmTech; translated)

The idea of a **pick & place** robot was regarded by all asked interviewees as a research project and nothing more. The technology is far from ready for practical use, especially in an active supermarket with customers walking by. Due to its impracticality, we will not discuss the subject any further and ignore the reasoning behind its ACL.

Autonomous Transport of Goods

As mentioned previously in the **intelligent intralogistics** (subsection 4.3.1), the definition of the `Pallet` entity was connected to some controversy. For the `autonomous transport of goods`, we circulate back to its meaning and what it indicates. The project manager described the process as the transport of any sort of package, which can be *both* a trolley and a pallet. Interviewee B (from the intralogistics use case) mentions that the main purpose of the autonomous transport of goods is not to carry the trolleys inside the store, but solely to move around pallets. Similarly to the store monitoring, this subcomponent supposedly runs during the night to place pallets of goods in the specified aisles, in front of the right shelves, to have an employee fill the shelves in the morning. There were other propositions that also included a combination with the pick & place robots, resulting in a combined vehicle driving through the store and automatically refilling the shelves. However, due to the unrealistic integration of pick & place, we focus on the transport of pallets at night.

Taking a look at its ACL, it was pointed out that to correctly position a pallet in front of a shelf, the robot needs to know the Product area as well as the specific information on a Shelf. Furthermore, since it is solely used to transport pallets and place them somewhere it has no need to write pallets, as reading access is plenty to understand where it needs to be moved to.

Store Monitoring

As previously mentioned, all of the interviewees argued that the store monitoring would be done at night when the store is empty. At the moment, the technology is executed by the external company, Ubica Robotics, and there are no plans in place at the moment to change that.

In terms of its access rights, we previously thought that it only needs to read information. However, as pointed out by multiple interviewees, the component needs to write on Article and Shelf. This enables the process to correctly update the different shelves and their items. Interviewee E further elaborated that the system can only see the facings of each product. Therefore it can only check whether articles of a product are still there, but not how many exactly, resulting in its main purpose: the creation of realograms.

Pepper Assistant

The last subcomponent has a very clear objective and was not met with any confusion during the interviews. Interviewees D and E (robotics experts) touched upon how the robot needs to know where products are even when they are positioned in multiple shelves. Therefore, Pepper needs access to Article and Shelf to correctly navigate the customer to a point in the store. Additionally, just like with the intelligent refrigerator, the opportunity of using Customer data was discussed. Interviewee E discussed the possibility of using anonymized information about the customers on how they interact with the assistant. Our second expert took this one step further, considering saving user specialized information to help the customer with the example of allergies.

> If he has access to customer information, it is of course very, very cool for [the Pepper Assistant]. It could say: "You are allergic to nuts, I am not allowed to give you this product [...]".
> (Interviewee D, Uni Bremen; translated)

Therefore, *read-write* access on the Customer should be considered.

4.4. On-Premise or Cloud

For the hosting style of the K4R platform, we presented three options: deploying the system to every single retail facility, having a centralized system inside a supermarket chain and a fully cloud-based hosting approach, hence external hosting.

The first option meant to deploy the platform to every single supermarket facility separately. In general, all partners said that this is not an option. In the example of dm, it would imply the need for approximately 2000 separate installations of the system. Especially the benefit of branch overlapping analysis would not be possible.

The most preferred option lies in centralized hosting *inside* a supermarket chain. Especially, the dm-related interviewees strongly suggested this method. The biggest presented advantage lies in the ability to further develop the platform and hence be in full control of the system. Secondly, it comes down to how the data is stored, because supermarket chains want to keep their information as safe and confidential as possible.

> [Supermarket chains] will probably say we want that in-house. So I could imagine some kind of mixture. [...] They can imagine the whole thing in the cloud, but they want sovereignty over the data. (Interviewee F, Nagarro; translated)

The way how to enable a secure service for the companies is the key difference between full cloud hosting and corporate internal hosting. As many pointed out, data sharing across different companies through the cloud would not be acceptable in any form. Interviewee A commented that eventually, it comes down to a company's size. A larger retailer, such as Rewe or dm has the manpower to host and administrate their own instance of the platform. Nevertheless, a cloud-hosted version is connected to less maintenance for the company and therefore easier for smaller, more local retailers.

4.5. Manufacture Integration

The last question deals with the potential integration of suppliers and manufacturers into the system. As we learned from multiple interviewees, most supermarkets already provide analytics to the manufacturers. Especially, regarding sales data for individual stores. What would make the K4R integration special, is the completely remote oversight over a manufacturer's product from its current positioning in the store to its actual pricing. The general feedback regarding the proposition was very positive. All of the interviewees agreed that from the perspective of the supplier, it would serve as a great addition to their analytics tools. Interviewee F mentioned that currently, suppliers have to physically walk into the stores to analyze where their products are placed and at what price. The contract may include where the product is positioned or at what price it needs to be sold.

Nevertheless, from the perspective of the retailer, the addition would also have some drawbacks. Interviewees A and E discussed the problem of too much transparency.

> It's a double-edged sword at the moment because not everything is always adhered to everywhere. [...] Whether I want to have this full transparency, to pay

> a contractual penalty right away because the robot has seen something different, that employees have made a mistake somewhere. I don't know if I want that.
>
> (Interviewee A, dmTech; translated)

As the partners have also pointed out the increased transparency would naturally be countered through the service's payment. Therefore, in general, the partners agreed that it could become an intriguing addition for both supplier and retailer, once an equitable arrangement is found.

5. Data Access Control Concept

On the basis of the interview results, this section presents the finalized data access control concept for the K4R platform. The given methodology is described in detail in section 3.2. To arrive at the heart of the concept, we first need to define the different data flows and subjects that need to be controlled in section 5.1. After that, going one layer deeper the objects of the system have to be defined through the digital twin interpretation in section 5.2. Lastly, the different roles are interpreted from the chosen subjects in section 5.3 and access control matrixes are created in form of ACLs in section 5.4. Additional results for the hosting and supplier integration are presented in section 5.5.

5.1. Data Flows and Architecture of K4R

In the context of the interviews, we presented a rough overview of the components (Figure 3.1). This overview has been refurbished and improved to best integrate into our concept. Figure 5.1 shows the final outcome of the component diagram. In the following, we go through every aspect of the diagram and describe its nature.

5.1.1. Use Cases

The largest changes in the diagram are made to the different use cases of K4R.

Intelligent Intralogistics

In terms of the **intelligent intralogistics**, the initial amount of subcomponents has stayed the same, but the communication between each of them has changed. Just like before all four subcomponents require storing information from the digital twin. This results in a link between the use case itself and the store information interface of the digital twin. Through the interviews, we learned that the `optimized stocking strategy` is solely tasked with assisting employees with the refilling of shelves. In order to do so, it does not need to communicate with any of the other subcomponents. Communication with the retail store (digital twin), to know where items of a shelf belong, is sufficient. For the other three, it was assumed that they all depend on the `generic tour planner`. While the `presorting` is still requesting information from the `generic tour planner`, regarding the chosen route. The `support branch commissioning` provides information to the planner. It has been argued that the commissioning could influence the tour planner in its chosen path, by for example placing not only items to refill on the route but also ones to take out for click&collect and similar. The `generic tour planner`'s route is built using the **Hetida Designer**, which is influenced

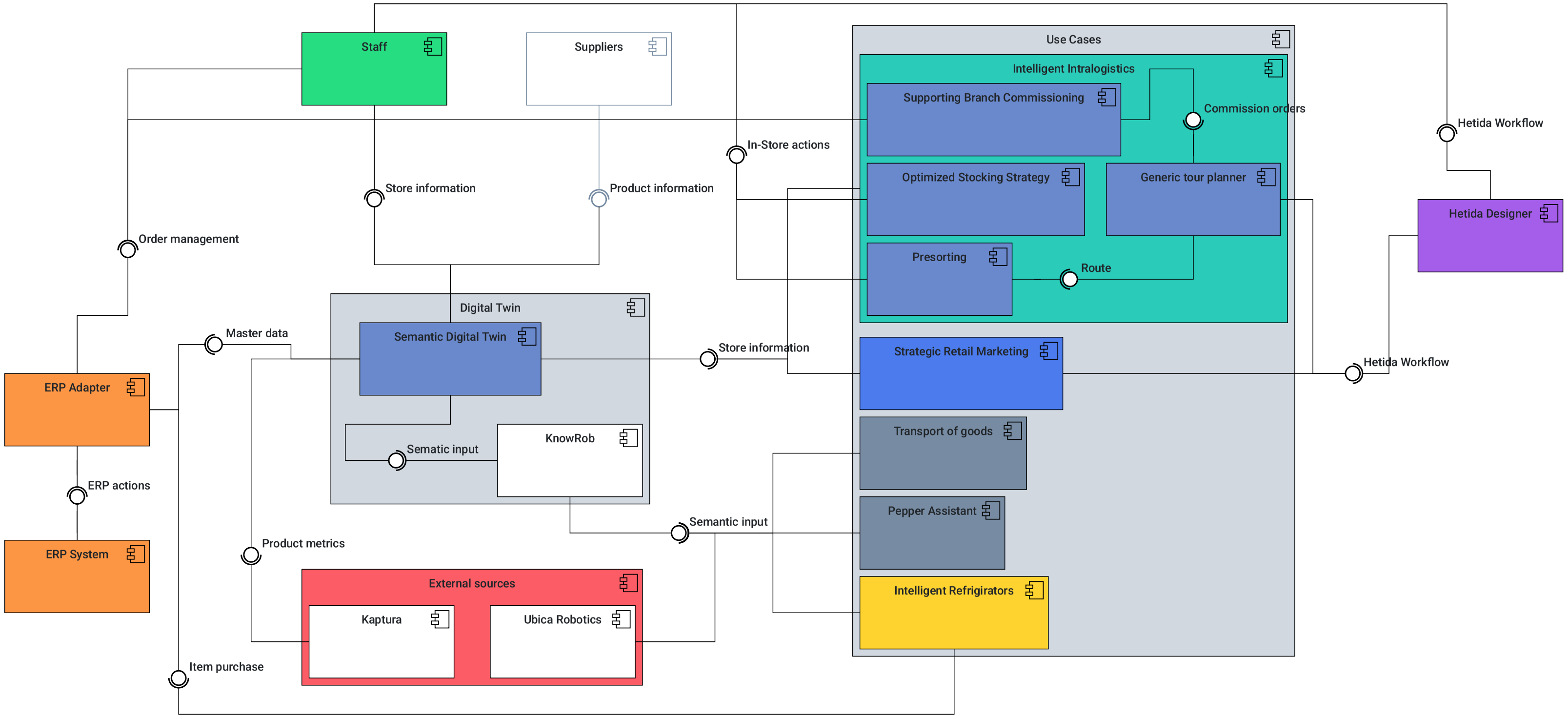

Figure 5.1.: Final component diagram for the K4R platform

through AI on how the optimal path can be determined. In the end, to perform `presorting` the optimal path has to be retrieved to also stock the trolleys in the most optimal fashion. Just like the `optimized stocking strategy`, is the `presorting` directly linked to the `staff`, since employees of the retail store physically perform the actions that the two components suggest.

Strategic Retail Marketing

The **strategic retail marketing** has been summarized to a singular component without any further distribution. The previous four subcomponents (obtained from the early documentation of the K4R project) are all part of a singular process to achieve one goal, namely, to optimize planograms. Every component of this diagram serves as its own individual subject in the access control concept. Since, each of the four components has rather minor exercises to fulfill, such as retrieving sales data or serving as the specialized planogram for the individual retail store from the central version (`custom application`). In conclusion, it is reasonable to fit all of them under one large component to keep the construct of component roles from becoming unnecessarily complex. Especially the two subcomponents `list layout` and `custom application` were closely integrated with the planogram, which becomes its own class in the digital twin in section 5.2, making the subcomponents obsolete. The resulting component requires information from the digital twin to calculate the optimized strategy and create an updated planogram for the store. To do so, it provides the interface to Hetida Workflows. These are on the one hand used to bring artificial intelligence into its methods, but also for staff members to oversee the process and make possible changes to the AI's suggestions.

Service Robotics

Similar to the **strategic retail marketing**, the different robotic applications were not kept in their initial constellation. After the `pick & place` robot was declared impractical for the next 20 years (subsection 4.3.4), it has been removed from the component diagram. Also, it has been established that the `store monitoring` is done entirely through **Ubica Robotics**' devices. Hence, there is no need for an individual component for the `store monitor`. The two remaining robotic services were placed as individual components, alongside **Ubica Robotics**. They each send and retrieve information from the digital twin via semantic input, translated by `KnowRob`.

Intelligent Refrigerator

Just like the three remaining robot applications, the **intelligent refrigerator** communicates with the semantic digital twin through `KnowRob`. Additionally, the smart refrigerator requires an interface to the `ERP Adapter` to handle a customer's item purchasing process through its Point-of-Sale.

5.1.2. External Components

In terms of the external components, **Ubica Robotics** stays as an external component of the K4R platform but acts similarly to the other two robot applications. On the topic of **Kaptura**, there were no newly gained insights through the interviews. It requires an interface to feed the digital twin with product metrics.

One component, that we initially presented in the interviews, is the **path analysis sensors**. This external component was not integrated into the final component diagram. We were not able to find enough information about its exact usage, neither from the documentation nor the interviews.

The ERP integration serves as a larger external component. While the `ERP System` itself only provides access to its data, the universal `ERP Adapter` communicates with the `Staff` and `support branch commissioning` to handle order management. Additionally, it feeds the digital twin with master data and handles the refrigerator's purchasing system.

5.1.3. Staff & Suppliers

The `Staff` component is the only component, which will not be displayed by a single role in the diagram. The number of roles, which are part of the `Staff` would go beyond the scope of this diagram. Nevertheless, the data flow with other components has been identified. As usual, a component requests access to the digital twin's data. Furthermore, certain staff members need to be able to access ERP information as well as others who may manage the **intelligent intralogistics** and **strategic retail marketing** through the **Hetida Designer**.

The concept of integrating `Suppliers` into the system has had a rather positive response. Still, since it is not currently in the scope of the project, the component is currently greyed out. Nevertheless, it would require access to the database of the retail store, even though the kind of access will differentiate strongly from the other presented components.

5.1.4. Digital Twin

The already mentioned `KnowRob` component still lies inside the digital twin-component as an additional processing tool with no changes regarding its data flow. Just like before it is used to translate the semantic input from the robots and the refrigerator.

Bringing everything together, the digital twin sits in the center of the architecture. Since we moved away from the conceptional entities to a class presentation of the semantic digital twin, the entities are no longer part of the diagram. Instead, the `Semantic Digital Twin` stands in its place and provides interfaces for the surrounding components.

5.2. Semantic Digital Twin

After discussing the component overview of the system, we can take a closer look at the semantic digital twin itself. The component is replicated by a number of interconnected classes, which serve as the access control's objects. The finalized diagram can be found in

Figure 5.2. In the progress of explaining the different classes, we take a rough clockwise route, starting with the Store.

The Store class is the heart of the digital twin. It includes meta attributes about the retail branch e.g. its Id, where it is located, or its name. It serves as the main access point for other components to clearly identify what store access is requested.

Moving left the Shelf is placed as an aggregation of a Store, since one Store has a number of Shelf object which can also exist without the Store. It holds information about its size and its position in the facility. The position is declared as a 6DPosition, a universal class describing an object's three-dimensional position and orientation. Inside a Shelf lie a multitude of ShelfLayer objects, which can be individually identified and used through their level, position, and size. One more level down the line, inside a ShelfLayer we have a number of Facings. A Facing has a relative position in a ShelfLayer with its width determined by the number of its Item. However, a Facing is not simply part of a ShelfLayer as it can occur in multiple places throughout a store (resulting in the one-to-many multiplicity).

A Facing represents the front row of numerous Item objects. Each individual Item has a relative position in its Facing and belongs to a Product. A Product has a large number of attributes differentiating it from others. Due to the model's conceptual nature, only a minimal amount of information attributes are included as examples, like its name or size. A Product is an aggregation of an Assortment. An example of such would be shower articles. An Assortment is typically placed on a single aisle or Shelf.

Moving away from the visible storefront, previously the assumption has been made that pallets serve as a major access point, due to their need in various use cases. In the end, the name has been changed to Trolley. A Trolley object has a current position in the store as well as a type. The type identifies, whether an actual supermarket trolley is moved around or a pallet is moved without being presorted onto a Trolley beforehand. Therefore, a pallet when moved around the store to refill shelves will be handled as a special instance of a Trolley.

A Trolley is filled with a multitude of DeliveredItem objects, a different kind of Item that is solely used as a delivered instance. Unlike Item, a DeliveredItem represents an entire packaging unit ("Kollo") and thus holds an amount of a Product. It has special attributes regarding its positioning on a Trolley as well as whether it has been sorted onto a Trolley or placed as a whole.

The remaining classes are more use case-specific. On the far right, the Customer is placed. A class that has been added to work in connection with the **intelligent refrigerator** to identify the person interacting with it. The identified person may get products assigned to it, depending on their take-out from the refrigerator. These products are stored in a ShoppingBasket with their corresponding amount and price to calculate the overall pricing of the basket. Additionally, through the interviews, the idea has been introduced to collect user data to improve their experience, but also help the AI applications improve the decision making on regional characteristics. A prominent example was with the pepper assistant robot, which could warn a customer when logged in about their allergy risks with certain products.

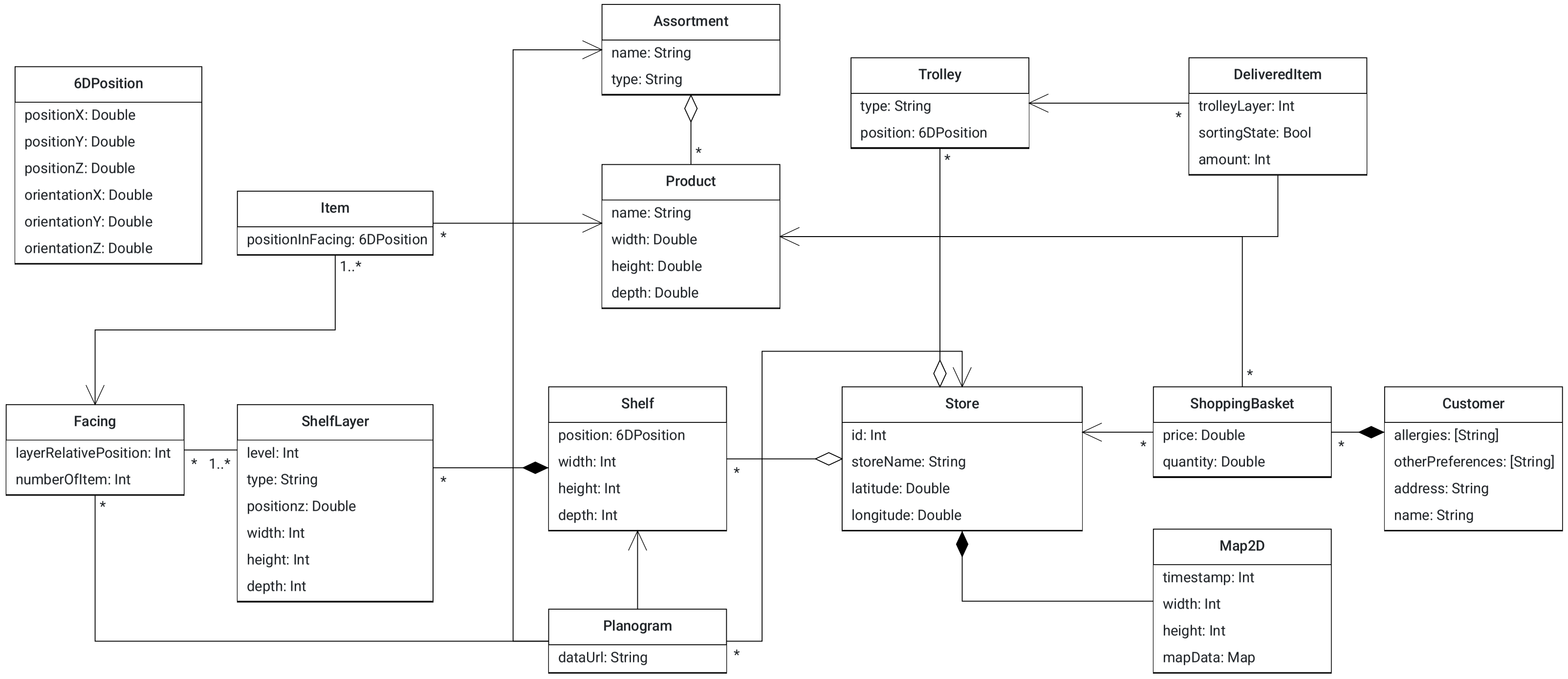

Figure 5.2.: Conceptual class diagram of the semantic digital twin

In order to navigate the store, a Map2D class has been added. Every Store needs to have some form of a map on which the robots can move around. Similarly, the generic tour planner uses it to plan its routes throughout the Store. As the different components need to read their properties regularly, it needs to be identified with a timestamp to evaluate the accuracy of the Map2D instance. As learned from the interviews, the layout of the store may change constantly (subsection 4.3.4). Its information is stored in some form of data. The type is insignificant for this thesis and hence simply marked as mapData.

Lastly, we introduced the Planogram class, positioned beneath the Shelf. Every Shelf needs a corresponding Planogram, describing how its Facings should be positioned in the corresponding Assortment. For the sake of simplicity, the planogram's attributes only include a *dataUrl*. Similar to the Map2D, the exact type of data is not relevant for our research. It brings the relationships with the three other classes together. The resulting object needs to be displayable and readable for employees of a Store, while also being machine-readable for the different use cases to process.

This concludes all objects of the retail store for the access control concept.

5.3. Access Roles

The following presents the individual access roles of the system. The section is divided into two parts, access roles for the digital components and for the human staff. The methodology behind the access role definition can be found in subsection 3.2.3.

5.3.1. Digital Roles

With the help of the UML component diagram, all roles on the digital side are already established. The corresponding roles have to be placed in role hierarchies. Due to the largely differentiating function of each component, it mainly serves as an overview of the roles. Only in the case of the **service robotics**, a basic role has been introduced to serve the basic access, that all of the robots require. The basic need to interact with the Map2D of a store, called for the needed introduction of a Robot role. Additionally, this also leaves the door open for other robotic applications. It provides the essential access rights to any future additions, such as the yet unrealistic pick & place robot. The overview is shown in Figure 5.3.

5.3.2. Staff Roles

On the Staff side, a rather common role hierarchy is introduced. The final illustration of this hierarchy can be seen in Figure 5.4. Through the results from the interviews, two main sectors of staff roles have been established: The **Store** staff and the **Headquarters** staff. Outside the two sectors, we placed the role of the **system admin**. A role that is necessary to alter almost anything, when the system does not behave as expected.

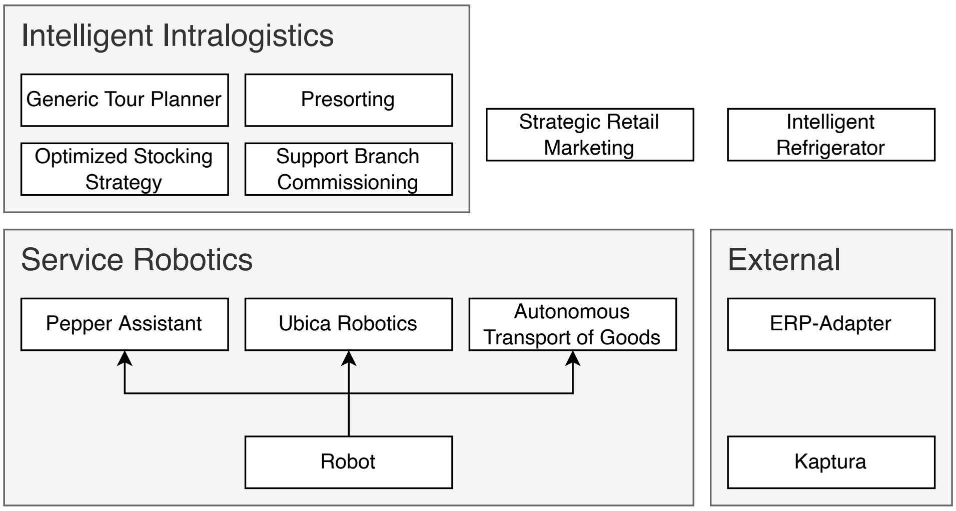
Intelligent Intralogistics
Generic Tour Planner
Presorting
Optimized Stocking Strategy
Support Branch Commissioning
Strategic Retail Marketing
Intelligent Refrigerator
Service Robotics
Pepper Assistant
Ubica Robotics
Autonomous Transport of Goods
Robot
External
ERP-Adapter
Kaptura

Figure 5.3.: Access role hierarchy of the digital components

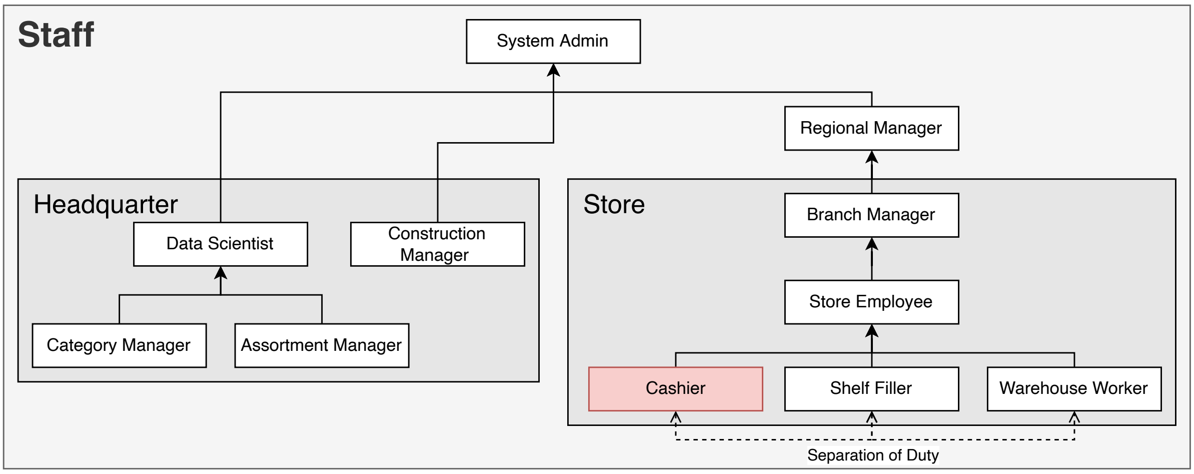

Figure 5.4.: Access role hierarchy of the staff

Store

Inside a supermarket branch, it has been established that there is a rather flat hierarchy. In a retail store, there are three main jobs that are done. Depending on the size of the store, the range of what a regular `Store Employee` does differs. In a supermarket facility, a `Store Employee` usually fulfills all the tasks of a `Cashier`, `Shelf Filler` and `Warehouse Worker` in a day of work. The differentiation between these three roles is essential, since an employee, working in the storefront to refill shelves, does not need access to information only relevant for the warehouse, which brings us to the separation of duty (subsection 2.1.3). The `Shelf Filler` and `Warehouse Worker` roles each have respective use case components that they interact with (`optimized stocking strategy` and `presorting`). The `Cashier`, on the other hand, has no direct impact on the system. Outside of Knowledge4Retail, the cash registry already directly communicates with the ERP application, thus a `Cashier` does not need any access rights on the system (resulting in the red coloring).

The three subroles were not only introduced to better control what an employee is allowed to do while working in the different areas. In addition, it enables the incorporation of work contractors, who are specifically hired to refill the shelves or work in the warehouse. In the same sense, bigger retailers might hire an external company specifically for one of the sectors, such as managing the warehouse. Consequently, a `Store Employee` has the option to do all the roles freely, but when doing one, the other two are inaccessible to prevent forms of accidental overlap.

The `Branch Manager` sits at the top of the store's hierarchy. He/she inherits all access rights of the `Store Employee` with quite a few additional permissions to manage his/her facility.

On top of the `Branch Manager`, `Regional Managers` may be placed who have a certain amount of stores that they are allowed to oversee. The number of layers of `Regional Manager` can differ from any supermarket company. dm has none, but others might have regional, state, and national managers, each overseeing a larger amount of stores.

Headquarters

Any supermarket chain needs a headquarter from where centralized decisions can be made. From the interviews, we have learned that dm controls most of its day-to-day activities from its headquarters.

On this account four roles were mainly discussed (subsection 4.1.3):

- The `Assortment Manager` evaluates the item placements in a branch according to its needs.
- The `Category Manager` creates the planograms for the stores.
- The `Data Scientist` analyzes information.

In the role hierarchy, he/she is placed above the two, due to the fact that the `Data Scientist` needs a great deal of access, going beyond what the other two may retrieve.

For the setup of new facilities of a company, an additional role is needed. The `Construction Manager` takes into account the interviewees' suggestions to give construction workers a writing access to initiate a new store. The construction and initiation could have been split up into a multitude of roles. One role that oversees the construction and installation brings more simplicity to the process without too many moving and interacting subjects (or actors).

5.4. Access control lists

The ACLs bring all previously presented results together to determine what a role should access (according to subsection 3.2.4). Similar to the last section, the final access control lists are split into the two sectors of roles: digital components and staff. The objects for the final access control feature the various new classes, but also other components of the system that have data flow between one another. Each object's name is abbreviated for the individual ACLs. Some objects were summarized in the model as the access of one implicates the access on the other. This is the case for `Shelf` and `ShelfLayer`, as well as for `Customer` and `Shopping Basket`.

5.4.1. Digital Components

The final list is created with the final set of roles and objects for each sector of components, found in Table 5.1.

Most changes from the initial ACLs are made due to changes in the subcomponents' interpretation in section 5.1. Additionally, some of the entities were renamed and reimagined through section 5.2. The `Layout Piece` itself no longer exists, instead its data lies in the `Shelf` and `Map2D`. Furthermore, the `Picking Order` no longer exists, but is part of `ERP`, hence when needed an association directly to the `ERP Adapter` is required. Lastly, the `Order Unit` has been renamed to `DeliverdItem` to fit the project's latest documentation.

Intelligent Intralogistics

The initially chosen subjects are in line with the declared final roles. Therefore, solely the newly retrieved set of objects needs to be integrated into the ACL, as seen in Table 5.1.

Due to the more passive interpretation of the `optimized stocking strategy`, the component has no writing access. To fulfill its task it needs mostly information to recognize products and where they should be placed in a `Shelf`. This results in the *read-only* access on the `Planogram` as well as the `Product`, `Shelf` and `Trolley` related objects.

For the `generic tour planner`, the reading access on the `Map2D` is added. In addition, on top of the `Product`, *read-only* access on the `Assortment` is required to locate the items in their respective product assortment in the store.

The `Presorting` component is the only active (writing) subject. It takes the suggestions from the planner and sorts the `DeliveredItems` accordingly, hereby the `DeliveredItems` are written on the `Trolley`. Additionally, the position of those new `Item` objects in the store is changed ahead of the actual refilling. Due to the fact that the `optimized stocking strategy`

Role	St[a]	Pr[b]	As[c]	It[d]	Sh[e]	Fa[f]	Tr[g]	DI[h]	Ma[i]	CS[j]	Pl[k]	ERP
Optimized stocking strategy	r	r		r	r	r	r	r			r	
Generic tour planner	r	r	r	r	r	r			r			
Presorting	r	r		rw	r	r	rw	rw				
Support branch commissioning	r											r*
Strategic retail marketing	r	r	r		rw	rw			r	r	rw	r**
Intelligent Refrigerator	r	r		r	rw*	rw*				rw		a
Robot	r	r	r						rw			
Autonomous transport of goods					r	r	r	r				
Ubica Robotics				rw		rw					rw	
Pepper Assistant				r	r	r				rw	r	
ERP Adapter	r	rw	rw					rw				
Kaptura		a										

Table 5.1.: Final access control list for the digital components

[a] Store
[b] Product
[c] Assortment
[d] Item
[e] Shelf & Layer
[f] Facing
[g] Trolley
[h] DeliveredItem
[i] Map2D
[j] Customer & Shopping Basket
[k] Planogram

is not writing anything, but only assisting. The correct refilling of the employee is expected but cannot be controlled (Interview results, subsection 4.3.1).

The rights of the *support branch commissioning* role are displayed in a very simple mane. It needs to know the store the commissioning order is for. In addition, the "r*" translates to *read-only* access on the `ERP Adapter`, but only for the commissioning orders.

Strategic Retail Marketing

In the case of the **strategic retail marketing**, the subjects, as well as the objects of the access control matrix, have changed. Since the `list layout` and `custom application` became obsolete, the initial subcomponents of `sales data` and `optimization marketing`'s access rights were combined to form a unified role.

The `optimization marketing` collects all the data to analyze most accurately how the store planograms could be optimized. To do so, it has read-only on all classes except for the delivery related ones (`Trolley` and `DeliveredItem`). From the ERP Adapter, it reads the turnover data of the past 60 days (as the *sales data* subcomponent does), which is marked with the "r**". At the end of a finished analysis through the `Hetida Designer`, the `Planogram` is updated accordingly. This might also bring changes to the positioning of a `Shelf` and its `Facings`. In the course of the interview, interviewee C mentioned the need for a **realogram analytics** use case in subsection 4.3.2. Through the proposed updatable nature of the `Planogram`, the changes to the `Planogram` are integrated directly and do not need to be displayed on a separate instance.

Intelligent Refrigerator

The concept of the refrigerator has always been rather simple without leaving much room for discussion. Accordingly, we solely updated the **intelligent refrigerator** to the new set of objects. The refrigerator mainly does two things, sales and self-updating its inventory.

Firstly, it needs to create a `Customer` or read and update its information if the user already exists and manage its `Shopping Basket`. On top, it can *append* checkout processes to the `ERP Adapter` once the `Customer` proceeds to checkout. Secondly, the **intelligent refrigerator** reads `Product` and `Item` information and has "rw*" on `Shelf` and `Facing`. In this case, we are looking at *read-write* on its own shelf. This means that it can alter the placed items inside itself depending on the ownership of the `Shelf` and `Facing`. This adds an aspect of discretionary access control (subsection 2.2.2) to the "smart shelf"'s access role, where it can only access the ones that belong to itself.

Service Robotics

Service robotics is the only use case, where taking advantage of a role hierarchy has proven reasonable. For the individual robot roles, the rights already provided through the basic `Robot` role have been marked grey. The basic `Robot` role provides the basic access rights all robots should have such as the information about the `Store`, `Products` and `Assortments` to recognize products in their respective area. Additionally, any robot needs to be able to *read-write* the Store's Map2D to find its way around and update any sudden changes in the environment (such as special offers standups in the storefront).

The *autonomous transport of goods* moves Trolleys to their designated refill location. Thus it needs to know in addition to the basic rights where specific DeliveredItems need to be placed, regarding the Facing and corresponding Shelf and ShelfLayer.

The externally integrated Ubica Robotics robots roll through the store, updating the Facings of the facility. Therefore, it needs *read-write* access on the Item and Facing to update its position if necessary as well as the corresponding Planograms.

Last but not least, the *pepper assistant application* needs to know where what is to show a Customer the way. This also extends to the individual Items as it also needs to know whether it is still in place or sold out entirely. One optional access right for the future lies in the writing capability of the Customer, which would enable more extensive data to work with, to assist customers, and drive sales with the **strategic retail marketing**.

External components

The two external components the ERP Adapter and Kaptura serve supportive positions. The ERP system provides all of the necessary store-specific master data of the system through the ERP Adapter, with regards to Product and Assortment. In addition, it initiates the delivery to the store by providing new DeliveredItems.

Kaptura updates the provided Product information from ERP with sizes and similar metrics. Notably, Kaptura has no *read-only* access on the Store as the Product metrics cannot be regionally or store-specifically specialized. Instead, the system *appends* metric information to the existing Product objects.

5.4.2. Staff Access Control List

For the ACLs of the staff, individual fragments are presented separately. In light of the Staff component's data flows to other components apart from the Digital Twin, we only show the objects the respective subjects interact with to avoid displaying 17 columns on a table. Throughout this section, we present the access rights of the store working our way up in the hierarchy, and start over in the headquarter. The role of System Admin, who stands outside the two areas in Figure 5.4 has all the rights possible to change any flaw imaginable.

Store Employee

Regarding the lowest level, the Cashier has no access rights, thus only the Shelf Filler and Warehouse Worker are relevant for the ACL, shown in Table 5.2. The role of a Shelf Filler mainly needs to interact with *the optimized stocking strategy* to refill shelves more efficiently. Additionally, it has been discussed that an employee should be able to check the Planogram objects in the Store. Since this validation process should only come while working in the storefront, it has been placed on the role of a Shelf Filler. To better work around the

Role	Trolley	Delivered Item	Planogram	Optimized Stocking Strategy	Presorting
Shelf Filler	read-only		read-only	execute	
Warehouse Worker		read-only			execute

Table 5.2.: Access control list for the jobs of a Store Employee

stocking strategy, the decision has been made to give the `Shelf Filler`, additional insight into the `Trolley`.

The `Warehouse Worker`, on the other hand, only needs warehouse-related subjects. This includes what is coming in the store on a day through `DeliveredItem` objects and evidently the execution of the `Presorting` use case.

The reasoning behind the inclusion of the `Store Employee` role is that he/she should be able to seemingly switch between the different activities. In contrast, any form of contract worker must not be able to switch his set of access rights.

Branch & Regional Manager

The `Branch Manager` is meant to have overruling rights. In the case of dm, being able to interfere and change decisions made by the headquarter, regarding discounts and pricing. For this reason, the following ACL is created. In addition to what can be seen in Table 5.3, he/she may also access the rights of the `Store Employee` in Table 5.2, due to the inheritance of lower roles.

Role	Store	Product	Shelf & Layer	ERP Adapter	Store Employee
Branch Manager	read-only	read-only	read-only	read-write*	control

Table 5.3.: Access control list for the Branch Manager

The `Branch Manager` should be able to read additional information that the employee cannot. This may include information about the `Store`, the `Products`, and the `Shelf` and `ShelfLayer`. More importantly, to hire new people, he/she should be able to assign new `Store Employee` roles (through the *control* access right), as well as roles specifically for one of the three subsections of the `Store Employee`. And most of all, the manager needs to manage the ordering of products as well as the altering of prices through a special "*read-write**" on the `ERP Adapter`, which restricts the access to solely orders of his/her own branch.

A `Regional Manager` would have the same overseeing access on his/her set of branches. The role would allow to hire and assign `Branch Manager` roles, control order management and pricing for his/her region, and everything the `Branch Manager` may access (on all of his/her branches).

Headquarter

Moving over to the headquarter of a supermarket chain, we start off with the analytical roles. The respective access control list is shown in Table 5.4.

Role	Store	Product	Assortment	Planogram	ERP Adapter	Hetida Designer
Assortment Manager	read-only	read-only*	read-only*	read-only	read-only*	
Category Manager		read-only	read-only	read-only		read-write
Data Scientist	▨	▨	▨	▨	read-only	▨

Table 5.4.: Access control list for the analytical roles

The role of the `Assortment Manager` needs all the necessary information to analyze the performance of his/her responsible assortment. Therefore, the role needs `Store`-specific access on the corresponding `Products` of his/her `Assortment` as well as the current `Planograms` of the facility. Most importantly, the expert of an `Assortment` should be able to analyze the performance of its product through the `ERP Adapter`. Since the role should only have access to the performance of its own `Assortment`, the *read-only* access right is marked with a "*" (likewise for the `Product` and `Assortment` object).

The `Category Manager` takes the suggestions into account from the various `Assortment Managers` to update the `Planogram`. The `Planogram`, however only has *read-only*. This is due to the fact that the `Planogram` is updated and changed through the `Hetida-Designer`, where accordingly a `read-write` access is necessary to make an impact on the **strategic retail marketing** through the Hetida workflows.

While the `Data Scientist` does not have a higher position in the company, hierarchically speaking, the role needs all the same rights the other two do. The only difference is the full reading access to the different classes to analyze the information from K4R.

The last role of this concept needs all the necessary information to create and administer a new facility as the `Construction Manager`. This involves being able to have writing access on class objects that should not be changed through staff members normally as seen in Table 5.5.

Role	Store	Shelf & Layer	Map	Robot
Category Manager	read-write	read-write	read-write	read-write

Table 5.5.: Access control list for the Construction Manager

He/she sets up the basic `Store` as well as all the `Shelf` objects and the `Map` for the new `Store`. Additionally, through this role, one would be able to initialize the various `Robot` applications that are applicable for this facility.

5.5. Additional Findings

For the integration of a `Supplier` role, we propose access rights that would let them see information *only* relevant to their products. This would include *read-only* rights on their own `Product` and where their `Item` objects are positioned in the store, as well as the `Planogram` holding their `Product` instances and the performance of the `Product` through a restricted ERP-Adapter access.

Apart from the role-based access control concept itself, we also discussed the hosting situation. On this basis, we now propose a possible solution. Through the interviews' results (section 4.4), the most feasible style has to be differentiated between the size of the company. For larger companies, such as dm or Rewe an internal hosting approach is more advantageous. It gives the company full control over the system and allows them to enhance and change the system according to their needs. Furthermore, it ensures that **their** data stays on their **own** servers.

For a smaller supermarket chain, only consisting of a few stores, a cloud approach is more realistic. Through this Knowledge4Retail would take care of hosting and initializing the store instances and each company would have their own login into their "K4R account".

6. Discussion

With finished elements of a data access control concept established, we take a critical look at the accomplishments of the thesis. For that, the evaluation of the concept with the two experts is discussed. Afterward, we use the insights from the evaluation interviews to discuss the drawbacks and limitations of our research.

6.1. Evaluation

The two evaluation interviews were held with two separate areas of expertise in mind. One interviewee with a more technical implementation-focused background (Interviewee G) and the other had a more profound background in the structures of a supermarket (previous Interviewee A, working for dmTech). A more detailed definition of the evaluation interviews' approach is in Section 3.3.

Regarding the component diagram, interviewee G was generally in line with the chosen data flows necessary to answer **RQ1**. Nevertheless, he pointed out that he does not agree with the naming of the component of the digital twin. The semantic digital twin should not be the inner subcomponent, represented through the UML class diagram (Figure 5.2). He prefers to name it the digital twin and the outer part the semantic digital twin. He argues that the *semantic* section of the twin is translated by the `KnowRob` component. As we established in Section 2.3.1, the semantic digital twin serves as a specific layer of a digital twin, resulting in the current approach in Figure 5.1. A solution for a more precise integration of the digital twin-component would be to completely separate `KnowRob` from the digital twin. This would result in two new separate components with the `KnowRob` representing the semDT and the component `Digital Twin` as the actual database (depicted in Figure 5.2). Hence, it would eliminate the need for an outer component of the digital twin.

For the digital twin itself, interviewee G was mainly in line with the presented classes of a supermarket but says that it is a simplification of the actual implementation (as it is meant to serve as access objects). Concerning the class of the `Customer`, he mentioned that it was solely included for the **intelligent refrigerator** to handle transactions. In his case, the class should only store anonymized information. In regards to various results from the expert interviews, it has been determined that a more data-driven `Customer` will serve useful for the future of the project.

On the digital side of the roles, there were minor suggestions. For the role of the **strategic retail marketing**, interviewee G argued that it should have no writing access on the physical shelves. Accordingly, it should not be able to change the physical layout of the store, but give suggestions to other human roles which could then execute a change in the layout and accordingly in the `Map2D`. Furthermore, in the case of the `ERP Adapter`, the access on the `Store` itself should be with some sort of writing access to also provide meta-information on a `Store`.

Interviewee A agreed with the majority of created elements of the concept. In terms of the staff roles, he agreed with all the presented roles. Nevertheless, C added that the inclusion of a logistical manager role on the headquarter side should be considered. The logistical manager would be someone who can oversee the deliveries to manually optimize processes in this regard. As interviewee A also pointed out, the rights of someone overseeing the logistics should not be very different from the role of the Category or Assortment Manager. This begs the question of whether there is a need for a specified role. It would certainly be more precise to design a specified role for this job task, but due to the research's focus on the In-Store focused use cases of Knowledge4Retail, this will not be further elaborated.

In addition to the data access control, the chosen hosting approach has been validated. While interviewee A was in line with the approach for bigger retailers such as dm (his employer), interviewee G argued that the solution for smaller retailers is not practical. According to his vision, a retailer will never have their K4R application running on servers hosted by Knowledge4Retail. Instead, the servers of the system would be hosted by big cloud providers, such as Microsoft, Amazon, or Google. Large chains would have full control over the program with in-house administration and implementation. On the other hand, K4R would only serve as a consultant partner for smaller supermarket chains, providing setup assistance and support.

6.2. Limitations

One great factor in the creation of the concept was the influence of different interpretations from the numerous partners. Knowledge4Retail is an interdisciplinary research project. Thus there are numerous partners and opinions on the final creation of K4R. As seen through the various interviews, this also created a challenge for the creation of a data access control concept for the platform. Each of the four use cases is managed by a different team. In this research and especially in **RQ1**, the goal was to define clear data flows between the different organizations and roles. To do so, the different interpretations of the various subcomponents were put into perspective to create a unified system. This problem was found in the research on the requirements for the concept in regards to **RQ2**. Since all the use cases of the platform are in a purely research-focused state, an accurate interpretation of what each component should be able to accomplish was demanding.

This may also be seen, regarding the conducted evaluation interviews. Taking the example of the **strategic retail marketing**, it leaves room for discussion, on whether the program should be able to write a specific type of object or not. The issue carries over to the hosting approach where interviewee G has pointed out that the previously thought style of cloud hosting is not applicable. It is something that none of the other six interviewees were aware of when we first discussed the hosting style.

The previous two examples give valuable insight into one of the major limitations of this research. The interview partners were chosen to have expertise from all the major departments of the project. Almost all experts were not necessarily connected to the implementation side of the project. On one hand, this benefited the insights on the more theoretical side, how a supermarket functions and how the system's use cases *could* be integrated. On the other hand, having a great extent of interviewees implementing components of K4R would have presented more information on how the components *should* be integrated.

Potentially, another limitation lies in the focus on dm. For the selection of the staff roles, the goal was to create a universally applicable set of roles. Through the course of the interviews, we attempted to keep a generalized view of the supermarket's infrastructure. Nevertheless, a slight nudge towards the infrastructure of dm seems inevitable. Out of the six interviewees, three came from dm. The hierarchy on the headquarter side of the roles is influenced by dm but was designed with a general perspective in mind since all the roles' activities should be fulfilled in any supermarket corporation. In addition, the role of the `Branch Manager` was designed to have overruling rights from the headquarter's decisions to individually change the pricing and order management. This may be something that is not common for any supermarket chain. With a more diverse set of interviewees from different supermarket chains, the final list of staff roles might have a different standing, potentially more universal. Apart from this, the selection of interviewees was also restricted to German participants, since K4R is a German project. Accordingly, the defined roles might only be applicable in German supermarket chains.

In terms of the evaluation process, it could be argued that a larger pool of interviewees would have led to a more compelling evaluation. However, for the two performed evaluation interviews, partners were chosen with great expertise in their respective areas of the project to provide a higher quality of responses. A larger number of interviewees might bring up more flaws of the concept, but due to the previously specified problem of interpretation, this could lead to misleading statements for the access control concept.

In the initial research process, we strongly relied on the documentation from Confluence. This led to some complications in the creation of the initial results for **RQ1**. Only through a broader understanding of the ecosystem, the initial misinterpretations were corrected. Most notably, the initial set of nine entities of the digital twin, which came from the project's documentation, had to be discarded to establish the needed objects for the access control concept with the necessary precision.

In addition, components, initially thought to be important, were discarded upon a better understanding such as the **path analysis sensors** through little to no intel from the interviewees. Similarly, the various subcomponents of the **strategic retail marketing** were removed due to their strongly overlapping functionality.

7. Conclusion & Future Work

Throughout this research, we pursued a single objective, the creation of an access control concept for the Knowledge4Retail platform to provide confidentiality and integrity for the corporation's data. The research objective was divided into three separate research questions. In **RQ1**, we introduced the data flows between the various organizations and roles. With the help of the conducted expert interviews and prior research on the documentation of the project, a diagram has been created. It showcases how the different organizations and roles intertwine. In the center, is a digital replica of a supermarket store with all the various components operating on its continuously updating data. The objective of **RQ2** was to collect the requirements of retailers and partner organizations. This resulted in clear requirements for certain elements of the concept, but also some controversial opinions on others.

Together, the results from **RQ1** and **RQ2** built the base for the final third research question. In the course of designing elements of a data access control concept, the objects, as well as subjects in form of roles, were defined. For each role a set of access rights in form of access control matrixes was created, depicting what and how they are supposed to access different parts of the system, notably separated into digital and staff roles. With the clearly defined roles and objects in mind, the thesis provides a role-based access control concept for the K4R platform. In addition, the thesis established a methodology on how a role-based access control system for complex corporate platforms such as K4R could be pursued.

To build on the prior research, the thesis included two additional objectives in the expert interviews in form of the hosting style and supplier integration. Both concepts were mildly touched upon, but need further research. For the hosting style, we provide a proposal for companies in the supermarket chain domain with regard to the company's size. With the additional insights from the evaluation interviews, the research establishes a reasonable foundation for the implementation of a hosting approach for the platform. In light of the second additional theme, research has shown that the potential supplier integration could establish an intriguing addition. Through its uniquely analytical capabilities, it may serve as a highly captivating extension to the K4R platform.

The proposed elements of a data access control concept build the foundation of a future implementation of access control on the K4R platform. With the evaluation process in mind, future research could improve on the currently created concept. Additionally, with a broader set of interviewees, the accuracy of the presented elements might be increased. This also includes other elements of access control, which were not investigated as part of this thesis. Role-based access control's fourth component (apart from the `Users`, `Rules`, and `Permissions` from Section 2.2.3) defines an integration of a finite set of `Sessions`. Beyond the scope of this thesis, this could be explored to complement the established elements and provide a solution for the management of user sessions in the system who take on the research's defined roles. Ahead of the implementation of the access control concept, the research also presents a complete integration of all organizations into one system, which is an integration process that has not been done yet in the scope of the interdisciplinary research project.

In conclusion, the presented research creates a foundation for research in the Knowledge4Retail ecosystem, primarily in the creation of a data access control concept but also IT security in general for K4R.

A. Addenda

A.1. Updated entity-based access control lists

Red marks deleted access rights, green newly added access rights.

Intelligent Intralogistics	Pr	A	LP	Sh	St	PO	Pa	OU	C
Optimized stocking strategy	r	rw/ r	r	r	r	rw	r	r	
Generic tour planner	r	r	r	r	r	rw			
Presorting	r	rw	r	r	r		rw	rw	
Support branch commissioning	r	r	r	r	r	rw			

Table A.1.: Entity-based access control list for the intelligent intralogistics

Strategic retail marketing	Pr	A	LP	Sh	St	PO	Pa	OU	C
Sales data		r	r		rw				
List layout	r		r	r	r				
Customer application			rw	rw	rw				
Optimization marketing	r	rw	rw	rw	rw	rw	rw		rw

Table A.2.: Entity-based access control list for strategic retail marketing (no changes)

	Pr	A	LP	Sh	St	PO	Pa	OU	C
Intelligent refrigerator	r	rw	r	r	r	rw		rw	rw

Table A.3.: Entity-based access control list for the intelligent refrigerator

Service robotics	Pr	A	LP	Sh	St	PO	Pa	OU	C
Autonomous transport of goods	r	rw/ r	r/ rw	r	r		rw/ r		
Store monitoring	r	rw	r	r/ rw	r				
Pick&Place	r	rw	r	r	r				
Pepper assistant	r		r		r				rw

Table A.4.: Entity-based access control list for service robotics

List of Figures

List of Tables

Bibliography

[ins22] neusta analytics & insights GmbH. *Hetida Designer*. 2022. URL: `https://github.com/hetida/hetida-designer` (visited on 03/13/2022).

[And20] R. Anderson. *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons, 2020.

[Bee+18] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoğlu, and G. Bartels. "Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 512–519.

[BL73] D. E. Bell and L. J. LaPadula. *Secure computer systems: Mathematical foundations*. Tech. rep. MITRE CORP BEDFORD MA, 1973.

[BFF96] T. Berners-Lee, R. Fielding, and H. Frystyk. *Hypertext transfer protocol–HTTP/1.0*. 1996.

[Böt+21] T. P. Böttcher, L. Rickling, K. Gmelch, J. Weking, and H. Krcmar. "Towards the Digital Self-renewal of Retail: The Generic Ecosystem of the Retail Industry". In: *International Conference on Wirtschaftsinformatik*. Springer. 2021, pp. 140–146.

[Bus+02] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *A System of Patterns: Pattern-Oriented Software Architecture, volume 1 of Wiley Series in Sotware Design Patterns*. 2002.

[Cos+20] M. Costanzo, S. Stelter, C. Natale, S. Pirozzi, G. Bartels, A. Maldonado, and M. Beetz. "Manipulation planning and control for shelf replenishment". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1595–1601.

[Cru+08] I. F. Cruz, R. Gjomemo, B. Lin, and M. Orsini. "A constraint and attribute based security framework for dynamic role assignment in collaborative environments". In: *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer. 2008, pp. 322–339.

[Die04] R. Dierstein. "Sicherheit in der Informationstechnikder Begriff IT-Sicherheit". In: *Informatik-Spektrum* 4.27 (2004), pp. 343–353.

[DHP94] X. Dreze, S. J. Hoch, and M. E. Purk. "Shelf management and space elasticity". In: *Journal of retailing* 70.4 (1994), pp. 301–326.

[DS10] L. Dusseault and J. Snell. *Patch method for http*. Tech. rep. RFC 5789, March, 2010.

[Eck18] C. Eckert. *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. De Gruyter Oldenbourg, 2018. ISBN: 9783110563900. DOI: `doi:10.1515/9783110563900`. URL: `https://doi.org/10.1515/9783110563900`.

[Eig20] M. Eigner. “Digitaler Zwilling–Stand der Technik”. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 115.s1 (2020), pp. 3–6.

[FKC03] D. Ferraiolo, D. R. Kuhn, and R. Chandramouli. *Role-based access control*. Artech house, 2003.

[Gar18] Gartner. *Strategic Technology Trends for 2019*. 2018. URL: `https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019` (visited on 03/15/2022).

[Ger22] Germany. *Federal ministry of economic affairs and climate action*. 2022. URL: `https://www.bmwi.de/Navigation/EN/Home/home.html` (visited on 03/13/2022).

[Gmb22a] K. GmbH. *Kaptura*. 2022. URL: `https://kaptura.de/` (visited on 03/29/2022).

[Gmb22b] team neusta GmbH. *team neusta*. 2022. URL: `https://www.team-neusta.de` (visited on 03/13/2022).

[Gol10] D. Gollmann. “Computer security”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.5 (2010), pp. 544–554.

[GV17] M. Grieves and J. Vickers. “Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems”. In: *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Ed. by F.-J. Kahlen, S. Flumerfelt, and A. Alves. Cham: Springer International Publishing, 2017, pp. 85–113. ISBN: 978-3-319-38756-7. DOI: `10.1007/978-3-319-38756-7_4`. URL: `https://doi.org/10.1007/978-3-319-38756-7_4`.

[HB19] A. Haidu and M. Beetz. “Automated models of human everyday activity based on game and virtual reality technology”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 2606–2612.

[Hei17] G. Heinemann. *Die Neuerfindung des stationären Einzelhandels: Kundenzentralität und ultimative Usability für Stadt und Handel der Zukunft*. Springer-Verlag, 2017.

[Hu+13] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al. “Guide to attribute based access control (abac) definition and considerations (draft)”. In: *NIST special publication* 800.162 (2013), pp. 1–54.

[Hu+15] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas. “Attribute-based access control”. In: *Computer* 48.2 (2015), pp. 85–88.

[Jon+20] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks. “Characterising the Digital Twin: A systematic literature review”. In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 36–52. ISSN: 1755-5817. DOI: `https://doi.org/10.1016/j.cirpj.2020.02.002`. URL: `https://www.sciencedirect.com/science/article/pii/S1755581720300110`.

[Jos+05] J. Joshi, E. Bertino, U. Latif, and A. Ghafoor. “A generalized temporal role-based access control model”. In: *IEEE Transactions on Knowledge and Data Engineering* 17.1 (2005), pp. 4–23. DOI: `10.1109/TKDE.2005.1`.

[K4R20] K4R. *Knowledge4Retail (K4R)*. 2020. URL: `https://knowledge4retail.org/qa_en/` (visited on 03/13/2022).

[KT+19] C. Kaplan, S. Tewes, et al. "Redesigning Business Model Strategy: The Digital Future of Retailing in Europe". In: *Journal Of International Business Research And Marketing* 4.3 (2019), pp. 7–13.

[KSB+19] A. Karar, S. Said, T. Beyrouthy, et al. "Pepper humanoid robot as a service robot: a customer approach". In: *2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART)*. IEEE. 2019, pp. 1–4.

[Kaz+20] G. Kazhoyan, S. Stelter, F. K. Kenfack, S. Koralewski, and M. Beetz. "The Robot Household Marathon Experiment". In: *CoRR* abs/2011.09792 (2020). arXiv: `2011.09792`. URL: `https://arxiv.org/abs/2011.09792`.

[KMB21] M. Kümpel, C. A. Mueller, and M. Beetz. "Semantic Digital Twins for Retail Logistics". In: *Dynamics in Logistics*. Springer, Cham, 2021, pp. 129–153.

[Lam69] B. W. Lampson. "Dynamic protection structures". In: *Proceedings of the November 18-20, 1969, fall joint computer conference*. 1969, pp. 27–38.

[Leh07] K. Lehmann. "Modelle und Techniken für eine effiziente und lückenlose Zugriffskontrolle in Java-basierten betrieblichen Anwendungen". PhD thesis. Technische Universität München, 2007.

[MB19] P. Mania and M. Beetz. "A framework for self-training perceptual agents in simulated photorealistic environments". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 4396–4402.

[Mol08] K. Molitorisz. "Rollenmodelle für die Zugriffskontrolle in Unternehmen". In: *Magisterarb. url: http://www. ipd. kit. edu/Tichy/uploads/publikationen/223/Molitorisz-RollenmodellefrdieZugriffskontrolleinUnternehmen. pdf (siehe S. 9, 24, 34)* (2008).

[PB18] A. Polacco and K. Backes. "The amazon go concept: Implications, applications, and sustainability". In: *Journal of Business and Management* 24.1 (2018), pp. 79–92.

[RU10] R. A. Russell and T. L. Urban. "The location and allocation of products and product families on retail shelves". In: *Annals of Operations Research* 179.1 (2010), pp. 131–147.

[SS75] J. H. Saltzer and M. D. Schroeder. "The protection of information in computer systems". In: *Proceedings of the IEEE* 63.9 (1975), pp. 1278–1308.

[SFK+00] R. Sandhu, D. Ferraiolo, R. Kuhn, et al. "The NIST model for role-based access control: towards a unified standard". In: *ACM workshop on Role-based access control*. Vol. 10. 344287.344301. 2000.

[SK07] Y. Sattarova Feruza and T. H. Kim. "IT security review: Privacy, protection, access control, assurance and system security". In: *International journal of multimedia and ubiquitous engineering* 2.2 (2007), pp. 17–32.

[SO17] D. Servos and S. L. Osborn. "Current Research and Open Problems in Attribute-Based Access Control". In: *ACM Comput. Surv.* 49.4 (Jan. 2017). ISSN: 0360-0300. DOI: 10.1145/3007204. URL: https://doi-org.eaccess.ub.tum.de/10.1145/3007204.

[SS10] S. Staab and R. Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.

[Tao+19] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Y. C. Nee. "Digital twin-driven product design framework". In: *International Journal of Production Research* 57.12 (2019), pp. 3935–3953. DOI: 10.1080/00207543.2018.1443229. eprint: https://doi.org/10.1080/00207543.2018.1443229. URL: https://doi.org/10.1080/00207543.2018.1443229.

[TB09] M. Tenorth and M. Beetz. "KnowRob—knowledge processing for autonomous personal robots". In: *2009 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2009, pp. 4261–4266.

[Tha+21] S. Thalhammer, M. Leitner, T. Patten, and M. Vincze. "PyraPose: Feature Pyramids for Fast and Accurate Object Pose Estimation under Domain Shift". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13909–13915.

[WSS10] Y. Wei, C. Shi, and W. Shao. "An attribute and role based access control model for service-oriented environment". In: *2010 Chinese Control and Decision Conference*. IEEE. 2010, pp. 4451–4455.

[Zho+14] W. Zhou, L. Li, M. Luo, and W. Chou. "REST API design patterns for SDN northbound API". In: *2014 28th international conference on advanced information networking and applications workshops*. IEEE. 2014, pp. 358–365.