TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Ragnar Toomla
176456IAAM

Architecture Design for the Customer Access Management of Digital Channels in a Financial Services Company Based on SEB Pank AS Example

Master's thesis

Supervisor: Silvia Lips
PhD

Co-Supervisor: Henrik Leinola
Enterprise area architect
SEB Pank

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Ragnar Toomla
176456IAAM

Finantsasutuse digitaalsete kanalite pääsusüsteemi arhitektuuri kavandamine SEB Pank AS näitel

Magistritöö

Juhendaja: Silvia Lips
Ph.D

Kaasjuhendaja: Henrik Leinola
Valdkonna IT arhitekt
SEB Pank

Tallinn 2024

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Ragnar Toomla

14.05.24

Abstract

This thesis presents an architectural design for customer access management in a financial services company, using SEB Pank AS as an example. A review of the existing research identifies current challenges within existing access control models, such as Role-Based and Attribute-Based Access Control models. The thesis also evaluates regulations that influence access management practices. By collecting stakeholder requirements and analysing the current solution, challenges in the current solution are identified. Specifically, authorisation decisions are internal to each functional module, there is code duplication, and the access rights model is not self-explanatory and is too complex for smaller customers. The thesis describes the architecture for a new customer access management system in which access control decisions are externalised, policy management is clearly defined, and the user rights model is more flexible and easier to use. This work aims to contribute to the field by providing a user-centric access management architecture that aligns with current and future digital banking needs.

This thesis is written in English language and is 123 pages long, including 11 chapters, 22 figures and 9 tables.

Annotatsioon

Lõputöö käsitleb finantsasutuse pääsusüsteemi arhitektuuri kavandamist, SEB Pank AS näitel. Töös analüüsib autor pääsusüsteemi teoreetilist tausta, et selgitada välja kaasaegsed parimad praktikad. Käsitletakse rollipõhist ja atribuutidel põhinevaid pääsusüsteemi mudeleid, nende eeliseid ja puudusi. Samuti antakse ülevaade uuematest suundumustes selles valdkonnas.

Ülevaade kohalduvatest seadustest ja määrustest annab sisendi juriidilistele ja vastavuskontrolli nõuetele, mida peab arvestama pääsusüsteemi planeerimisel. Autor viib läbi sidusgruppide intervjuud ja analüüsib olemasolevat juurdepääsuõiguste mudelit, et selgitada välja tänase lahenduse kitsaskohad ja nõuded uuele süsteemile. Olemasoleva lahenduse põhilised kitsaskohad on seotud puudustega arhitektuuri ülesehituses ja õiguste parameetrites. Autoriseerimisotsuste loogika on osa igast funktsionaalsest moodulist, mis viib koodi dubleerimisele ja raskustele autoriseerimisotsustest ülevaate saamisel, lisaks on muudatuste tegemine sellise lähenemise puhul väga keerukas ja aeganõudev. Pääsuõiguste atribuudid ei ole iseenesestmõistetavad ja on väiksemate klientide jaoks liiga keerulise ülesehitusega.

Lõputöö kirjeldab uue pääsusüsteemi arhitektuuri, kus autoriseerimise otsused on koondatud kesksesse moodulisse, mis pakub teenust kõikidele äriloogikaga tegelevatele süsteemi komponentidele. Lisaks pakutakse välja võimalikud muudatused pääsuõiguste atribuutides, et muuta süsteemi lihtsamaks ja parimini mõistetavaks.

Selle töö eesmärk on anda oma panus valdkonda, pakkudes kasutajakeskset pääsusüsteemi arhitektuuri, mis on kooskõlas praeguste ja tulevaste digitaalse panganduse vajadustega.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 123 leheküljel, 11 peatükki, 22 joonist, 9 tabelit.

List of abbreviations and terms

ABAC	Attribute-Based Access Control
ACL	Access Control Lists
ACM	Access Control Mechanisms
AI	Artificial Intelligence
AISP	Account Information Service Provider
AML	Anti-Money Laundering
API	Application Programming Interface
BAT	Baltic Architecture Team
BFF	Backend for Frontend
CEO	Chief Executive Officer
CIAM	Customer Identity and Access Management
DAC	Discretionary Access Control
DORA	Digital Operational Resilience Act
DP	Digital Policies
DSD	Dynamic Separation of Duty
DSR	Design Science Research Methodology
EC	European Commission
EE	Estonia (Country Code)
eIDAS	The Regulation on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market
eIDAS 2.0	The European Digital Identity Framework
eIDs	National Electronic Identification Schemes
ERD	Entity Relationship Diagram
ERP	Enterprise Resource Planning
EU	European Union
EUIDW	European Digital Identity Wallet
EUR	Euro (Currency)
FIDA	Financial Data Access Regulation

FK	Foreign Key
GDPR	General Data Protection Regulation
IAM	Identity and Access Management
IB	Internet Bank
IBAN	International Bank Account Number
IBGW	Internet Bank Gateway
ICT	Information and Communications Technology
ID	Identifier
IEC	International Electrotechnical Commission
IKS	Isikuandmete kaitse seadus
IP	Internet Protocol
IS	Information Systems
ISO	International Organization for Standardization
IT	Information Technology
ITDS	Isikut tõendavate dokumentide seadus
JSON	Javascript Object Notation
JWT	JSON Web Token
KYC	Know Your Customer
LT	Lithuania (Country Code)
LV	Latvia (Country Code)
MAC	Mandatory Access Control
MFE	Micro Frontends
MP	Metapolicies
NFR	Non-Functional Requirement
NIS-2	Network and Information Security Directive
NIST	The National Institute of Standards and Technology
OBS	Objects
OPA	Open Policy Agent
OPS	Operations
OWASP	The Open Worldwide Application Security Project
PA	Permission Assignment
PAP	Policy Administration Point
PBAC	Policy-Based Access Control
PDP	Policy Decision Point

PEP	Policy Enforcement Point
PIN	Personal Identification Number
PIP	Policy Information Point
PISP	Payment Initiation Service Provider
PK	Primary Key
PKI	Public Key Infrastructure
PRMS	Permissions
QSCD	Qualified Signature Creation Device
RBAC	Role-Based Access Control
RTS	Regulatory Technical Standards
SCA	Strong Customer Authentication
SFA-Matrix	Suitability, Feasibility and Acceptability Matrix
SIM	Subscriber Identity Module
SLR	Systematic Literature Review
SME	Small and Medium-Sized Enterprises
SOD	Separation of Duties
SQL	Structured Query Language
SSD	Static Separation of Duty
SSI	Self-Sovereign Identity
TLS	Transport Layer Security
UA	User Assignment
UML	Unified Modeling Language

Table of contents

List of Figures

List of tables

1 Introduction

Customer access management is a cornerstone of modern digital applications, crucial for protecting sensitive information, ensuring compliance with regulations, enhancing user experience, and maintaining operational efficiency.

The thesis aims to design a new, modern customer access management system on the example of SEB Pank. SEB is Estonia's second-largest bank and part of a Skandinaviska Enskilda Banken AB banking group. The author has been working there for several years and has in-depth knowledge of the digital service channels.

Customer access management is a subset of a wider discipline called Customer Identity and Access Management, which deals with customer access rights and how these rights are checked and managed. In the thesis, the author analyses the theoretical background of access rights management to identify state-of-the-art in the area. The applicable laws and regulations are reviewed, and an analysis of the current access rights model is performed before key problem areas are identified and a new proposal is formed. This work focuses on designing software architecture and data models to suit current stakeholder needs and be future proof for future requirements.

The topic is actual for the following reasons:

1. SEB is transitioning to a new micro-frontend architecture model in the development of the Internet Bank. This architecture will allow many development teams to develop individual self-service components. It will place additional demands and challenges on how access rights management is built up.
2. In the coming years, the European Union will introduce several regulations that will affect this area, either by setting more strict governance requirements or introducing a new approach to managing identities and credentials.
3. The access rights and access rights management in SEB digital channels have not been fundamentally updated since the introduction of the system 18 years ago.
4. Broken access control is listed as the most serious web application security risk in the latest OWASP Top 10 report[1]. This means that the topic is relevant to a wider range of applications.

So far, the existing access rights system used by the bank has been changed according to specific needs, but a comprehensive and widespread analysis of todays and future business requirements has not been done.

The thesis addresses the following questions:

1. What is the state of the art in Identity and Access management? The goal is to understand what modern approach to access management should be considered a potential solution for SEB customer identity and access management.
2. What are SEB's current access control solution's capabilities and challenges? This information helps establish a baseline for possible improvement suggestions.
3. "What should the architecture design of a new system for customer access control solution look like?". Outlining the overall approach, key components, and data model helps to communicate the intent and changes required for the existing model. This is used later in the implementation phase to build the solution.

The thesis contains 11 chapters. Chapter 1 introduces the thesis, followed by Chapter 2, which describes the research methodology. Chapter 3 overviews the related works and theoretical background, and Chapter 4 discusses applicable European and Estonian legislation. Chapter 5 describes stakeholder requirements, Chapter 6 describes the existing solution, and Chapter 7, on page 56, gives the main findings and authors' recommendations. Chapter 8 contains the design artefacts of the proposed solution, and validation in two iterations is described in Chapter 9. Limitations of the current thesis and future work recommendations are discussed in Chapters 10 and 11, respectively.

2 Research Methodology

The chapter describes the research methodology chosen, the methods used for conducting the research, and the architecture.

2.1 Design Science Research

The chosen approach for this study was Design Science Research Methodology (DSR). The methodology has been comprehensively described by Hevner [2][3][4]. Hevner outlined its key principles, phases, and evaluation criteria. DSR's problem-solving focus, action-oriented approach, iterative process, user-centricity, and practical knowledge creation make it a highly suitable methodology for information systems design. It provides a framework for creating innovative and effective Information Systems (IS) solutions that address real-world challenges and improve organisational outcomes [5]. It aims to deliver innovative solutions for real-world problems. DSR produces Information Systems artefacts and design knowledge describing means-end relationships between problem and solution spaces [6][7]. Figure 1 describes the DSR model by Hevner et al [3] with an overlay of three inherent research cycles [4].

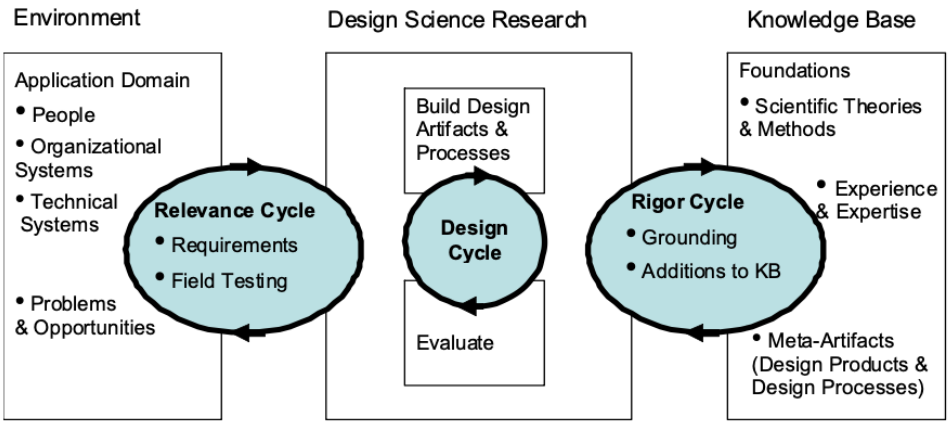


Figure 1. Design Science Research Model [4].

The Relevance Cycle bridges the research context with design science. The Rigor Cycle connects design science with experience and expertise. The central Design Cycle iterates between building and evaluating design artefacts.[4]

2.2 Research Process

Research design was developed based on Vaishnavi's Design Science Research Process Model [8]. Figure 2 illustrates the research process undertaken for the thesis, helping readers better understand the methodology employed. The structure of the figure is based on [9].

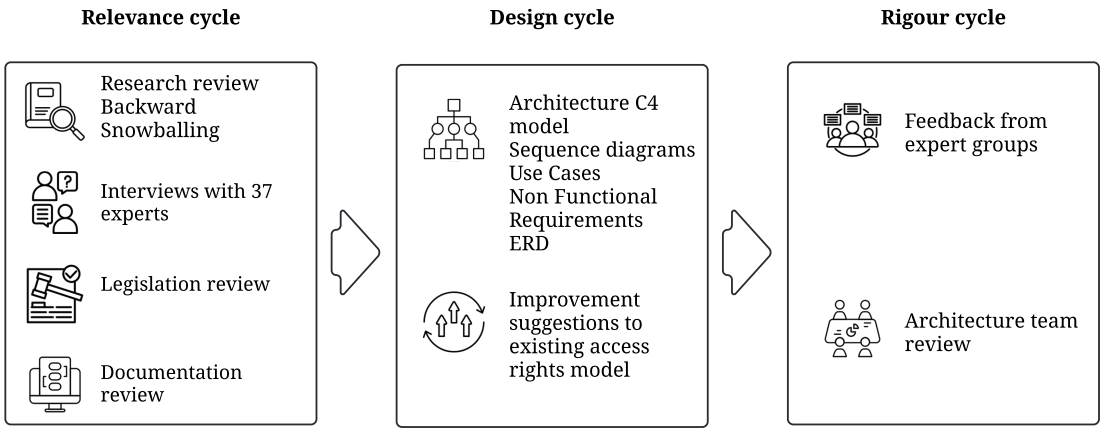


Figure 2. DSR process based on [9].

The research was carried out in three cycles: relevance, design, and rigour.

2.2.1 Relevance Cycle

In the relevance cycle, the author reviewed existing scientific research to establish a theoretical background and identify best practices. Google Scholar was used to search relevant studies. A review was carried out from December 2023 to February 2024.

The following keywords were used for the search:

1. "Identity and access management" articles published since 2020 (6 770 matches). Abstracts for the forty most relevant matches were reviewed, and seven papers were selected for detailed review. Backward Snowballing was used to identify other relevant studies from earlier periods. Backward Snowballing looks backwards in the literature. It takes relevant articles and looks at their reference lists for other articles that might be relevant [10].

2. "Customer identity and access management" (92 matches). Abstracts for all matches were reviewed, and 17 papers were selected for detailed review.

It is important to clarify that the objective was not to conduct a systematic literature review (SLR)[11] or examine every research paper related to CIAM. Rather, the author's intention was to understand the domain and relevant research papers better.

Semi-structured focus group interviews were conducted with representatives from three distinct stakeholder groups: IAM subject matter experts, existing solutions experts, and employees in customer service. Interviews are more valid because the responses are more reliable than questionnaires [12]. Interviews were conducted with 37 persons via Microsoft Teams, transcribed and later coded in software for Qualitative Content Analysis[1]. The author reviewed current and upcoming applicable legislation to identify potential future system needs. The author used existing system documentation to understand the setup of the solution.

2.2.2 Design Cycle

In the design cycle, an architecture design artefact was developed. The author chose to use the C4 model created by Simon Brown [13] to visualise higher-level architecture due to the need to communicate the architecture to an audience who is not familiar with Unified Modeling Language (UML) notation [14]. The C4 model is designed to help developers understand and develop software architectures through a multi-level approach, dividing the architecture into multiple layers of abstraction: context, containers, components, and code. The C4 model complements traditional UML diagrams to enhance the software design process. It helps professionals start designing systems without diving into complex details, easing the understanding of software architecture and requirements elicitation.[15] To complement the C4 model, a sequence diagram and entity relationship diagram (ERD) of the suggestion were developed. Also, the main use cases and non-functional requirements were specified.

[1] QCAmap; https://www.qcamap.org/

2.2.3 Rigour Cycle

In the rigour cycle, emphasis should be placed on properly evaluating artefacts, a key activity in any DSR project [16]. Evaluation and improvement were done in two steps:

For the first step, the author used an expert evaluation method: a focus group interview. Based on the feedback received from the expert group, an updated version of the architecture design was created.

For the second step, a body of enterprise area architects at SEB – the Baltic Architecture Team (BAT) conducted an evaluation of the architecture design proposal in the form of a workshop. Based on the feedback received from BAT, an updated version of the architecture design was created.

3 Theoretical Background and Related Works

This chapter provides an overview of the research conducted on Customer Identity and Access Management (CIAM). Examining the existing literature is a crucial step in any research project, as it lays the groundwork for the thesis. The purpose of this analysis is to assess, analyse, and consolidate the significant discoveries and contributions of previous research about CIAM.

3.1 Identity Access Management (IAM)

The acronym "IAM" represents "Identity and Access Management", which comprises policies and technologies that guarantee that technology resources are only accessible to authorised users. IAM systems are crucial in corporate environments, where individuals require access to sensitive systems, databases, or applications. They manage user identities and regulate access to company resources, preventing unauthorised access and ensuring that users can only access the resources necessary for their job duties. IAM is composed of two components - Identity Management and Access Management. Identity Management deals with identity provisioning, while Access Management deals with authentication, authorisation, and policy management. IAM decides who has access rights to what resources. [17] This research focuses on the access management component of IAM. The following are the main functions [18]:

4. **Identity Provisioning**: The provisioning of identities within an organisation addresses the provisioning and revocation of user accounts.
5. **Authentication**: Authentication ensures that the individual is who he claims to be, and is identified through various mechanisms, such as password, certification, biometrics, etc.
6. **Authorisation**: The authorisation module provides an interface to enforce authorisation rules as clients attempt system operations. These rules apply to accessing data within the system and to operations that can be applied t system data.
7. **Policy Management**: The policy management module enforces the policies that associate users with resources. It resolves the appropriate policies for a user and determines the resources for which that user is authorised.

The requirements for an IAM system have been extensively researched by Glöckler using a systematic literature review and refined by twelve domain experts. The study proposes four clusters of requirements: 'Security & Compliance', 'Operability', 'Technology', and 'User' [9]. structuring of IAM requirements is illustrated in Figure 3.

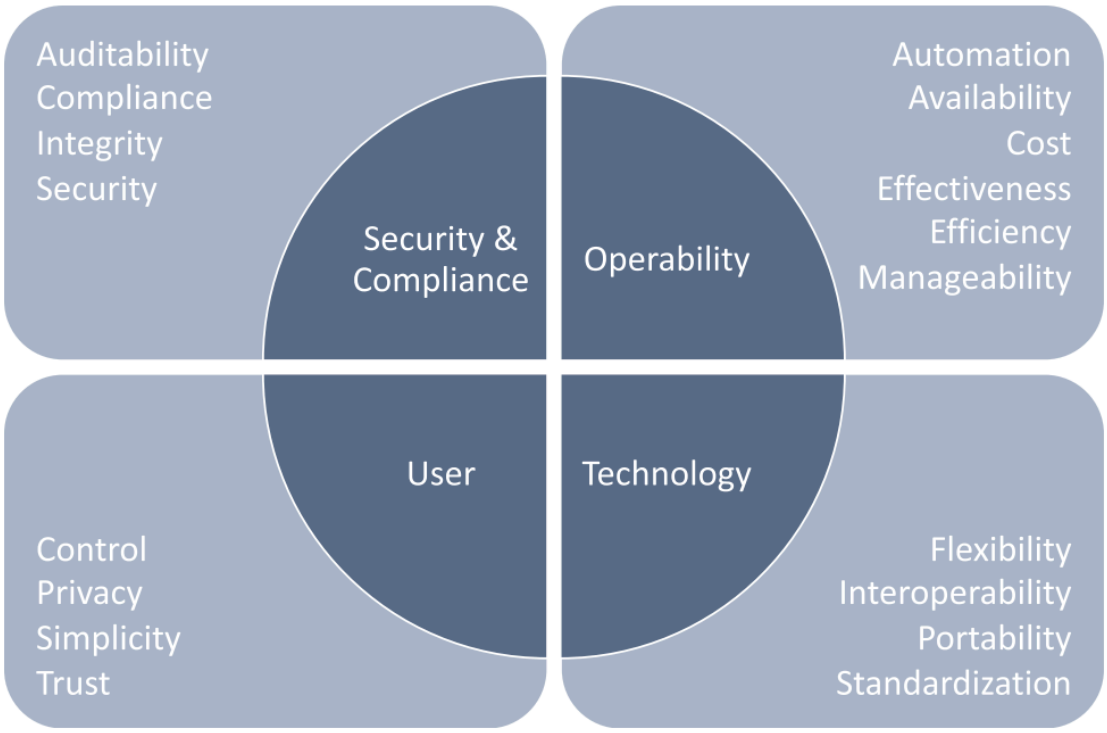


Figure 3. Requirements for an enterprise IAM system [9].

IAM, as a field, has evolved over time, integrating with various technologies, and adapting to the changing landscape of digital identity and security.

3.2 Customer Identity Access Management (CIAM)

The lifecycle phases for different identities, including humans like Workforce or Customers and non-human types like Systems or Devices, vary; Enterprise IAM comprises a set of established processes that provide governance capabilities and ensure that only authorised accounts have access to the required resources and applications. Customer IAM has an entirely different set of requirements representing value to a business due to its defining interactions with customers [19]. Customer IAM has evolved

more recently to support the processes that govern consumers' User Experience as they interact with digital business. One of the significant differences between IAM and CIAM is that IAM is an enterprise solution and can also be used as an internal IAM solution, whereas CIAM is an external IAM solution. CIAM is a critical and influential enabler for developing digital business strategies because it covers positive customer interactions, scalability and customisation across all channels required for digital transformation [20].

Most customer experiences represent a customer's interactions (Authentication, Registration, Profile Update) when engaging with digital services. Figure 4 describes the phases of the CIAM Lifecycle [19].

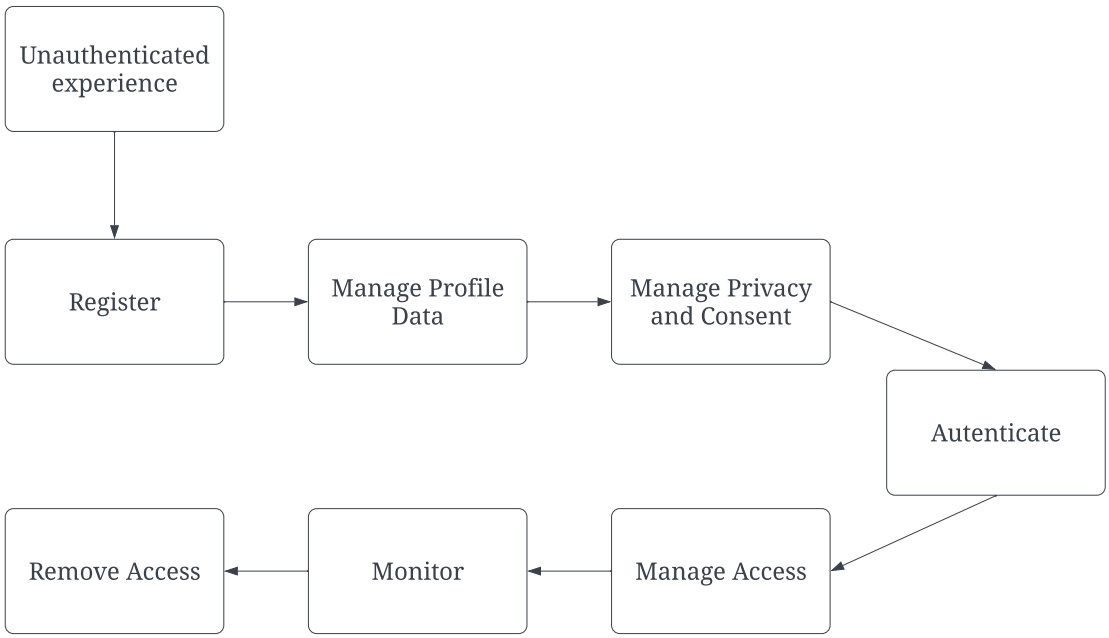


Figure 4. The Customer Identity Lifecycle [19].

Modern CIAM solutions must support the rapid change in business models and technologies and adapt to new trends and capabilities, including decentralised identities, fraud reduction, and passwordless authentication. Emerging trends dictate these six capability areas that stand out [21]:

1. **Central Identity Service**: In the Digital Age, CIAM must serve as the backend for digital services. It must be able to perform many such digital services without complex adaptation to enable rapid service delivery and decrease time to value.

2. **Focused capabilities & integration**: While CIAM in the past tended to integrate a wide variety of capabilities, including Marketing Automation and Customer Data Platform features, the current trend is towards specialisation and analytics.
3. **Fraud Intelligence**: FRIPs (Fraud Reduction Intelligence Platforms) have gained substantial momentum due to a steady and steep increase in cyber-attacks. Thus, such technologies must be part of the overall CIAM solution, either as built-in capabilities or via integration. This helps to detect Account Take Over attacks and attacks while onboarding and executing transactions. It is also a supportive technology for Adaptive Authentication and Progressive Profiling.
4. **Support for Decentralised Identities**: Another major trend in the market has been toward Decentralised Identities. These aren't owned and managed by enterprises but by individuals. Modern CIAM must allow for the integration of decentralised identities and the mapping of these to internal customer records.
5. **Support for Passwordless Authentication**: Because passwords have inherent weaknesses that affect security and convenience, passwordless authentication has seen a steep increase in adoption for workforce and external identities. CIAM solutions ideally support phishing-resistant, passwordless authentication capabilities that build on biometric authentication and device trust.
6. **Built for Zero Trust**: Zero Trust has emerged as the leading security principle. The concepts of "don't trust, always verify," including continual authentication, are essential to modern CIAM.

With the trend of delivering solutions as a service, the requirements for the architecture and implementation of CIAM solutions have also changed fundamentally. CIAM must start by thoroughly assessing technical requirements, business requirements, and the evolution of business models.

3.3 Access Control Models

Without access control, identity management would be unnecessary. Access control is primarily responsible for ensuring that users can only access the resources they are authorised to access. [22]

3.3.1 Discretionary Access Control (DAC)

Discretionary Access Control (DAC) policy is determined by the end user who has permission [23]. For instance, access control is usually employed to limit user's access to a resource. In this case, the resource owner has the authority to regulate other users' access to the resource. DAC policy is very flexible and widely used, however it is known to be inherently weak for several reasons [24]:

- There is no guarantee on the flow of information in a system since it can be easily copied from one object to another.
- After the user has gotten the information, its usage has no restrictions.
- The owner of an object determines access privileges instead of a system-wide policy reflecting organisational security requirements.

3.3.2 Access Control Lists (ACL)

Access Control Lists (ACLs) is the most used mechanism for implementing Discretionary Access Control (DAC) policies. Access to protected resources is controlled based on their classification level. ACLs are used to control access to objects. The owner defines which users can read, write, update, or delete the object. This system is easy to manage for individual objects, but it can be limiting as the number of users and objects grows. [22]

3.3.3 Mandatory Access Control (MAC)

In this model, a system administrator centrally manages security policies and controls which users can access what resources. Access to resources is based on security levels, which are determined by the sensitivity of the information. Both resources and users are tagged to one of the security levels. When a user wants to access a particular resource, the system verifies whether the security level of the resource and the user match. If they match, the user can access the resource, but if they don't match, the system will deny access [23].

3.3.4 Role-Based Access Control (RBAC)

Role-based access control (RBAC) policies regulate user access to information based on their activities. Role-based policies require the identification of roles in the system. A role is a job function that determines a person's actions, responsibilities, and resource

permissions. Instead of specifying all the accesses each user can execute, access authorisations on objects are specified for roles. Users are authorised to adopt roles. [24]

Sandhu et al. [25] have identified four RBAC levels. These four levels of RBAC are described in Table 1 and are viewed as a standard RBAC reference model. Each RBAC level includes the conditions of the previous level.

Table 1. Standard RBAC reference model [25].

Level	**Name**	**RBAC Functional Capabilities**
1	Flat RBAC0	users get permissions through roles. many to many user role assignments many to many permission role assignments user role assignment review users can use multiple roles simultaneously
2	Hierarchical RBAC1	Flat RBAC + role hierarchy partial order, senior roles acquire the permissions of their juniors
3	Constrained RBAC2	Hierarchical RBAC + separation of duties (SOD), this can be either Static Separation of Duty (SSD) or Dynamic Separation of Duty (DSD)
4	Symmetric RBAC3	Constrained RBAC + combines RBAC1 and RBAC2, which means role hierarchies and constraints. permission role review

RBAC includes sets of five basic data elements called users (USERS), roles (ROLES), objects (OBS), operations (OPS), and permissions (PRMS). Individual user relationships and permissions are defined in the core RBAC model. Additionally, the model includes sessions (SESSIONS) which map users to activated roles [26]

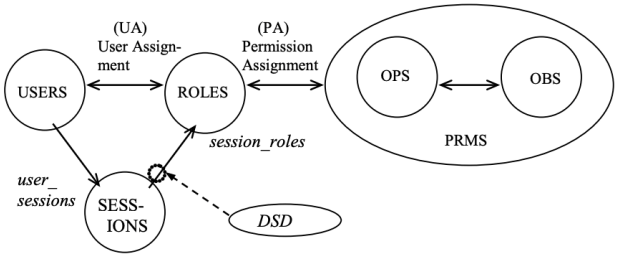


Figure 5. RBAC model with Dynamic Separation of Duty Relations [26].

Each user session can activate multiple roles assigned to them. Each user is associated with one or more sessions, and each session is associated with a single user. The function session_roles gives us the roles activated by the session and the function user_sessions gives us the set of sessions associated with a user. The permissions available to the user are those assigned to the roles activated across all the user's sessions. [26] Dynamic Separation of Duty (DSD) relations and Static Separation of Duty (SSD) relations have the same objective of limiting the permissions of a user. However, the difference between DSD and SSD is in the way these limitations are imposed. While SSD relations restrict permissions available to a user, DSD relations limit the permissions by limiting the roles that can be activated for a user.[27]

3.3.5 Limitations of RBAC

Connecting organizational roles to application roles for efficient authorisation management is easy to complicate by nesting groups [22].

RBAC assumes all permissions needed to perform a function can be neatly encapsulated. Role engineering has turned out to be a difficult task. For stronger security, it is better for each role to be more granular, thus having multiple roles per user. For easier administration, it is better to have fewer roles to manage. Organisations must comply with privacy and other regulatory mandates and improve enforcement of security policies while lowering overall risk and administrative costs. [28]

The Role-Based Access Control (RBAC) model has a significant shortcoming in that it is static. This means that once a user is granted entitlements, they remain available to the user until they are manually revoked. As a result, users may continue to have access to certain resources even when they switch roles, unless there are proper clean-up actions taken to revoke their access. [22]

3.3.6 Attribute-Based Access Control (ABAC)

RBAC's groups and permissions were inadequate for distributed systems. Enterprises added attributes like location and time. During this period, attribute-based access control (ABAC) was identified as a replacement for or adjunct to RBAC [29]. Initial ABAC approaches were introduced by Wang et al.[30], Yuan et al.[31], Later, Hu et al. gave a more comprehensive view of ABAC [32]. ABAC considers various attributes and characteristics of users when deciding whether to grant access or not, it can be used both actively, in real-time, to control access during a transaction, and passively, by assigning roles and entitlements based on user metadata [22]. In ABAC, permissions to access the objects are not directly given to the subject. It uses attributes of the subjects and objects to provide authorisations. For subjects, we consider static attributes like a subject's name, designation or role in an organisation and dynamic attributes like age, current location, or an acquired subscription for a digital library. For objects, we can consider metadata properties such as the subject of a document [33]. Figure 6 displays these basic core capabilities of ABAC systems, according to Hu [32].

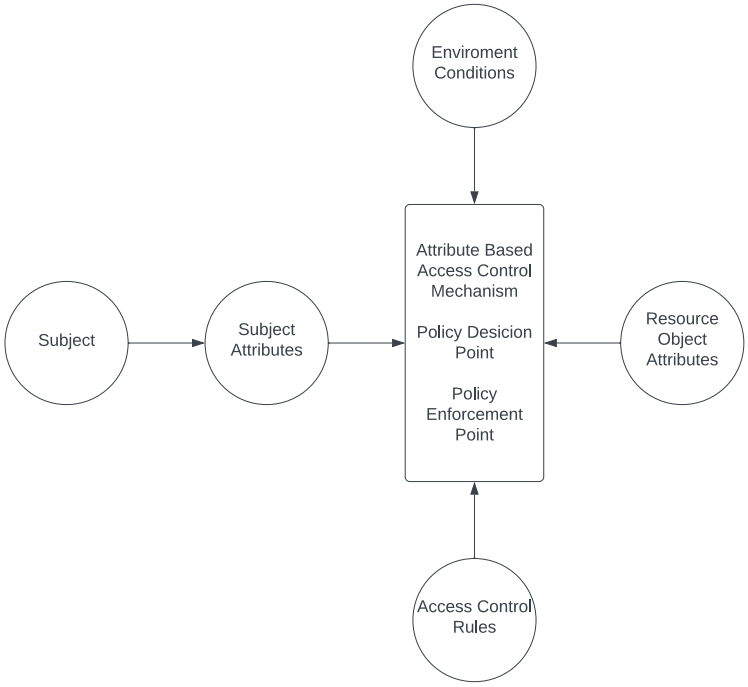


Figure 6. Core ABAC Mechanisms [32].

Subjects require specific attributes, while objects need a policy defining access rules for subjects, operations, and environmental conditions. This policy is derived from

documented or procedural rules that describe the organization's business processes and authorised actions. Access Control Mechanisms (ACM) are designed to restrict access to objects by allowing only specific operations to be performed by authorized subjects. The ACM gathers policy, subject attributes, and object attributes and then decides based on the provided policy and logic. It enforces the decision by granting or denying access to the object. The ACM is responsible for managing the process of making and enforcing decisions related to access control. This includes determining the policy that needs to be retrieved, the order in which attributes need to be retrieved, and where to retrieve them from. Once this information is gathered, the ACM performs the necessary computations to determine the appropriate course of action.[32]

Within the ACM are several components that deal with retrieval and management of the policy, together with components handling the policy and attribute assessment. Figure 7 shows the main functional points [32].

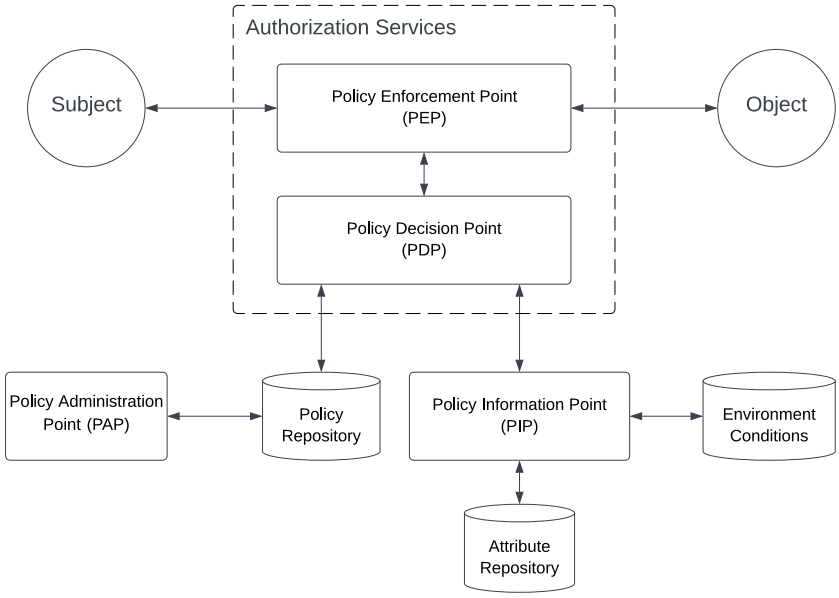


Figure 7. ABAC ACM Functional Points [32].

The policy repository contains information about digital policies (DP), access control rules compiled directly into machine executable codes. Subject/object attributes, operations, and environmental conditions are the fundamental elements of DP, the building blocks of DP rules, which are enforced by an access control mechanism. When there are multiple Data Points (DPs) involved, metapolicies (MP) may be needed. These policies help manage the hierarchical authorities of DPs, resolve conflicts between DPs,

and ensure proper storage and updates of DPs. The Policy Decision Point (PDP), which evaluates DPs and MPs, makes access control decisions.[32]

The Policy Enforcement Point (PEP) is responsible for requesting authorisation decisions and enforcing them. In essence, it is the point of presence for access control and must be able to intercept service requests between information consumers and providers. Although the diagram depicts the PEP as a single point, it may be physically distributed throughout the network. The most important security engineering consideration for implementing a PEP is that the system must be designed so that the PEP cannot be bypassed to invoke a protected resource [31]. Policy Information Point (PIP) is the source for attributes or data required for policy evaluation to help PDP make decisions. [32].

3.3.7 Limitations of ABAC

Identity and Access Management (IAM) challenges in modern companies are becoming increasingly complex and diverse. Attribute-Based Access Control (ABAC) is a more advanced version of Role-Based Access Control (RBAC) and provides greater flexibility in managing access. However, implementing ABAC presents challenges that research has not yet fully addressed. A lack of attribute quality can result in dysfunctional access control decisions and security vulnerabilities [34].

Lack of role hierarchy.

In hierarchical RBAC, the role hierarchy allows for roles to be related in a way that more closely resembles that of actual organisations. This allows for more simplistic administration in terms of role engineering and reviewability of existing role-based policies. However, most "pure" ABAC models lack this inheritance and expressiveness. While a role can be easily modelled as a single attribute of a subject, this simplistic representation cannot emulate the hierarchical nature of RBAC without allowing for complex data types in an attribute's value. [35]

Complexity in managing many attributes.

With ABAC, there's no need to engineer roles if role names aren't used as attributes. Dynamically changing attributes, such as time of day and location, can be accommodated in access control decisions. However, many attributes must be understood and managed, and attributes must be selected by expert personnel. Furthermore, attributes have no

meaning until they're associated with a user, object, or relation, and it's not practical to audit which users have access to given permission and what permissions have been granted to a given user. [29]

Challenging to compute the set of users who have access to a resource.

An important aspect of access control for legal and security reasons is the ability to easily determine the set of users who have access to a given resource or the set of resources a given user may have access to (sometimes referred to as a "before the fact audit"). In RBAC, this is relatively straightforward, normally requiring the system to calculate the union of the set of effective privileges from each role the user is assigned. However, in ABAC, this is considerably more complicated. [32] As ABAC is an identityless access control system and users may not be known before access control requests are made, it is often impossible to compute the set of users with access to a given resource [35].

Complexity in managing multiple attribute sources.

When multiple attribute sources are used in an ABAC system (e.g., using attribute authorities from different organisations in a distributed system), complications can arise in terms of evaluating the trustworthiness of attributes and ensuring that differing attribute sources are using compatible attributes (e.g., using the same namespace and data type for common attributes). [35]

It is difficult to design an access control model that is easy to understand for end users.

Creating and maintaining access control policies in open distributed systems can be complex. ABAC's "human aspect" is often overlooked, how usable the system is for users, access control admins, and policy engineers. Although ABAC systems have desirable theoretical properties and have demonstrated utility when used by security-conscious researchers with a background in mathematical logic, the problem of providing an adequate open system access control solution for the average user is far from solved. [36]

3.4 Combining RBAC and ABAC

RBAC has been criticized for its complexity in setting up a role structure and lack of flexibility in rapidly changing domains. There is a growing interest in Attribute-Based

Access Control (ABAC), which could potentially replace or simplify Role-Based Access Control (RBAC) by using attributes and rules. Merging RBAC and ABAC can create better access control model for distributed and rapidly changing applications. Kuhn et al. from The National Institute of Standards and Technology (NIST) have proposed three approaches for adding attributes to role-based access control [37]:

- **Dynamic roles**. A front-end module uses attributes such as time of day to determine the subject's role, keeping a conventional role structure but changing role sets dynamically. There are different ways to implement dynamic roles, some of which allow the front-end attribute engine to fully determine the user's role, while others only permit the front-end to choose from a predetermined set of authorised roles.
- **Attribute-centric**. In the context of Role-Based Access Control (RBAC), a role name is one of several attributes. However, unlike the conventional RBAC approach, where a role is a collection of permissions, in this approach, a role is simply the name of an attribute known as a "role". The main disadvantage of this approach is that it can lead to a significant loss of RBAC's effectiveness over time.
- **Role-centric**. Attributes are used to limit the permissions granted by RBAC. When attributes are incorporated into constraint rules, they can only decrease the permissions available to the user, not increase them. Although this approach reduces some of the flexibility provided by ABAC, it still allows the RBAC system to determine the maximum set of permissions that a user can obtain while being constrained by their assigned role.

Balancing administrative ease and dynamic control is essential when managing access to resources. That's why a hybrid approach that combines the best aspects of RBAC and ABAC should be considered.

3.5 The Future Direction of Access Control

In modern implementations, a policy engine evaluates access policies centrally. The business process owner or data owner defines the policies for which they are accountable. Multiple 'business owners' are sometimes assigned, each responsible for their part of the corporate security policy, resulting in continuously changing access control policies.

Modern applications rely on authorisation systems that decide user access based on access policies.[22]

3.5.1 Dynamic Authorisation

Dynamic authorisation represents the definition of access rights in terms of real-time evaluation of rules and policies [38]. An example of such an approach is when a user starts a session accessing services requiring no identification or only a light low risk identification with a cookie might be enough. Later, higher trust level may be required. For instance, two-factor identification might be needed when performing a transaction [22].

Dynamic authorisation takes ABAC a step further by enabling fine-grained access control. Adding fine-grained attributes allows you to evaluate the context of each request. Unlike traditional authorisation, which typically has static rules for access, fine-grained authorisation lets you control access beyond the application and resource levels to require that certain conditions are met [39].

Another key principle of dynamic authorisation is that policies are evaluated in real time, allowing access decisions based on the most up-to-date information. This reduces the risk of security breaches due to outdated or incorrect access permissions [40].

3.5.2 Policy-Based Access Control (PBAC)

Policy-based Access Control (PBAC) is a reliable way of managing permissions through structured rules. Unlike the RBAC model, which bundles permissions intentionally, PBAC uses an ABAC concept to automate fine-grained, decoupled permissions. PBAC uses permissions expressed as policies to determine who can access what within an application. By leveraging ABAC's approach of calculating permissions based on user information, PBAC provides increased precision by supporting appropriate access conditions [41].

Administrators and developers have two options when defining application access: static admin-time authorisation and dynamic run-time authorisation. Admin-time authorisation is based on users and groups defined by roles and responsibilities. On the other hand, run-time authorisation applies access controls based on contextual elements like time or location when a user tries to access a specific application resource. These two policy types

make policy-based access control (PBAC) a powerful authorisation engine. A central policy store and engine constantly evaluate these policies in real time to determine resource access. PBAC is a more dynamic access control model, allowing developers and administrators to create and modify policies according to their needs, such as defining custom roles within an application or enabling secure, delegated authorisation [42].

In developing requirements and deploying a solution for PBAC, the following should be considered [43]:

1. **Access control decisions should be externalised**. Modern applications should rely on external access decision control, rather than maintaining static entitlements at the data level or in their databases to determine user access rights, which can be difficult to integrate. Policies can auto-generate and manage entitlements.

2. **Policy management should be centralised**. It is important to ensure that policy definition, management, governance, policy creation, and management are aligned with common corporate policies, even if administration points are distributed for different use cases. Business units should take responsibility for these tasks to ensure consistency and adherence to corporate policies.

3. **Support for on-premises, cloud, and cloud-native assets is essential.** For efficient decision point deployments, keeping them near the connected applications and databases is advisable.

4. **Decision data should be as real-time as possible.** Solutions maintaining their information point data must have a mechanism for ensuring data quality and governance.

5. **Data governance is generally required for information used for decision-making.** PBAC requires well-governed policies and current data to make authorisation decisions at run-time.

6. **Support for corporate governance is required**. It is recommended that monitoring and event management integration be deployed, along with tools like policy analytics.

7. **Compliance with regulatory requirements** is essential in today's business environment.

The main element of this architecture is the Policy Decision Point. It assesses access policies and provides a response to the access request. Then, the Policy Enforcement Point enforces the response through code embedded in the application or, more commonly, via an API gateway [41].

3.5.3 Self-Sovereign Identity and Verifiable Credentials

Also known as Decentralized Identity, User-Centric Identity, Self-Managed Identity and User-Controlled Identity, the Self-Sovereign Identity (SSI) is an identity management model in which the identity holder has broader control over their data and decides how and under what conditions their data should be shared with others. The SSI model is intended to preserve the right to selectively disclose the identity holder's data in different contexts. The identity holder is a role an entity might perform by possessing one or more claims.[44]

Typically, users are provided with digital wallet apps on their mobile phones that enable them to self-manage digital representations of identity documents such as passports, qualifications, or access authorisations [9]. Verifiable credentials are digital versions of physical credentials that use digital signatures to make them more trustworthy and tamper evident. Verifiers can use these to establish trust from a distance as they can be transmitted quickly. The verifiable credentials data model is published as a W3C recommendation [45]. Click or tap here to enter text. The OpenID Foundation is developing specifications and software libraries for OpenID for Verifiable Credentials [46]. Verifiable Credentials are bringing a paradigm shift in the trust model, enabling users who possess the credentials to present credentials to the verifiers, who verify the credential without directly contacting the issuer of that credential and to control their relationship with the verifiers independent from third-party identity providers' decisions or lifespan [47].

4 Applicable Legislation

This section presents an overview of the relevant European and Estonian legislation. Given the highly regulated nature of the financial services industry, it is crucial to incorporate legal considerations into the system analysis and design document. Such an approach serves as a fundamental aspect of responsible system development, as it ensures that the system not only satisfies stakeholder requirements but also adheres to applicable legal and ethical guidelines. This is crucial for ensuring the long-term viability and acceptability of the system.

4.1 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is a regulatory framework that establishes standards for collecting and processing the personal data of European Union citizens [48]. To comply with GDPR, it must be ensured that personal data can only be accessed by authorised individuals for the specific reason for which it was gathered and for the period it was gathered. The regulation emphasises data reduction. The information on each individual must be as small as possible. Regarding IAM, one approach to data minimisation is providing the least amount of access possible. While access must be limited, the organisation cannot restrict the authentication and authorisation that may be used. Many companies are developing, for example, centralised single-sign-on capabilities for all assets. This provides user convenience and organisational security, but it could be a potential GDPR problem if implemented globally. The risk is that users would get broad access to assets for which they are not authorised [49].

4.2 Payment Services Directive 2 (PSD2)

The revised Payment Services Directive (EU) 2015/2366 (PSD2) and the accompanying Regulatory Technical Standards (RTS) have had a significant impact on access control in European retail banks. The key component introduced by PSD2 is the mandate for Strong Customer Authentication (SCA), which requires financial institutions to ensure that electronic payments are performed with multi-factor authentication to increase security. PSD2 also introduces several exemptions to the SCA mandate, such as for low-risk payments or trusted beneficiaries [50], [51].

The RTS provide specific guidelines on how SCA should be implemented, focusing on security and communication protocols. Service providers should keep detailed logs and records of access events to create a traceable audit trail. This trail is essential for monitoring and reviewing access patterns, which can help identify potential security threats or breaches. PSD2 requires banks to create secure and efficient access interfaces, such as APIs, for Third Party Providers (TPPs) like Account Information Service Providers (AISPs) and Payment Initiation Service Providers (PISPs), enabling them to access customer account data for providing their services with customer consent [52].

Overall, PSD2 has substantially changed how access control is managed in European retail banks, emphasising security, user authentication, and the need for banks to adapt to new technologies and emerging fraud trends.

4.3 Financial Data Access Regulation (FIDA)

The European Commission has proposed a new Financial Data Access framework (FIDA), an initiative of the EU Digital Finance Strategy. FIDA grants consumers and small and medium-sized enterprises the right to authorise third parties to access their data held by financial institutions [53]. One of the primary impacts of FIDA on access control is that it mandates customer consent as a central element. Retail banks and other financial institutions will be required to ensure that data sharing is fully controlled by the customers, who must explicitly permit access to their financial data. This includes a broad range of financial information, not limited to payment accounts, which can now be shared with authorised third parties, such as fintech companies and other financial service providers [54]. Regulation is expected to be in force by 2027 at the latest. It will require significant adjustments in how banks manage access control, as it should be possible to manage fine-grain access for most financial services.

4.4 The Regulation on Electronic Identification and trust services for Electronic Transactions in the Internal Market (eIDAS)

The eIDAS Regulation was adopted by the European Council in July 2014 to help businesses, citizens, and public authorities interact securely through electronic means. The regulation provides a regulatory environment for electronic identification and trust services [55]. It ensures that citizens can use their own electronic identification schemes

(eIDs) to access public services in other EU countries. In practice, implementation of the regulation has faced several challenges. For example, varying interpretations of eIDAS regulations, different practices among member states, and insufficient collaboration and cooperation frameworks. The absence of a common EU-wide identifier also complicated the interoperability of eID systems across the EU [56].

4.4.1 The European Digital Identity Framework (eIDAS 2.0)

On 26 March 2024 the European Council adopted a new framework for a European digital identity (eID) [57]. Users can data, healthcare information, and electronic driving license information, to share and with whom, using digital identity wallets based on the specific use case and required security level [58]. The implementation of this framework marks a remarkable breakthrough in how digital identity management is conducted across the EU. Adopting a more integrated, secure, and user-centred approach will ensure a seamless and hassle-free user experience while guaranteeing the highest level of privacy and security. The Regulation will enter into force on 20 May 2024 with specific implementation deadlines [59]:

- by 21 November 2024, the Commission ‘shall establish a list of reference standards, specifications, and procedures for the requirements on the implementation and specifications and procedures for the certification of the European Digital Identity Wallet (EUDIW).
- by 21 November 2026, each Member State shall provide at least one digital identity wallet to its citizens and accept EUDIWs from other member states.
- no later than 21 November 2027 upon the voluntary request of the user, private relying parties shall also accept EUDIWs that are provided in accordance with the eIDAS 2.0. Banks must also accept EUDIW for secure customer authentication and transaction confirmation by that time at the latest.

4.5 Network and Information Security Directive (NIS-2)

The NIS-2 (Network and Information Security) Directive entered force on 16 January 2023. As a directive on measures for a cyber security in the European Union, the NIS-2 Directive aims to create a uniform level of protection for critical infrastructure networks and information systems and must be transposed into national law by 17 October 2024 [60]. In terms of access control, companies must implement robust systems to manage

and monitor access to sensitive information and critical systems. This includes ensuring access is restricted to authorised personnel, and adequate authentication mechanisms are in place to prevent unauthorised access. The directive encourages a shift towards more proactive cybersecurity practices, including regular audits, staff training, and the continuous assessment of cybersecurity risk. [61]

4.6 The Digital Operational Resilience Act (DORA)

The Digital Operational Resilience Act (Regulation (EU) 2022/2554)[62] is an EU regulation that entered into force on 16 January 2023 and will apply as of 17 January 2025. With the implementation of DORA, financial institutions must adhere to rules that govern the protection, detection, containment, recovery, and repair capabilities against ICT-related incidents. DORA explicitly addresses ICT risk and mandates regulations surrounding ICT risk management, incident reporting, operational resilience testing, and ICT third-party risk monitoring. This Regulation recognises that ICT incidents and a lack of operational resilience can threaten the stability of the entire financial system, even if there is "adequate" capital for traditional risk categories.[63]

4.7 Estonian Legal and Regulatory Acts

This chapter provides information regarding laws and regulations specific to the Estonian Republic.

4.7.1 Electronic Identification and Trust Services for Electronic Transactions Act

The act regulates electronic identification and trust services for electronic transactions for areas not regulated by the eIDAS regulation. Trust service providers must notify the competent authority within 24 hours of becoming aware of any security incident, and they must hold liability insurance with an annual sum of at least 1 million euros for each insured event [64].

4.7.2 Identity Documents Act (ITDS)

The act establishes the document obligation and regulates the issuing of identity documents to Estonian citizens and foreigners. The act specifies that the certificate enabling digital identification and the certificate enabling digital signing are linked to the user's personal data and can be publicly verified through the personal code [65].

4.7.3 Personal Data Protection Act (IKS)

This Act elaborates and supplements GDPR provisions for protecting natural persons' personal data. From the access management system perspective, it is important to consider that if services are provided directly to a child, the child's personal data processing is permitted only if the child is at least 13 years old. If the child is below 13, processing of personal data is permitted only in the case consent has been given by the child's legal representative [66].

4.7.4 Money Laundering and Terrorist Financing Prevention Act

This Act regulates measures to prevent the use of the financial system for money laundering and terrorist financing. The act sets obligations to identify a customer or a representative of a customer participating in a transaction. A financial institution is not allowed to provide services that can be used without identifying the person participating in the transaction, except in case the customer cannot make transactions until the full application of the due diligence measures [67].

4.7.5 Financial Supervisory Authority's Advisory Guide "Organizational Solution and Preventive Measures for Credit and Financing Institutions to Prevent Money Laundering and Terrorist Financing".

On 26.11.2018, the Estonian Financial Supervisory Authority issued a guide for measures to prevent money laundering and terrorist financing. Among other items, the guide specifies that in case the customer is onboarded remotely via self-service, the monthly turnover cannot exceed 15 000 euros for private customers and 25 000 euros for legal customers [68]. Banks must maintain effective limit handling procedures to ensure smooth management of customer relationships.

4.7.6 Cybersecurity Act

The act provides requirements for maintaining information systems and sets principles for preventing and resolving cyber incidents. The service provider must identify and analyse security risks to their system and implement appropriate technical and organisational measures for risk management, including continuity management, monitoring, auditing, and testing. The digital service provider must take appropriate measures to minimise the impact of a cyber incident on service continuity [69].

4.7.7 Credit Institutions Act

The act regulates the activities of credit institutions. Among other obligations, the credit institution is required to ensure the safety and regular monitoring of information technology systems used by the credit institution and systems used for the safekeeping of customers assets [70].

5 Stakeholder Requirements

Interviews are a crucial component of the requirements elicitation process when it is imperative to gain a comprehensive understanding of stakeholder needs for project success. In the SEB case, the stakeholder groups are diverse, leading to possibly conflicting or challenging needs that require resolution. The author conducted interviews with 37 individuals currently employed in various positions at SEB. Responsibilities included managing IAM, client services, and solution experts in IT and business development. The interviews are described in Appendixes 2, 3, 4 and 5.

Two areas were addressed in the analysis of the interviews - SHR1 "*What are the key success factors of a good implementation?*" and SHR2 "*What are the challenges in the current solution?*". Based on the responses, SHR2 was further divided into 3 categories – 1. Challenges in the current solution, 2. Future ideas, and 3. Issues not directly related to the CIAM solution; the latter part is not discussed further as it is not relevant to the scope of this thesis. The full list and category statistics for the stakeholder requirements elicited from the interviews are presented in Appendix 6.

5.1 Success Factors of Successful Implementation

One of the most important considerations for any redesign project is **to secure proper sponsorship** — "*the main thing is recognition of senior management because without investment you won't do much in this area*" was mentioned by expert 1.

Compared to internal IAM systems, **customer-facing systems must be very easy to use and understand**. Expert 1 mentioned, "*User centricity and user experience are probably the only thing the actual user sees*". The design of such systems must be handled professionally, and enough time and resources must be dedicated to the project.

It was also brought out that the biggest challenge is not building the solution but **handling the migration from the existing to the new model**.

5.2 Challenges in the Current Solution

One of the most frequently voiced issues in Latvia is the challenge of **company-based maximum daily limits**. The customer needs to contact the bank to change these limits; the available **limit is not visible to regular users**. This is uncommon for Lithuania and Estonia, as company limits are not widely used. In the current solution, user limits are not restored if a payment is rejected during review. Therefore, the client executive must **temporarily increase the limit** to allow the customer to still execute payments.

The **structure of the access rights model is complex and too detailed** for smaller companies. In the case of only a few users who need all access rights, the granularity is not needed - "*You know when you when you have only one accountant, and you need to tick all the boxes everywhere in order she or he would be able to do the things in the bank from the user experience perspective, it seems to bureaucratic or too difficult*". The **current model is not self-explanatory** and requires employees to refer to documentation to set up correct access rights. "*From one side it is good that we can split the rights in different directions and amounts. But from the other side, for customers, it is sometimes very complicated to understand what rights every employee needs as even here in the bank we are looking in the knowledge database if we need for the customers some definite right.*".

Another cluster of issues is related to administrators. Administrators are set as part of the service agreement, and **it is impossible to change administrators via self-service** as customers can do for regular users. This is especially difficult for customers with non-resident board members – "*Because I have more non-resident companies and there's a big problem, because there are business controllers, who have all rights to do all. Some of them have a power of attorney, but they do not have an opportunity to add to another administrator. We should ask for another power of attorney, or some of the board members should come to Estonia. It is complicated, it's very expensive for the companies. And so, they do not agree with it, and we should find some ways to add a new administrator. Some companies support members are changed every six months.".*

In Lithuania, there are some customers who have a restriction to use only Internet bank for businesses and are lacking access to private Internet banks. This is so because of

historical reasons and the bank is looking to remove this as it is restricting usage of some of the functionality like e-commerce payments and view of business bank cards.

From the solution architecture side, it was mentioned that in the current implementation, access rights are checked inside the functionality, making it hard to track and maintain. This has created a situation where there is a lack of a clear overview of the exact rules applied across the solution. This also results in challenges in maintaining the solution. As was brought out by expert 2: "*I think one thing that we already saw was that how do we actually work with the rights from a technical point of view that the service plans had this hiccup, that they had hard-coded available access rights in their side and after a new one was added, their service didn't work anymore*". As the solution has been evolving over many years, there is currently several internal API services that need to be used in combination to get needed access rights. "*For example, when you open the mobile app and we have to pull the list of customer accounts that are accessible by this customer, we have to call different APIs, at least three of them and then merge the results into one list because one of the APIs will give us which accounts are available for this company for this customer, another API will show us accounts which are available to this Internet bank contract. But it doesn't tell which company it is for, for example. Then, we need a third API to get the accounts' aliases. We also need to pull the account rights for those accounts in case we want to know whether we are showing the balances or not. So that's quite a lot*".

5.3 Future Improvement Needs

A periodic review of authorised users was brought out as one of the areas for improvement as this would help to reduce risks from banks' and customers' perspectives. "*This year, I have many clients, who want to see internet bank all users who have access*", was mentioned by expert 3. There is also a need for **a self-service solution to audit who has made a particular payment** from the customer account, as customers sometimes need to track the activities of users.

In some cases, even **more granularity is required for the access rights model**. There are cases when Internet bank is used to exchange sensitive documents between bank and customer, that person dealing only with payments, should not have access to – "*For example, AML team sends a request to provide some additional documents related to the*

payment, but if someone from payment unit will get access to the documents, it means that this person also is able to see the credit agreements or commercial conditions with the bank. It's quite confidential information". Also, **possibility to set up multiple temporary limits to several different date ranges** was mentioned. In the case of administrators, there have been cases where customers have requested the possibility to **limit actions that are allowed for administrators**. "*Admin in the bank can do everything, and business customers cannot limit this. Business customers would like to set like to grant admin rights to a particular person but wouldn't like this person to grant rights to another user, with 100% of that without some restrictions. You can be an administrator, but the CEO or board member could have the possibility to set up some restrictions for this administrator. This is from a risk perspective and really needed, especially in our current situation with some big incidents from a fraud perspective.*"

Possibility of having separate credentials for logging in to the business and private accounts. This is mostly related to larger customers who want to have more control over users. "*We probably should think about the possibility of setting up that the user can log in to one customer or to at least to business and private separately. Sometimes, the same person logs into many different, not related to business customers, and business customers wouldn't like to have such a risk that in case of fraud, the fraudster will have access to all business customers' accounts. It can be discussed, but anyway, it would be from a security perspective for the bank and for, at least for large corporates, a big plus.*" mentioned expert 4.

Simplification of managing access rights was mentioned on several occasions. This could be in the form of being able to copy access rights from one user to another or implementing standard pre-set access rights for a particular role. In case users need to get access to several companies or a group of companies, there could be a possibility to grant access rights in an easier way.

Managing access rights for private persons and special types of customers like bailiffs, etc., was highlighted, as there are use cases when this is needed – for example, having custodian rights over a private person or a simple parent-child setup.

The possibility of using foreign digital signatures would help to simplify customer service for non-resident customers. Currently, digital IDs issued, for example, in Spain or Belgium are not possible to use**.**

The usage of external registry data for managing access rights could also be a future possibility to further automate customer service. "*Estonian companies are registered in the business register, and if the board members are changing, maybe we can do this so that our bank system is smart enough to automatically get this information from the business register. And if there is a new board member, the system can somehow just automatically get this board member into our bank system. This is what our customers are lately asking.*"

6 Existing Solution Analysis

6.1 Digital Channels Context and Setup

SEB digital services have evolved as solutions have grown more complex. Internet bank solution consists of two main areas: private and business internet banks. Private Internet bank is the oldest model, introduced in 1999. The access rights model is relatively simple; a solution for business customers with support for multiple users and a more granular access rights model was added in 2006 and later received an update in 2011. The data model and overall architecture have remained unchanged, including duplication of code, and resulting problems associated with this, such as high cost of changes and being prone to errors [71]. In 2023, SEB launched a new internet bank platform that uses micro frontend architecture.

The concept of micro frontends (MFE) involves looking at a website or web application as a combination of features managed by separate teams, each with a specific area of expertise and responsibility within the organisation, developing its features from start to finish, including the database and user interface[72] as shown in Figure 8 [73].

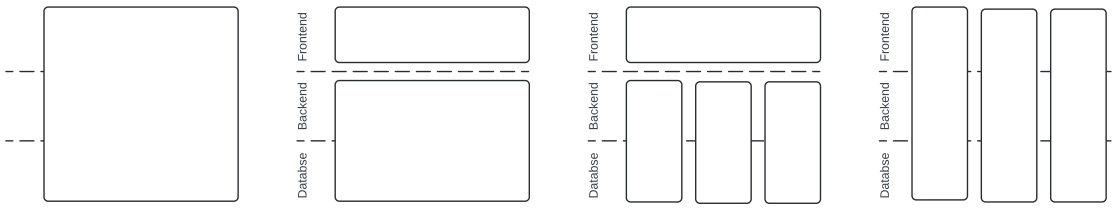


Figure 8. The architectural evolution. From monolithic to microservices to Micro-Frontends [73].

The change was made to enable scaling of development capabilities and reduce dependencies between various agile development teams. Currently, the solution is in the transition phase, with some functionality moved to a new micro frontend model and some in the legacy application. Once the transition is complete, it is expected that there will be 20 cross-functional teams managing 300 MFE's.

The mobile application uses the same access control model, and the bank also offers an API channel that provides a headless interface to external ERP solutions to access banking services. API channel also relies on the same digital channel's backend for authorisation. The bank also has separate API endpoints for providing access to Account Information Service Providers (AISP) and Payment Information Service Providers (PISP); these are licenced entities using customer consent to access account and payment services. Based on existing system documentation [74], the author created a context diagram of the digital channels. The diagram is shown in Figure 9.

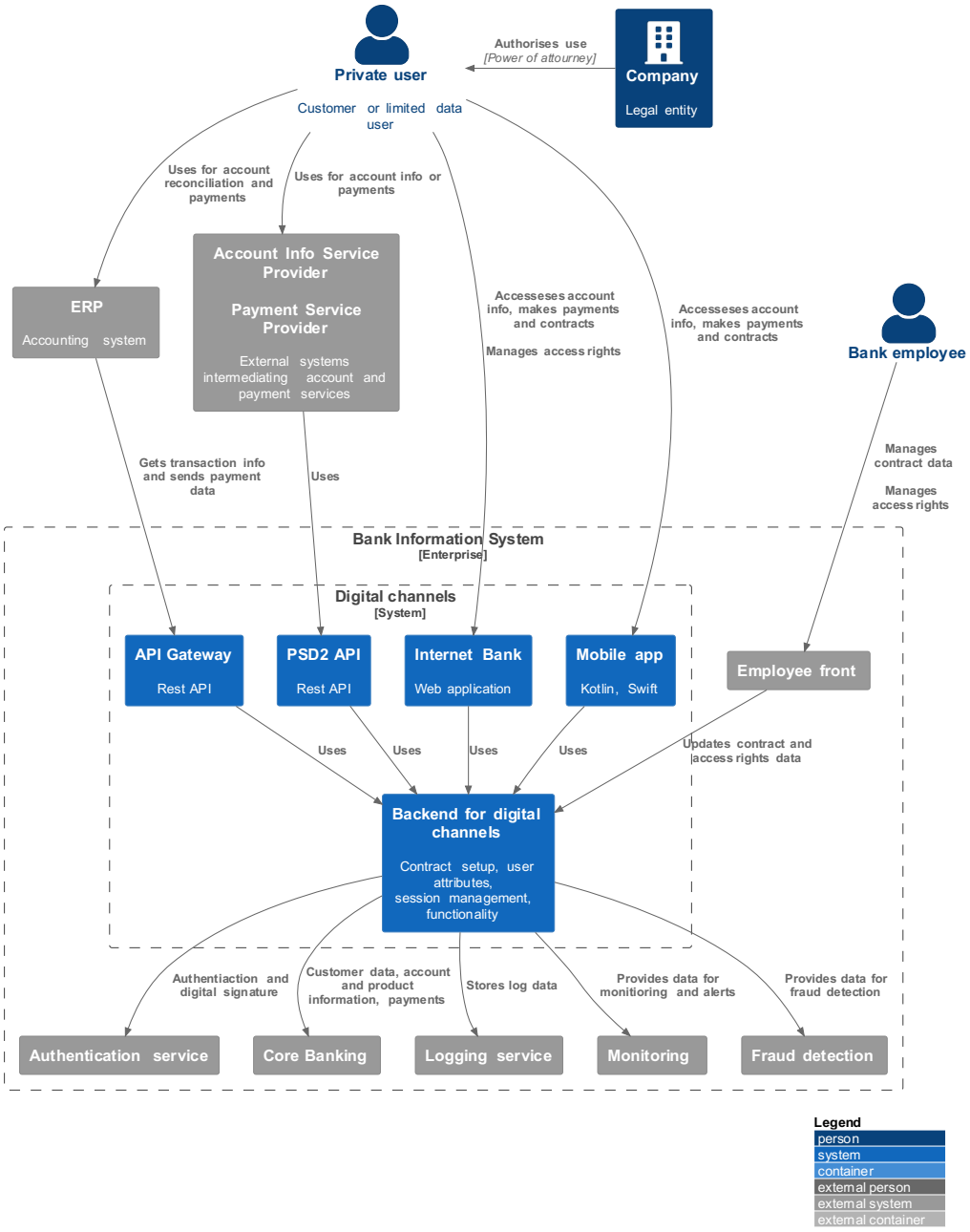


Figure 9. C4 Context diagram for banks' digital channels. Created by the author, based on [74].

Legal entities have users who are authenticated using their private credentials (for example, Smart-ID or Mobile-ID). Users can switch contexts within the internet or mobile application depending on whether they need to use personal or business services. Some private users have administrator roles, enabling them to manage access rights via Internet bank.

Bank employees have a separate employee front system to manage contract setup and access rights for legal entities. Digital channels' backend consists of different components dealing with session management, access control, providing business functionality, and data caching. Part of the backend is modernised and based on Java Spring Boot framework, legacy part of the systems uses PROGRESS 4GL and Progress Webspeed.

Authentication services are provided by a separate service that manages data exchange with external service providers like SK ID services and Latvijas Valsts Radio un Televīzijas Centrs for state and commercial authentication and signing services.

The bank is transitioning from a legacy monolithic front-end application to an MFE-based architecture. Today, the old part of the application is encapsulated in an iframe inside the new platform. Functionality is gradually being moved to a new architecture model, and the number of teams developing MFE applications is projected to grow.

Figure 10 describes the relevant elements inside the digital channels' backend. Figure is simplified and shows context in the example of only one service, other services use same pattern. Micro frontend (MFE) is a Javascript application running in the browser, requests data from the backend application. IB Gateway is an application that exchanges session information (session key, cookie, and client IP) to a JWT token and adds it as an authorisation header for backend applications to consume. JWT-s are also cached for a short time in the Token database for better performance.

The IB Session service is responsible for login and signing functionalities. It collects data about the user profile, creates a JWT token, and provides that to the IB Gateway.

The product's backend-for-frontend (BFF) application uses JWT tokens for authentication, requests needed data about user access rights attributes from the customer information service and decides whether the user is authorised to use the service.

The user rights database holds relevant user access rights attributes and is typically accessed via a customer information service, which provides API endpoints to access this information.

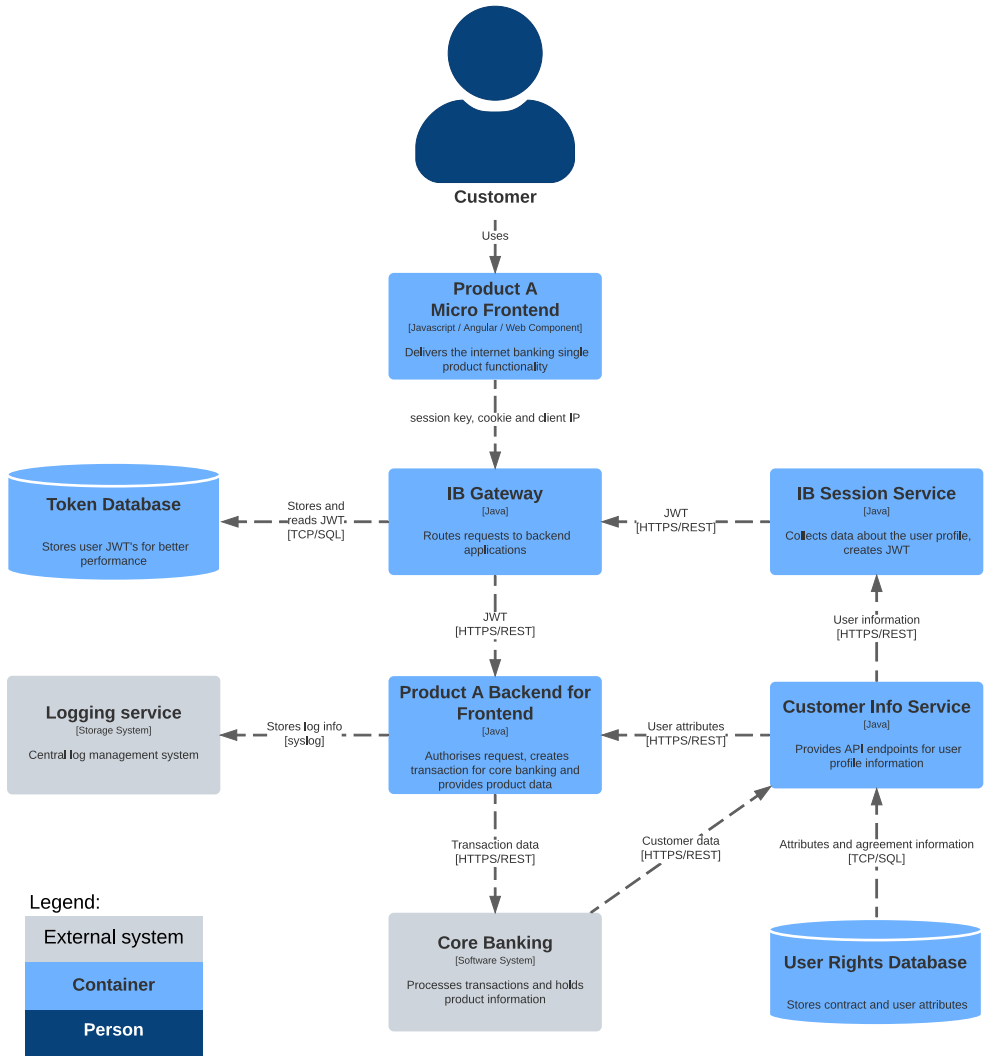


Figure 10. C4 Container diagram for digital channels backend. Created by the author, based on [74].

Policy decision points and policy enforcement points are inside the backend for frontend (BFF) module that provides business functionality to the user. As described in the sequence diagram in Figure 11, the backend for frontend (BFF) module requests access rights information from the repository and the module makes decisions regarding policy enforcement.

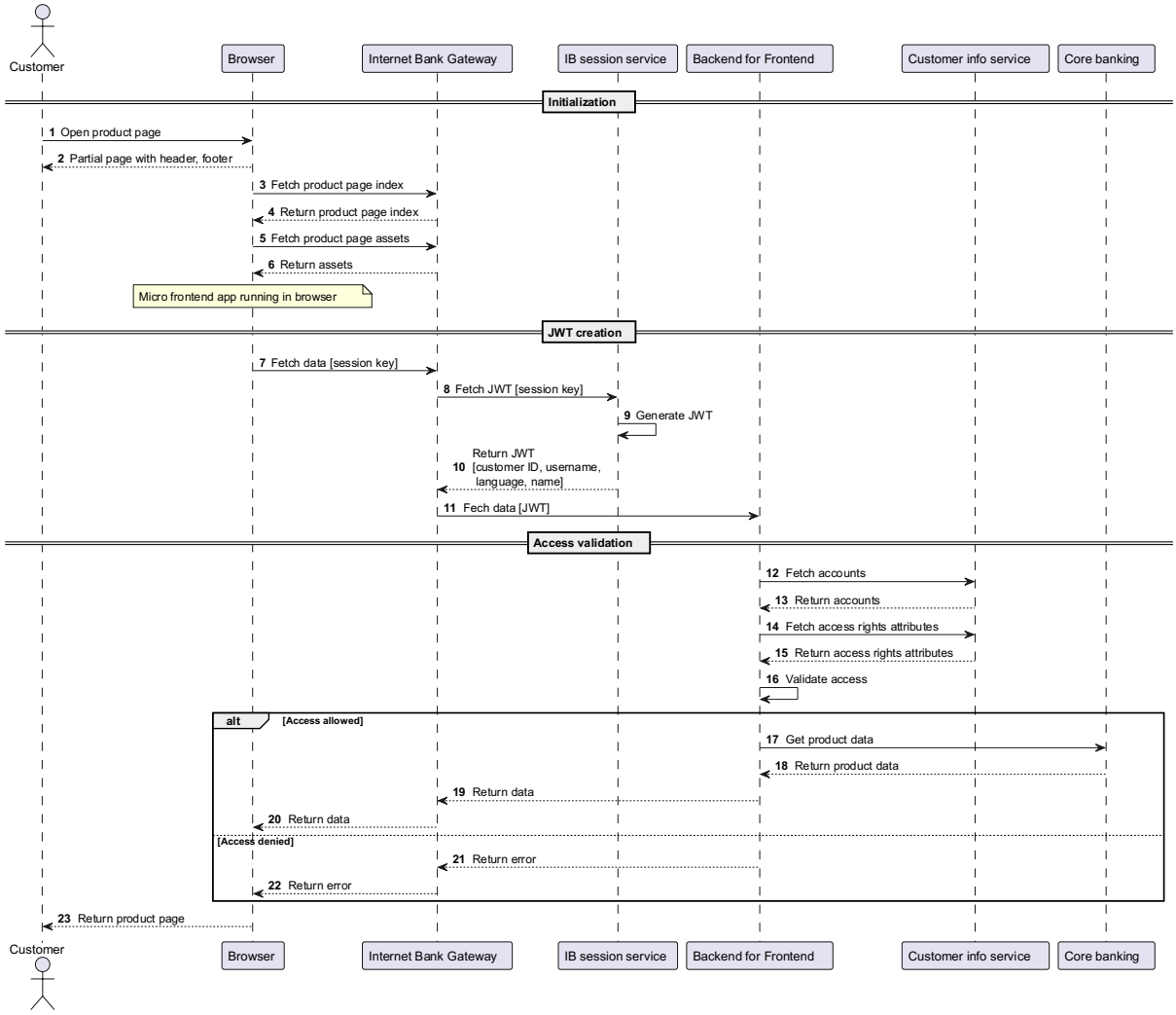


Figure 11. Sequence diagram for user rights enforcement [74].

The BFF application gets part of the customer data (ID, Name, Language) from Java Web Token (JWT) passed to the module by Internet Bank Gateway. It then needs to query accounts and applicable access rights and use this information to validate whether the user can access the requested service or not. This means that the authorisation decision is native to the application and each BFF application contains its own logic how to use the access rights for making authorisation decision.

6.2 Internet Bank for Private

Private customers can have several contracts, each distinguished by a username, which must be entered at the login. All users must have a personal digital identity solution (ID-Card, Smart-ID), or the bank will issue a device for them (PIN-Calculator).

Table 2 lists the authentication solutions supported in SEB online channels as of 2024.

Table 2. Authentication solutions used at SEB [74].

Device	**Notified eIDAS Level of Assurance**	**QSCD**	**Description**	**Notes**
ID-Card	High	Yes	Smart-card, PKI	Estonia only
Mobile- ID	High	Yes	SIM based, PKI	Not notified under eIDAS in Latvia and Lithuania
Smart-ID		Yes	Mobile app, split key PKI	
Smart-ID Basic		No	Mobile app, split key PKI	Latvia, Lithuania only Online enrolment
eParaksts mobile		No	Mobile app, server-based PKI	Latvia only
eID karte	High	Yes	Smart-card, PKI	Latvia only
Digipass		No	Hardware token providing one-time passcodes and challenge-response	
SEB Mobile App		No	Biometric identification in SEB mobile app is used for authentication and transaction confirmation	

There is a mix of solutions, some providing qualified digital signatures and some providing just identification and confirmation capabilities. The reason for maintaining several solutions is based on several aspects:

- compliance - it is mandatory in Latvia to support state-issued digital identity solutions.
- business continuity - to have redundancy in case one service provider experiences extended periods of service quality issues or the solution security is compromised.
- solutions availability to customers – users who lack access to state-issued identity solutions can use bank-issued solutions.

- cost – solutions based on external service providers are more costly compared to internal solutions.
- user experience and convenience – customers expect easy-to-use solutions.

The bank maintains a separate setup for defining which contracts and transactions can be signed/approved with which device.

Access rights are defined at the private Internet bank contract level; each contract has a unique username, status, and Digipass identification number if a device is issued to the customer. Current accounts need to be defined in the contract setup; customers' accounts and accounts belonging to other persons can be added if such a power of attorney exists. Accounts have limits as part of the information model; in the legal model, limits are decoupled from the contract. By default, standard limits apply, and the customer can change the limits without changing the contract. The class diagram for private Internet bank is shown in Figure 12.

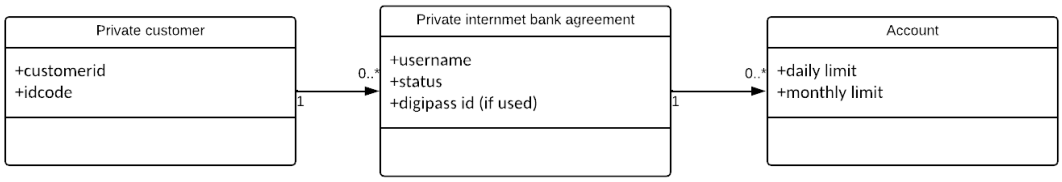


Figure 12. Class diagram for private Internet bank access rights.

When a customer initiates a new session after logging in, the system requests all deposits and securities accounts belonging to the customer and makes them available in the Internet bank.

If a user has power of attorney or custodian rights over another customer, the system does not automatically make these accounts available; they must be defined in the contract setup. Likewise, the contract setup must be changed if such rights are revoked.

Part of the access right model is based on the customer's attributes outside of the internet bank contract. If the customer is under legal age, there are restrictions on what services can be provided via digital channels. This also varies by country; for example, in Latvia,

customers aged 14-17 do not require parents' consent, while in Estonia and Lithuania, consent is required. In Lithuania, payments are restricted for underage customers, and in Latvia, parents can initiate withdrawal from child's savings deposits only if there is a court agreement.

As discussed in Chapter 4.7.5, there is a special limitation for remotely onboarded customers in Estonia where the total amount of payments outgoing for a natural person in any calendar month should be less than EUR 15 000 and, in the case of a legal person, less than EUR 25 000.

6.3 Internet Bank for Business

Internet banking for business customers enables multiple users and a more granular access rights model. Multiple users can be added to the contact. Each user is identified by his personal Internet bank contract, and the link between the person and the user in the contract is made using a personal ID code. Users can have administrative rights, which allows them to add or change users and their access rights.

There are several attributes of access rights at user and account levels. The class diagram in Figure 13 describes the user rights model and the full list is described in Appendix 7. The model and list of attributes have evolved; for example, in addition to general rights, which regulated access to non-account related services, several specific service-related access rights were added later, like access to the e-documents portal or the right to apply for loan disbursements.

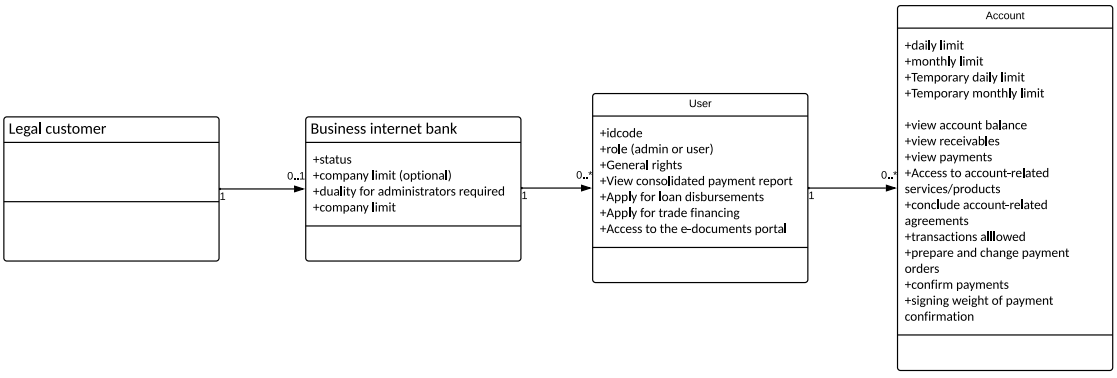


Figure 13. Class diagram for Business internet bank access rights.

Usually, a combination of several attributes is needed to allow customers to use the functionality. For example, opening a new account requires General rights, the Right to conclude account-related agreements, and Transactions permitted for at least one existing current account. There is no formal agreed-upon process for determining the required access rights; the usual practice is to agree on this ad hoc as new functionality is developed and added to the Internet bank.

As the system has been modernised and new functionality has been added, there has been a certain degree of inconsistency in the principles of what access rights are required for similar functionalities. The bank has reviewed the required rights, but this initiative has not yet concluded.

6.4 Analysis of User Rights Attributes

On the account level, 7 attributes can be set for the user. Theoretically, $2^7 = 128$ unique combinations can be set. The author performed an analysis of the system's actual user rights. This exercise aimed to determine whether such granularity in access rights is necessary. Data about the access rights of 588 871 accounts were extracted from the bank's system, and different user rights combinations were counted. 86 different combinations are used.

The authors' analysis showed that 84% of accounts have all 3 rights (view balance, view debit, and view credit). The distribution of attributes is shown in Figure 14. About 8% have a setup where they can only see debit transactions but no balance or credit information. The primary use case for this setup is related to payment preparation—90% of accounts with only the debit info attribute have the payment confirmation attribute present.

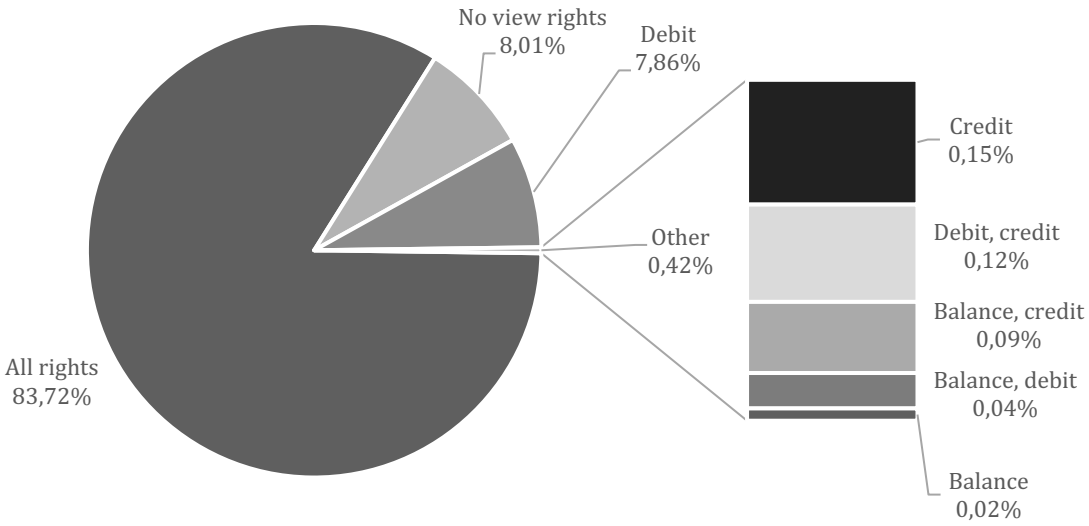


Figure 14. Distribution of account viewing attributes.

Almost 92% of accounts have binary attributes for account information viewing (have no right or have all rights), and only 8% use a more granular setup.

There is an attribute called “General rights of the company, " which is used in practice in combination with account-level attributes “Access to products and services” and “Right to conclude basic agreements.” There is ambiguity in the definition of these attributes, resulting in overlapping functions.

7 Discussion and Recommendations

An authorisation solution is central to the bank's ability to offer services through digital channels.

In the case of SEB, the model is partially based on ACL and ABAC principles (see Chapters 3.3.1 and 3.3.6), major shortcoming is that authorisation decisions are internal to each functional module of the system, and there is duplication of code as discussed in Chapter 6. The solution architecture and the way code is organised, does not follow the most modern principles of access control models as discussed in Chapters 3.3.6 and 3.5.

Changes to the current business logic of the access rights model are expected. Upcoming legal requirements, discussed in Chapter 4 will require banks to accept decentralised identities and verifiable credentials, NIS-2 and DORA directives enhance digital operational resilience requirements for the financial sector across the European Union, requiring clearer governance and reporting capabilities. Stakeholders have expressed the need to improve the functionality of the existing access control model by making it simpler and more flexible (see Chapter 5).

The bank has decided to move to a micro frontend architecture and development model. As a result, the number of teams developing their functional services will grow, and the issues related to coordinating changes in software, monitoring and auditing will grow. This creates challenges in case the business logic of the access rights needs to be changed.

The main problems identified are listed below:

1. **Deficiencies in the way the architecture is set up.**

    - Access control is not externalised.
    - Policy management is not clearly defined and governed.
    - High cost of making changes due to inflexible solution.
    - Difficult to audit and monitor.
    - Available internal services providing data for authorisation decisions are scattered between several endpoints.

2. **Complicated access rights model**

- The model is not self-explanatory to customers and employees.
- The model is too complex for smaller customers.
- Administrators are fixed in contract and cannot be changed via self-service.

The architecture choice for the authorisation model and structure of the attributes in the access rights model are discussed below.

7.1 Authorisation Architecture Choice

Regarding architecture choice, the main question is whether to centralise or continue with the as-is model where the authorisation model is part of each service. This decision will affect how the model is managed and how any possible changes will be introduced later. Several options exist:

Native authorisation pattern - authorisation will be performed at the service level. In this pattern, the authorisation decisions are native to the application. i.e. the code is handled within the application in its native source code [75]. Pattern is described in Figure 15. This is the current solution used by the banks' digital channels. Each service/function has embedded its local policy decision and policy enforcement points within the application.

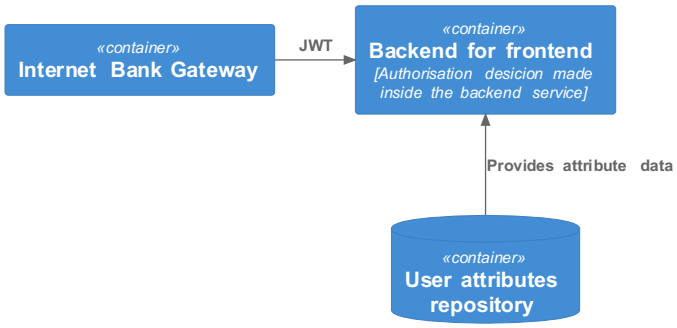


Figure 15. Native authorisation pattern [75].

This approach has worked successfully, as the solution's development has been centralised to a small number of teams, the frontend solution has been a monolith, and the data model for attributes has remained stable without significant changes.

Disadvantages:

- It would require extensive alignment and coordination between the product teams developing micro frontends and the team setting up principles for access control.
- Implementing sweeping changes in the micro frontend environment would be expensive and complex.
- Enforcing governance and inspection is challenging. Embedding authorisation logic in the service makes readability and inspection unfeasible in organizations where security or compliance teams must inspect such logic.

Proxy Pattern / Gateway Pattern. One possible solution to the authorisation model is to include a user's authorisation attributes and role information in requests routed to the services, for example, inside the JWT token. The pattern is described in Figure 16. This can be done at the proxy level between the browser and services. Then, the service can make its own authorisation decisions based on the provided information. This simple and clean solution allows developers of downstream services not to care about where attribute data is coming from [76].

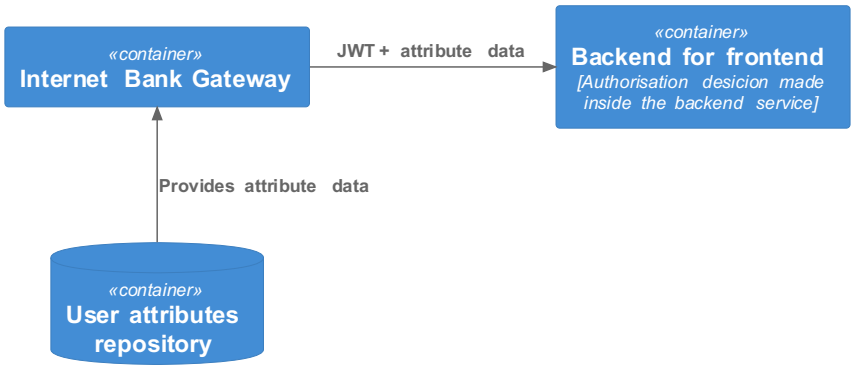


Figure 16. Gateway pattern [76].

Disadvantages:

- The request size can grow unreasonably large if there is a lot of permission data (for example, many accounts, attributes, or authorised companies).
- It might be inefficient to fetch all authorisation data for every request.
- As in the case of Native Pattern discussed earlier, it is difficult to enforce common governance and changes as the logic for making authorisation decisions resides in every individual service.

Centralised authorisation service pattern. This solution places all authorisation data and logic into one place, separate from all the services that need to enforce authorisation. The most common way to implement this pattern is to build a dedicated authorisation service [76]. In this model, the service doesn't need to care about the user's attributes; it just needs to ask the authorisation service whether the user can perform the transaction or whether a user can view the requested information. The model is described in Figure 17. The authorisation service itself contains everything it needs to make that decision. This option will simplify auditing existing access control policies and avoid fragmentation in a micro-frontend-heavy environment.

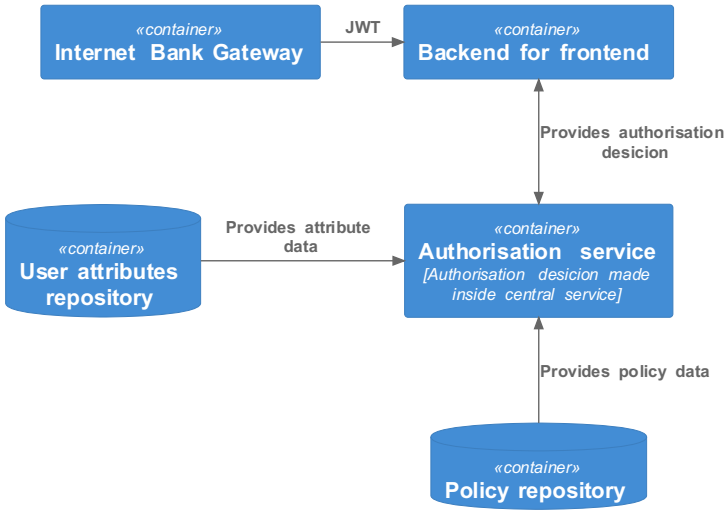


Figure 17. Central authorisation service pattern [76].

Changing policy can be done centrally without affecting many frontends. SEB could apply different policies to different customer groups (large corporation vs. small SME) if needed. Also, if such support is introduced in the future, it would be simpler to introduce support for verified credentials from EUIDW. From an implementation perspective, it is a significant change in access control; however, as the digital channels system is transitioning from monolith to micro frontend architecture, introducing this change is a good opportunity.

Disadvantages:

- Authorisation data needs to be collected in a single place. To enable decision-making, data should be either moved or replicated.
- The authorisation system must understand the data model, including contracts, account types, instances, and customer profiles.
- A change in a microservice might require an update to the authorisation service, which may introduce cross-team dependencies that the bank is trying to avoid.
- Challenges in achieving high availability and low latency by introducing a central component that all the services rely on.

The three options are evaluated by the author and compared in Suitability, Feasibility, and Acceptability aspects in the SFA-Matrix described in Table 3.

Table 3. SFA-Matrix for authorisation architecture options.

Group	**Aspect**	**Option1 Native authorisation**	**Option 2 Proxy**	**Option 3 Centralised authorisation**
Suitability	Is there a fit to banks overall strategy?	-	+	++
	Does the option fit the technology requirements?	-	+	++
	Does the option meet the needs of stakeholders?	-	-	++
Feasibility	Is there sufficient budget available for this option?	+++	+	+
	To what extent can the organisation bear the change of this option?	++	+	+
	Is the option technologically feasible?	++	++	+++
Acceptability	Is the level of risk acceptable?	-	+	++
	Is the likely return acceptable?	-	+	++
	Will stakeholder reactions be positive?	-	++	+
		Worst		Best

Suitability. From the suitability perspective, Option 3 is the best fit as only central authorisation will satisfy the required capabilities discussed in Chapter 3.5.2 and legal requirements imposed by the NIS 2 directive described in 4.5 regarding audibility, flexibility, and ease of administration. It is also the most preferred option from the stakeholder's perspective as it would enable better governance and reporting capabilities. It is aligned with banks' strategy to move to MFE model and considering technology requirement to be ready for cloud, this is also best choice by authors opinion as several commercial solution exist and have proven to be effective in cloud-based solutions.

Feasibility. All three options are technologically feasible. With Proxy approach, the risk is that excess information is transported in the JWT token, making it bloated and potentially resulting in performance issues. The question might arise whether introducing additional components in the centralised authorisation pattern would result in an increase in latency and performance degradation. This has been researched by Sänger and Abeck [77], who demonstrated that the externalised authorisation adds ≈2 ms to the median latency compared to the native implementation. There is also evidence of scalability of such solutions in more extreme installations, for example, Google Zanzibar, which serves more than 10 million client queries per second [78].

Continuing with the current native model will require the least investment in the short term. However, considering the future needs and possible changes required by the future needs, whether it is future needs discussed in Chapter 5.3 or upcoming advances required by the need to implement European Digital Identity Wallet (see Chapters 3.5.3 and **Error! Reference source not found.**) or changes required by adding more granularity for the access control to enable data access required by the upcoming FIDA regulation discussed in Chapter 4.3, it is clear that there is long term benefit in being able to handle authorisation service centrally.

It is complex change and requires effort from organisation to change the development approach, therefore the aspect "To what extent can the organisation bear the change of this option?" is rated lower, compared to staying with option 1.

Acceptability. Central authorisation services will help manage many potential risks related to responding to future needs and legal requirements compared to other options. From the perspective of the development teams, option 2 would be most preferred as it

would make it easier to get the data needed for authorisation decisions while maintaining full control and avoiding cross-team dependencies in development.

The author recommends introducing a centralised authorisation solution for digital channels' access rights. This best matches the recommendations discussed in Chapter 3.5.2 on page 32. While this option likely requires the highest investment, it is the only one that would ensure strategic fit in enabling audibility and uniform governance in the micro frontend environment. Authorisation-related changes should be explicit and traceable, best achieved by dedicated service. There are several commercial solutions available to use, such as OPA, OSO, and Amazon Cedar [79], [80], [81]. The benefits of ensuring compliance and flexibility in introducing sweeping changes in the future outweigh the downsides of central service for SEB.

7.2 Simplification of the Structure of the Access Right Model

Private customers have different contractual setups compared to business users – there is only one user, and part of the accounts are defined in the contract. The emerging pattern in the banks' information systems development is to automatically add access to certain types of accounts in digital banking – for example, deposits and investment accounts and accounts belonging to customers for whom the user has custodian rights (for example, children) are added automatically. As digital channels have become the main front for customer service, customers expect to see all their accounts when interacting with the bank via digital channels. For a long time, it was necessary to specify each current account individually in the contract because of the need to agree on the limits. In 2021, the bank changed the limit process; now, all customers get standard limits automatically, and it is possible to change them via self-service. This means it is feasible to take the next step and automatically add all current accounts.

There are contradicting stakeholder needs regarding business customer access rights — employees and small customers expect a simpler solution, while large customers sometimes require even more fine-grained access control. A role-based approach should be considered for simpler authorisation model needs. As discussed in Chapter 3.4, combining ABAC and RBAC models is possible. The bank is using an attribute-based model today, and in this case, the best option would be to use an attribute-centric approach, where a role identifier is added as one of the attributes.

The current model for account-level attributes has not been adjusted since its inception in 2006. Chapter 6.4 analysis showed that most customers do not use such a detailed access control model. Therefore, the author proposes changing the access rights structure by removing the possibility of differentiating whether users can see balance, debit, and credit transactions. One attribute specifying whether the user can see account information can replace this. The to-be account-level attributes are described in Table 4.

Table 4. To-be account-level attributes.

Current account-level attributes	**To be account-level attributes**
1. Right to view account balance 2. Right to view incoming payments 3. Right to view outgoing payments 4. Access to products and services 5. Right to conclude basic agreements 6. Right to prepare and change payment orders 7. Right to confirm payments 8. Daily transfer limit (EUR) 9. Monthly transfer limit (EUR) 10. Signing weight for confirming payments	1. View account information 2. Right to prepare and change payment orders 3. Right to confirm payments 4. Daily transfer limit (EUR) 5. Monthly transfer limit (EUR) 6. Signing weight for confirming payments

As discussed in Chapter 6.4, functions overlap for user-level attribute “general rights” and account-level attributes regulating whether users can access contracts. In addition, several additional user-level attributes enable more fine-grain control over the possibilities of accessing specific product areas (for example, trade finance or loan disbursement). The author proposes to move all product-level attributes to the user level, resulting in more homogeneous principles for attributes (transactions on the account level and contracts on the user level). The to-be user-level attributes are described in Table 5.

The bank should discontinue the “General rights” attribute as this is ambiguous and creates confusion among employees and customers. It can be replaced with the attribute of “access to products and services” on the user level.

Table 5. To-Be user-level attributes.

Current user-level attributes	To be user-level attributes
1. Administrator rights 2. General rights of the company 3. Access to consolidated payment report 4. Right to apply for the trade financing products 5. Right to apply for loan disbursement 6. Access to e-documents portal 7. Access to the data on the legal entity	1. Administrator rights 2. Role 3. Access to products and services 4. Right to conclude basic agreements 5. Access to consolidated payment report 6. Right to apply for the trade financing products 7. Right to apply for loan disbursement 8. Access to e-documents portal 9. Access to the data on the legal entity

Additional role attributes can be used to define users who, for example, need full access to everything or only viewing rights. This can then be combined with an access policy, and detailed specifications of access rights are not needed.

Most of the administrators in the business customers' Internet bank are legal representatives of the company; this implies that part of the user attributes could be fetched from external registries where this information is stored. During the rigour cycle, discussed in Chapter 9.1, it was stressed that it should not be the only option, though, as sometimes it is necessary to overrule this, for example, when there is a need to make sudden changes in the list of persons who can access company accounts.

One of the identified stakeholder needs is enabling duality for private customers in case of custodian relationships between different customers (parent-child, elderly-custodian); this can be achieved if private customers are using the same data model for access rights as it is created for business customers. In this case, role attributes can be used to identify the nature of the relationship between two private users.

8 Solution Design

The Solution Design chapter offers a comprehensive overview of the proposed system's architecture and functionality. It explains the components, software relationships, and interactions that define how the digital channels authorization system operates.

8.1 Main Use-Cases

The main use-cases relevant to the system functionality are shown in Figure 18.

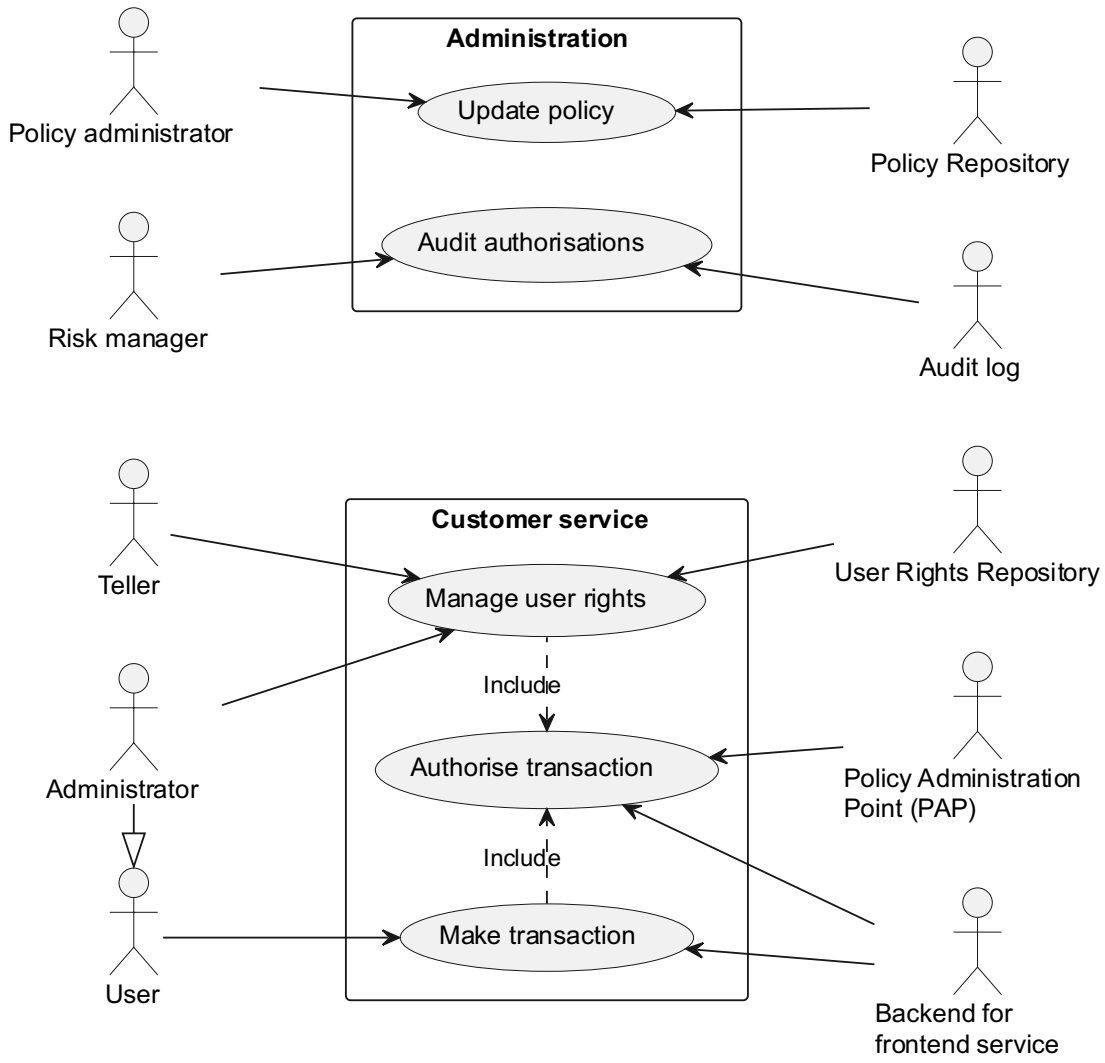


Figure 18. Use-case diagram.

Actors:

- Human
  - User: Customer who initiates financial transactions.
  - Administrator: Manages user rights and has administrative control over users. Administrator can also act as User.
  - Teller: Bank employee who manages user rights and permissions.
  - Risk Manager: Bank employee, who monitors risks and performs audits authorisation processes.
  - Policy Administrator: Bank employee who is responsible for updating and maintaining policy information.
- Systems
  - Policy Repository: Central storage for all policy-related information.
  - User Rights Repository: Repository for storing and accessing user rights information.
  - Audit Log: Keeps a record of all authorisation attempts and outcomes.
  - Backend for Frontend Service (BFF): Contains business logic for the specific service. Serves as an intermediary layer handling requests between the frontend interfaces and backend services.
  - Policy Administration Point (PAP): Manages policies.

Use Cases:

Make Transaction (UC1)

- Primary Actor: User
- Goal: To perform a financial transaction, conclude contract or get financial information.
- Trigger: User requests a transaction.
- Preconditions: User must be authenticated.
- Postconditions: Transaction is processed and logged.
- Main Success Scenario:
  - User initiates a transaction request.
  - Request is authorised by UC2.
  - Transaction is completed and recorded.

- Extensions:
    - 1a. If the transaction fails, an error is reported back to the user.
    - 1b. If request is denied, an error is reported back to the user.

Authorise Transaction (UC2)

- Primary Actors: Backend for Frontend Service, Policy Administration Point
- Goal: To authorise a transaction.
- Trigger: Transaction initiated by a user.
- Preconditions: The transaction must be valid, and the user must have the necessary rights. Policy Administration Point has needed policy and attributes data.
- Postconditions: Transaction is authorised.
- Main Success Scenario:
    - Backend for Frontend Service sends a transaction authorisation request.
    - Policy Administration Point evaluates and authorises the transaction.
    - Policy Administration Point writes log entry.
- Extensions:
    - 2a. If authorisation fails, transaction is blocked, and user is notified.

Manage User Rights (UC3)

- Primary Actors: Teller or Administrator
- Goal: To manage and update user rights and permissions.
- Trigger: Request to change user rights.
- Preconditions: Administrator must have administrative privileges. Teller must have valid customer order.
- Postconditions: User rights are updated.
- Main Success Scenario:
    - Administrator or Teller updates user rights in the User Rights Repository.
    - Request is authorised by UC2.
    - Update is completed and recorded.
- Extensions:
    - 3a. If the update fails, an error is logged, and the actor is notified.

Audit Authorisations (UC4)

- Primary Actor: Risk Manager
- Goal: To audit and review the correct application of policies.
- Trigger: Regularly scheduled audit or ad-hoc request.
- Preconditions: The audit report must be accessible.
- Postconditions: Audit report is generated.
- Main Success Scenario:
    - Risk Manager accesses the Audit Log.
    - Authorisations are reviewed, and discrepancies are noted.
- Extensions:
    - 4a. If any discrepancies are found, further investigation is initiated.

Update Policy (UC5)

- Primary Actor: Policy Administrator
- Goal: To update and maintain policy information in the system.
- Trigger: Policy change requirement or update.
- Preconditions: Policy Administrator must have permissions.
- Postconditions: Policy information is updated.
- Main Success Scenario:
    - Policy Administrator modifies policies in the Policy Repository.
    - Policy Administrator verifies that the policy is operating as intended.
    - Policy is applied to the authorisation rules.
- Extensions:
    - 5a. If the update fails, an error is reported.
    - 5b. If policy does not work correctly, error is reported.

8.2 Non-Functional Requirements

Non-functional requirements (NFR) are important for software development as they define the quality attributes that affect usability, performance, security, and maintainability. Author have defined NFRs for the proposed system based on ISO/IEC 25010:2023 SQuaRE (Systems and software Quality Requirements and Evaluation) [82].

1. Performance efficiency

NFR 1.1 The system should respond to authorisation requests within a defined timeframe.

- 95% of requests 10 milliseconds or less
- 99% of requests 25 milliseconds or less

Goal: Ensure the system responds to user interactions within a predefined time frame. Minimising the delay between a request and its corresponding response is important to enhance user satisfaction and system usability.

NFR 1.2 The system should handle 200 requests per second without performance degradation.

Goal: Ensure the system can handle peak system usage while maintaining acceptable performance levels.

2. Compatibility

NFR 2.1 Co-existence

Service must comply with loose coupling and high cohesion principles to prevent dependencies that could affect other services.

Goal: To ensure the application can co-exist with other microservices while sharing a common environment and resources without creating tight dependencies.

3. Interaction capability

NFR 3.1. Intuitive User Interface

The user interface for customers and employees must be intuitive and easy to use so that users can understand the features easily. User testing must not reveal any significant findings regarding access and use of the system.

Goal: Enable new users to navigate basic functions without guidance.

NFR 3.1 Accessibility of User Documentation

Comprehensive user documentation must be provided and be easily accessible within the system. Documentation shall include visuals and step-by-step instructions.

Goal: Good documentation reduces the learning curve and enables users to achieve basic competency with the system quickly.

NFR 3.2 Compliance with Accessibility Standards

The system shall be fully accessible, adhering to the WCAG 2.1 AA criteria to ensure that content is accessible to users with disabilities. Automated testing tools and manual reviews must confirm compliance.

Goal: To ensure that the system is usable by people with a wide range of abilities and disabilities.

4. Reliability

NFR 4.1. Automatic Recovery

In the event of a failure, the system shall automatically recover without data loss or corruption within 2 minutes.

Goal: To minimise system downtime and to keep business operations running smoothly.

NFR 4.2 Availability

The system should be available 99.8% of the time, minimising downtime. This is in line with other critical systems in the bank.

Goal: To minimise system downtime and to keep business operations running smoothly.

NFR 4.3 Redundancy

There should be redundant systems and data storage to ensure service continuity in case of hardware or software failure.

Goal: To maintain system operations in the face of single or multiple component failures.

5. Security

NFR 5.1 Confidentiality

All sensitive data, including credentials and permission details, must be encrypted in transit and at rest. Data transmitted across networks must use secure communication protocols (e.g., TLS) to prevent data tampering and eavesdropping.

Goal: To ensure that sensitive information is accessible only to those authorised to view it.

NFR 5.2 Data Validation

Before processing, the system shall validate all incoming data for accuracy, completeness, and adherence to format specifications. Input sanitisation shall be implemented to prevent SQL injection, cross-site scripting, and other attacks that could corrupt data.

Goal: To prevent inaccurate, incomplete, or inconsistent data from being stored or processed, which can lead to errors and problems in various applications.

NFR 5.3 Integrity of Audit Trails

The system must log all user access and actions for auditing purposes. Audit logs and other critical records shall be immutable and stored to prevent unauthorised alteration.

Goal: To detect unauthorised access, errors, and fraud.

NFR 5.4 Timestamping

The system shall employ a trusted timestamping service to record the exact time of transactions or data changes.

Goal: To create proof that certain actions occurred at a specific moment.

NFR 5.5 Access Control

Access to the system and its components shall be restricted to authorised users and applications, enforced through robust access control mechanisms.

Goal: To ensure that sensitive information is accessible only to those authorised to view it.

6. Maintainability

NFR 6.1 Analysability

The system should be capable of being effectively monitored and assessed to diagnose deficiencies or reasons for failure.

Goal: To enable monitoring and updating by the maintainers with effectiveness and efficiency.

7. Flexibility

NFR 7.1 Horizontal Scalability

The system must scale out seamlessly to handle increased load without performance impact by adding more nodes (servers, instances).

Goal: To handle growing (or shrinking) workloads and to ensure the system can handle increased performance requirements in the future.

NFR 7.2 Replaceability

It should be possible to replace a component with another product for the same purpose in the same environment.

Goal: By adopting measures to reduce the risk of being locked into a single software product, bank can enhance flexibility and remain open to exploring new and better alternatives.

8.3 High Level Architecture View

Diagram in Figure 19 describes relevant components in the system. The diagram shows the high-level shape of the software architecture and relationships between the components. The components are described in tables below on pages 75 to 77. Five existing components will require changes, and the model introduces six additional components required to externalise authorisation decisions.

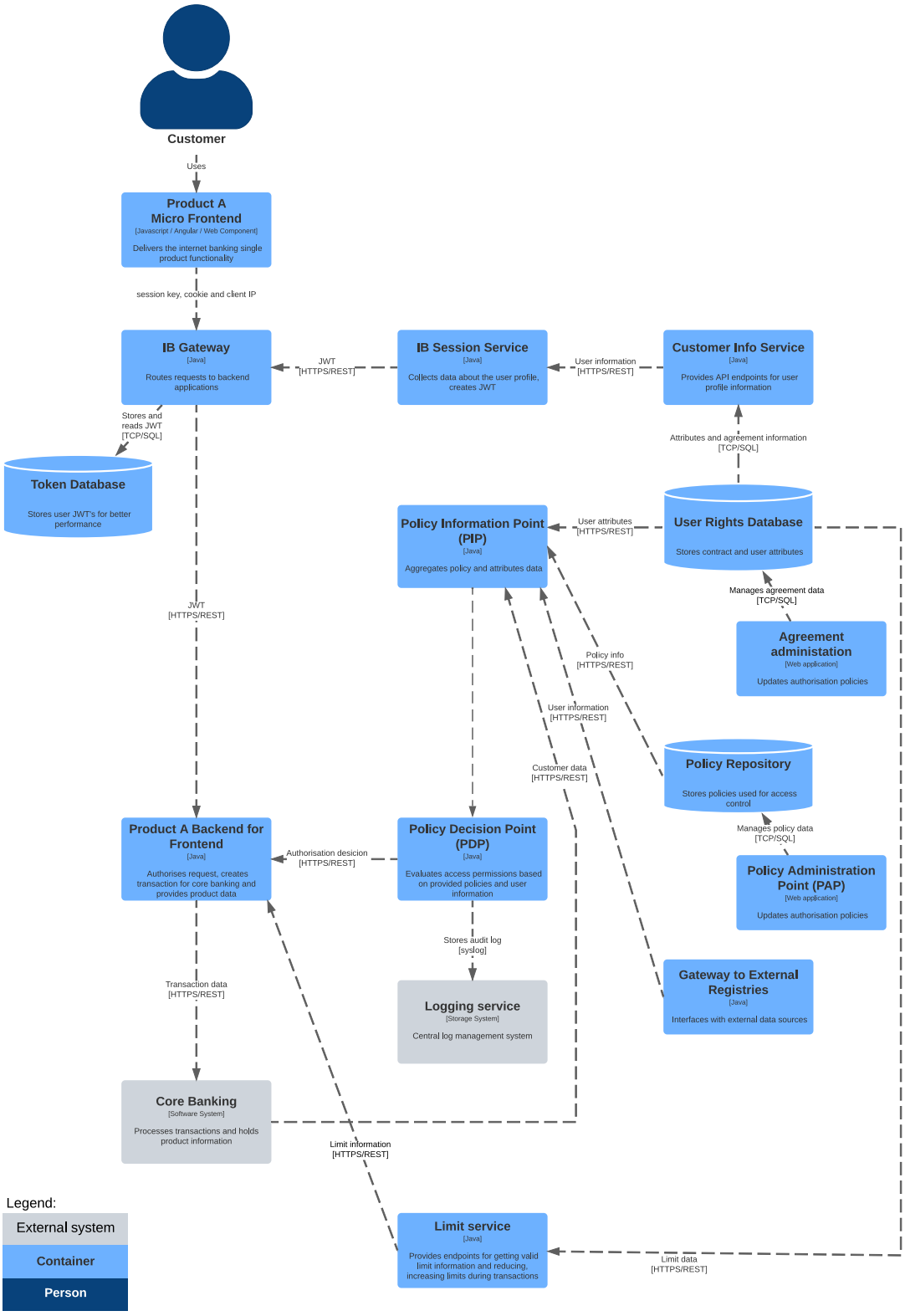


Figure 19. C4 Container diagram for the to-be digital channels authorisation system.

Table 6. Description of the components in the proposed system.

Component	Description	Functionality	Key Interactions
Existing components. No change required			
IB Platform	Main Internet Bank Web Component within the browser. Javascript application.	Serves as the customer-facing platform for internet banking solution, hosts all product related micro frontends.	Hosts micro frontends
Product A MFE	Example product specific Javascript application running in the browser, requests data from the backend application	Handles specific banking product related front-end functionalities	Interacts with IB Gateway for data exchange
IB Gateway	Application that exchanges session information (session key, cookie, and client IP) to a JWT token and adds it as an authorisation header for backend applications to consume.	Manages JWT tokens, routes requests	Interacts with Token database, IB Session Service and Backend for Frontend applications.
Token database	Caches JWT tokens	JWT-s are cached for short time	Receives and provides JWT tokens to IB Gateway
Core Banking	Manages core banking functionalities	Handles account management and transactions	Provides customer profile data to BFF and PIP
Logging Service (logging)	Captures and stores audit logs	Logs activities related to authorization decisions	Receives logs from PDP

Table 7. Existing components requiring changes in the proposed system.

Component	**Description**	**Functionality**	**Changes required**
IB Session Service	Responsible for login and signing functionalities. It collects data about the user profile, creates JWT token and provides that to IB Gateway	Generates JWT tokens for session management	Needs to include additional data about customer profile (accounts, aliases) in the JWT token.
Backend for Frontend (BFF)	Mediates between front-end and core systems	Product backend-for-frontend (BFF) application uses JWT for authentication	Needs to use customer data information from JWT. Policy decision point needs to be externalised.
Agreement Administration	Manages user agreements	Used by Bank employees to create and update internet bank agreements	Adjustments to the user interface to reflect change in the attribute model
Customer Info Service	Provides API endpoints for customer data	Provides customer profile information for session and transaction processing	Adjustments to the API to reflect change in the attribute model
User Rights Repository	Stores user rights and attributes	Manages data related to user rights and agreements	New data structure

Table 8. New components in the proposed system.

Component	Description	Functionality	Key Interactions
Policy Decision Point (PDP)	Makes authorization decisions	Evaluates access permissions based on provided policies and user information.	Receives policy data from PIP, interacts with Logging service and Backend for Frontend application
Policy Information Point (PIP)	Aggregates policy and attributes data from various sources.	Supplies policy data for decision making	Gathers data from external registries, policies, and provides info to PDP
Policy Administration Point (PAP)	Web application for managing policies	Used by the policy administrator to update authorisation policies	Stores policy data in Policy Repository
Policy Repository	Manages policy data	Stores policies used for access control	Provides policy data to Policy Information Point
Gateway to External Registries	Interfaces with external data sources	Enhances policy decisions with external data. Orchestrates data retrieval from various external data sources.	Provides external customer data to PIP
Limit service	Limit handling service	Provides endpoints for getting valid limit information and reducing, increasing limits during transactions	Provides limit information to BFF. Stores limit info in User Rights Repository

8.4 Component diagram

Components added to the proposed model are based on the logical models of ABAC and PBAC as discussed in Chapters 3.3.6. and 3.5.2. Figure 20 describes the components on the example of one backend service. It is important to have Policy Decision Point (PDP) module close to the backend application (in the same logical machine or container/pod), to ensure low latency of the communication. In the actual system, there can be hundreds of backend services, each with their own PDP.

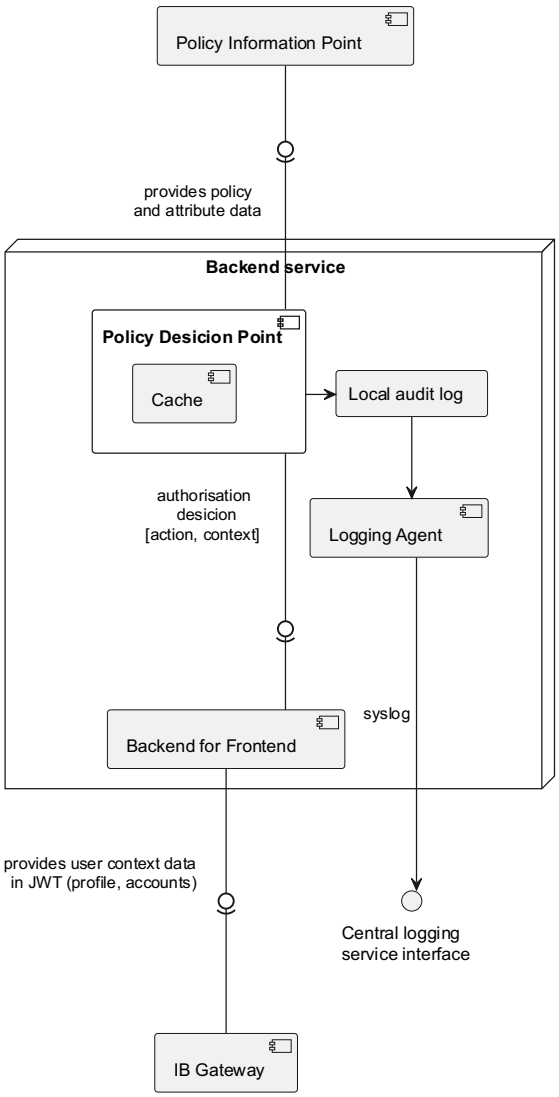


Figure 20. Component diagram of the authorisation components.

IB Gateway component provides the user context in the form of JWT. Token contains information about customer profile (Name, address, preferences, accounts). Backend for Frontend (BFF) application uses this information to request an authorisation decision from the Policy decision point (PDP) component. BFF provides context data together with the desired action. PDP retrieves applicable policies and attributes and evaluates whether transactions are allowed. Data is cached in PDP for better performance. The response will be returned to BFF, who can proceed with the transaction. PDP writes the event into a log, which is transferred to the central logging service for storage and auditing purposes.

8.5 Sequence Diagram

The sequence diagram in Figure 21 demonstrates the access validation flow of requests and data in an Internet banking system example when a customer tries to access a product page. The user is already authenticated, and there is a valid session key.

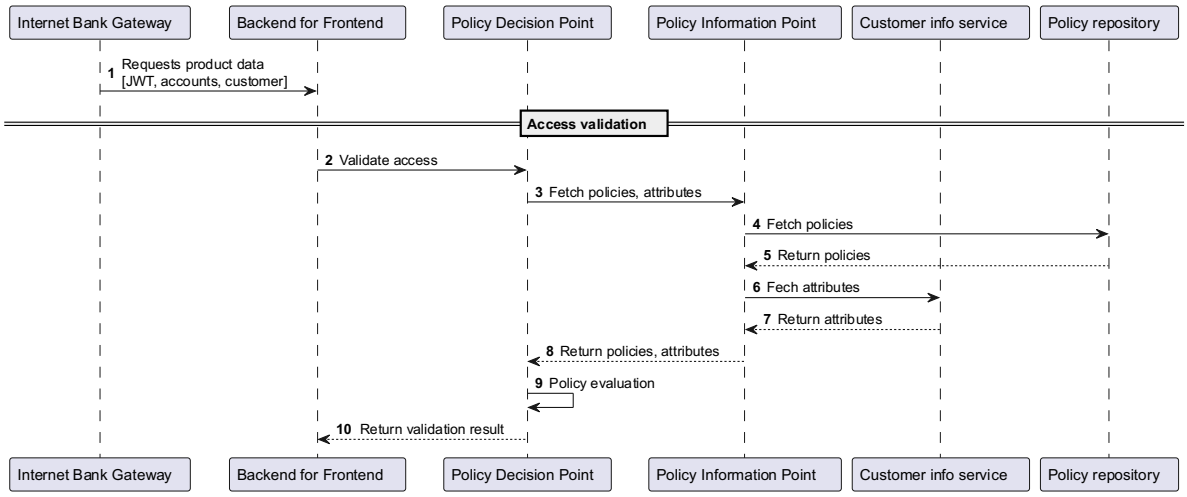


Figure 21. To-be sequence diagram.

Internet bank gateway routes request for data together with JWT to Backend For Frontend. JWT contains customer identifiers, accounts and language. Backend For Frontend requests authorisation decisions from Policy Decision Point (PDP). PDP asks for attributes and policies from Policy Information Point, which retrieves relevant policies and attributes from the Policy Repository and Customer Info Service.

Appendix 9 shows a larger diagram including interactions from the customer initiating the request up to the final delivery of the product page, depending on access permissions. The total flow is explained below.

Participants:

- **Customer**: Initiates requests via browser.
- **Browser**: Acts as the customer's interface. Runs micro frontend code for the internet bank.
- **Internet Bank Gateway** (IBGW): The gateway handles requests between the Browser application and backend services.
- Backend Services:
    - **IB Session Service**: Responsible for session management and JWT (JSON Web Token) creation. JWTs contain essential user context information such as customer ID, username, language, name, and account details. Fetches account details from the Customer Info Service
    - **Backend for Frontend**: Acts as a mediator between the specific micro frontend application and various core banking services. Uses Authorization Service to validate access.
    - **Policy Decision Point**: Service validates access rights by evaluating policies against the customer attributes, context, and action details.
    - **Policy Information Point**: Responsible for gathering policies and attributes necessary for the authorisation decisions.
    - **Customer Info Service**: Manages customer-specific information such as account details and personal attributes that may be required for processing requests or authorisation.
    - **Policy Repository**: Stores and manages access control policies.
    - **Core Banking**: This service contains the core functionalities related to banking transactions and data. It is crucial for retrieving the actual product data requested by the customer.

Sequence Details

- The customer opens the product page, leading the browser to fetch the product page.

- The browser launches a micro frontend app that dynamically loads components.
- Component in the browser requests session data from the IBGW, which then requests a JWT from the IB session service.
- The IB session service generates a JWT containing customer details and account information.
- Access Validation
- Backend for Frontend used data in JWT token to establish customer context (which customer, which accounts, etc.) and requests access validation for the desired action from the Policy Decision Point.
- The Authorization service interacts with the Policy Information Point to fetch relevant policies and attributes from the Policy Repository and Customer Info Service.
- After receiving policies and attributes, the Policy Decision Point evaluates these and returns the validation result to the Backend for Frontend.
- If Access Allowed, the Backend for Frontend fetches product data from Core Banking and returns it to the Customer. If access is denied, an error message is generated and returned following the same path back to the customer.

8.6 Entity relationship diagram

An Entity-Relationship Diagram (ERD) provides a visual and structured way to communicate the data needs and interactions within the proposed system. The conceptual diagram is shown in Figure 22 on page 82, and a detailed physical ERD with a description of data fields and semantics is included in Appendix 8. The data model is designed to consider the current structure of the service in the bank, ensure the fulfilment of stakeholder needs discussed in Chapter 5, and satisfy the best practice recommendations described in Chapters 3.4 and 3.5. Future improvements are expected based on decisions regarding vendor choice, feedback received during the proof-of-concept phase and needed migration steps.

This documentation outlines the structure and relationships of entities related to the access control model in the digital channels system. The entities for customer and account are included in the ERD solely to effectively communicate the relationships to the rest of the model. Any additional complexities of these entities are irrelevant to the current thesis.

For better readability, below is the conceptual data model. The detailed physical data model is included in Appendix 8.

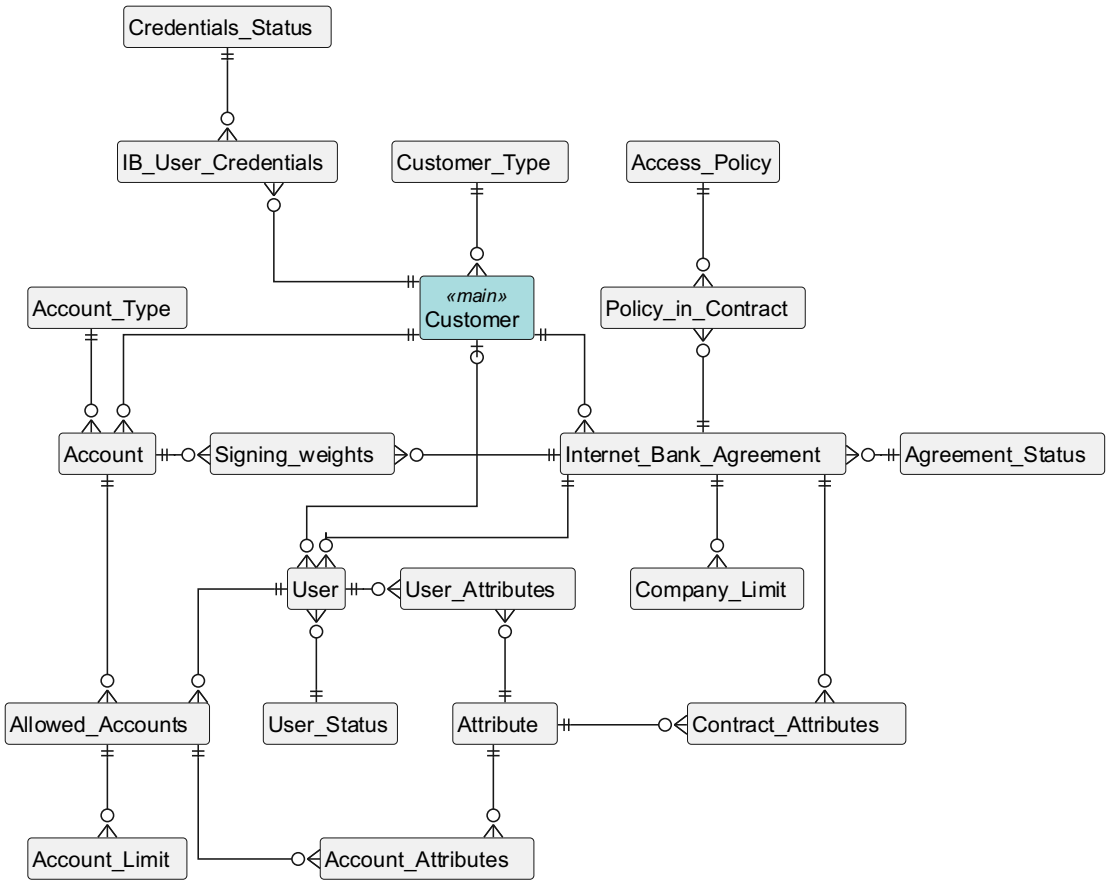


Figure 22. Entity relationship diagram of the access rights model.

One of the important differences compared to the current setup is that both private and business Internet bank contracts use the same data model. In the case of private customers, in the simplest setup, a private customer is a single user in the Internet bank contract. The model allows to set up several users for the private customer Internet bank, which was not possible before. User credentials are not directly connected to the particular contract but to the customer. Most authentication solutions used by the customers are issued by the state (see Chapter 6.2), this means that the bank could rely in the future fully on external identity credentials like ID-Card or European Digital Identity Wallet and would not need to have own issued identity credentials at all.

The new model satisfies the following properties:

- **Flexibility of the model**: All attributes in the data model can be dynamically changed; it is unnecessary to modify the data structures if additional attributes are needed for a user or account. Attributes can be on a contract, user, and account level.
- **Customer-specific models**: It is possible to store specific attributes only for certain users (for example, if a specific customer requires more fine-grain control).
- **Introduction of roles**: It is proposed that roles will be implemented via an attribute-centric approach, as discussed in Chapter 3.4, where a role name is just one of many attributes. It is not expected to have an elaborate structure of roles. Therefore, such an approach is a simple and effective solution.
- **Several parallel limits:** In the proposed model, an unlimited number of limits can be entered. Limits are time-bound, and business rules need to be set up to treat several applicable limits at the same time.
- **Central access control policies.** It is possible to define access control policies that can apply to all customers or only to specific customers.
- **Flexibility in adding users.** Users can be added to the contract without them being bank customers.

Table 9 below describes the entities present in the ERD.

Table 9. Entities in the data model.

Entity	Description
Customer	Represents the customers of the bank.
Customer_Type	Customers can be private, legal, or also respective partial data customers – used in case full KYC has not been completed.
Internet_Bank_Agreement	Manages the internet banking services agreements. Connects users to the agreement.
Agreement_Status	The status enumeration of agreement statuses to monitor active, blocked, or closed agreements.
User	Represents users who can access the banking system, with distinct access rights. Typically linked to a customer but it must be possible to enter users without having relevant customer record created. In this case user data is provisional and becomes valid once customer is created.
User_Status	The status enumeration of user statuses.
IB_User_Credentials	The authentication details for users of the internet banking system. Customer can have several credentials.
Credentials_Status	The status enumeration of credential statuses.
Account	Customer accounts, linked to specific types and statuses that determine the functionality and rules applicable to each account.
Account_Type	Enumeration of different types of accounts (e.g., current account, deposit, investment account).
Allowed_Accounts	Specifies which accounts a user has access to, including link to transaction limits and attributes.
Account_Attributes	Attribute values for attributes attached to the user accounts.
Account_Limit	Limits for user accounts. There can be several limits valid at any time as per stakeholder needs. Which limits to apply are specified by business rules.
User_Attributes	Attribute values for attributes attached to the user.
Attribute	Attribute enumeration.
Access_Policy	Policy repository for the access policies.
Policy_in_Contract	Policies connected to the internet bank contract.
Contract_Attributes	Attribute values for attributes attached to the internet bank contract.
Company_Limit	Limits for the agreement (company level limit). There can be several limits valid at any time as per stakeholder needs.
Signing_weights	Signing weights and limit values for the accounts.

9 Solution Validation

9.1 Iteration 1 – Stakeholder validation

The author conducted a workshop with stakeholders from various parts of the SEB organisation to validate the stakeholders' needs and improve the quality of the requirements. Findings and preliminary proposals were presented, and participants provided feedback and reflections.

Regarding the future access rights model, it was suggested that it is necessary **to interview some of the larger customers** later in the implementation to determine any additional needs. This is particularly relevant in case of customers having their own specific risk management approach.

Resulting adjustment to the access rights model: it should be possible to adjust access rights easily, and the system should support custom fine-grained access control for larger customers.

Another area discussed during the workshop was **the use of external registry data**, it was noted that even though it is beneficial to use external registry data, it should be possible to turn this feature on or off based on customer preferences. It was also noted that sometimes companies have joint representation rights, which can be quite complex. **Resulting adjustment to the access rights model:** It should be possible to choose whether external registry data is used for authorisation decisions.

The workshop participants validated the need for a central authorisation service. Stakeholders stressed the importance of improving the access rights model.

9.2 Iteration 2 – Architecture Review

The author presented the study results and proposals to the SEB Baltic Architecture Team, which consists of enterprise area architects and acts as the bank's main architecture decision-making body.

It was discussed that the bank has a specific area of transaction limits in the access rights model; typically, this is not part of features provided by commercial off-the-shelf

solutions. The context is different for access rights and payment; in the latter case, more data is required than regular action. Architects recommended separate services for authorisation and limits as this would better match a commercial solution for authorisation. **Resulting adjustment to the access rights model**: Separate endpoints for authorisation policy decision points and limits.

Another topic discussed was the possibility of **having several parallel authentication models**, as there are units like Life insurance and Leasing that serve different sets of customers. A similar need would be for internal users. It was agreed to address this at a later stage.

Architects also discussed the question of **policy ownership**. If several units own and develop their own services, who will manage and maintain the authentication policies? The conclusion was that it could work by having one team responsible for the policy or several teams contributing to it.

Resulting adjustment to the access rights model: A version control and governance model needs to be established for access rights policy maintenance.

Participants stressed that a central authentication service should be used for all channels, not only Internet banks.

Baltic Architecture Team approved the authentication service approach as the target solution for SEB and gave permission to proceed to the next step: creating a proof of concept to test the performance and gain practical experience.

10 Limitations

Customer access management in digital channels is critical in safeguarding sensitive financial information, ensuring compliance with various regulatory standards, and providing an excellent customer experience. Therefore, quality is extremely important. This chapter discusses the limitations of the current research.

Prototyping and end-user feedback

One significant limitation in the current research is the minimal involvement of prototyping and direct customer feedback during the initial stages of system design. This approach can lead to systems that, while technically compliant, may not address actual users' practical needs or usability concerns. After considering the possible scope, timeline and volume limitations of the thesis, the author decided to use other methods for analysis. The main argument is that the nature of the topic is very specific and technical, and it would be unlikely to get representative input directly from the customers. Instead, it was chosen to use front-line employees as a proxy for gathering customer needs as they interact with hundreds of customers every month and, therefore, have a wider understanding of the customer needs and issues. Also, data analysis was used to look at the actual usage of the system; this gives a more representative picture of the full customer base, helping in understanding the patterns of system use. Prototyping and user testing are planned in further steps during the implementation of the proposal.

Proof of concept

Another area where current research has limitations is in technical testing. Proof of Concept testing is crucial to demonstrate new technologies' feasibility and operational capabilities. Implementing advanced simulation tools and involving a broader range of operational parameters can provide deeper insights into the system's resilience and effectiveness. Instead, the author relied on existing research and best practice recommendations. Further work is necessary to create a proof of concept and to test the feasibility of various commercial solutions.

11 Future Works

Documentation plays a critical role in the software development process. It is important to have enough upfront design to start the development process and to facilitate meaningful conversations, it is also important to consider that no software or documentation is ever completely ready. This chapter describes the areas identified during the process that require further research and investigation.

1. Parallel authorisation services. During the rigour cycle, it was identified that a similar approach might be needed in other parts of the organisation. Further discussions need to be held to determine the feasibility of this.
2. Proof of Concept (PoC) helps determine whether a concept is technically and practically implementable before committing significant resources to the project. Several commercial solutions exist that can be tested during the PoC phase. As discussed in Chapter 5.1, top management support is very important in ensuring successful implementation. PoC provides evidence to stakeholders that can help to secure their support. It demonstrates the project's potential and its alignment with business goals.
3. Limits as a service. SEB architects suggested making limit checking a separate service as limits are not needed on every occasion, and some of the limits that need to be checked are not specific to digital channels — for example, customer limitation discussed in Chapter 4.7.5 and limits related to court orders or instructions from bailiffs. This needs to be further analysed and designed.
4. Support for decentralised identities. Legal requirements will be in place for banks to accept the EU Digital Identity Wallets in 2027. Once the solution standards have been set, support must be developed. Ensuring that a central authorisation service is in place by that time enables the bank to efficiently ensure support for the upcoming solution.
5. Prototyping and user testing. Once the new attribute model is accepted, a solution prototype and user testing are necessary. User centricity and user experience are important to deliver the proposed user rights model to users in an easy-to-understand way.
6. Transition strategy. Finally approach how to move from old to new model needs to be developed and agreed.

Summary

The thesis uses SEB Pank AS as a case study to design an architectural framework for managing customer access in digital channels within the financial services sector. The document comprises a detailed exploration of customer identity and access management (CIAM) theoretical foundations, including RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access Control), legal regulations, stakeholder requirements, and the practical application in creating a new access control system. Modern access control approaches include external components for access control decisions. Several legal acts require financial institutions to have effective governance and auditing capabilities for their information systems, including access control solutions.

The thesis highlights the current challenges SEB faces, including deficiencies in the architecture — authorisation decisions are internal to each functional module of the system, there is code duplication, and the access rights model is not self-explanatory and is too complex for smaller customers.

The author designed the architecture for a new customer access management system, in which access control decisions are externalised and policy management is clearly defined. Based on this work, SEB has decided to implement the proposed approach to access control in digital channels.

References

[1] 'OWASP Top 10 2021', Sep. 2021. Accessed: Mar. 25, 2024. [Online]. Available: https://owasp.org/Top10/

[2] S. Gregor and A. R. Hevner, 'Positioning and presenting design science research for maximum impact', *MIS Quarterly*, vol. 37, no. 2, pp. 337–355, 2013, [Online]. Available: http://www.misq.org

[3] A. R. Hevner, S. T. March, J. Park, and S. Ram, 'Positioning and presenting design science research for maximum impact', *Design Science in IS Research MIS Quarterly*, vol. 28, no. 1, pp. 337–355, 2004, doi: 10.25300/MISQ/2013/37.2.01.

[4] A. Hevner, 'A Three Cycle View of Design Science Research', 2014, [Online]. Available: https://www.researchgate.net/publication/254804390

[5] A. van der Merwe, A. Gerber, and H. Smuts, 'Guidelines for Conducting Design Science Research in Information Systems', 2020, pp. 163–178. doi: 10.1007/978-3-030-35629-3_11.

[6] A. Maedche, S. Gregor, S. Morana, and J. Feine, 'Conceptualization of the Problem Space in Design Science Research', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2019, pp. 18–31. doi: 10.1007/978-3-030-19504-5_2.

[7] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, 'A design science research methodology for information systems research', *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.

[8] V. Vaishnavi and B. Kuechler, 'DESIGN SCIENCE RESEARCH IN INFORMATION SYSTEMS', 2021. Accessed: Dec. 25, 2023. [Online]. Available: http://www.desrist.org/design-research-in-information-systems/

[9] J. Glöckler, J. Sedlmeir, M. Frank, and G. Fridgen, 'A Systematic Review of Identity and Access Management Requirements in Enterprises and Potential Contributions of Self-Sovereign Identity', *Business & Information Systems Engineering*, pp. 1–20, 2023.

[10] J. Higgins *et al.*, 'Cochrane Handbook for Systematic Reviews of Interventions version 6.4 (updated August 2023)', Aug. 2023. Accessed: Nov. 29, 2023. [Online]. Available: www.training.cochrane.org/handbook

[11] B. Kitchenham, 'Procedures for Performing Systematic Reviews', *Keele, UK, Keele Univ.*, vol. 33, Aug. 2004.

[12] J. R. Thomas, S. J. Silverman, and J. K. Nelson, *Research Methods in Physical Activity*, Seventh Edition. 2015.

[13] S. Brown, 'The C4 model for visualising software architecture', *InfoQ*, Jun. 25, 2018. Accessed: Feb. 02, 2024. [Online]. Available: https://www.infoq.com/articles/C4-architecture-model/

[14] 'Unified Modeling Language'. Object Management Group, Dec. 2017. Accessed: May 10, 2024. [Online]. Available: https://www.omg.org/spec/UML

[15] A. Vazquez-Ingelmo, A. Garcia-Holgado, and F. J. Garcia-Penalvo, ‘C4 model in a Software Engineering subject to ease the comprehension of UML and the software’, in *2020 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, Apr. 2020, pp. 919–924. doi: 10.1109/EDUCON45650.2020.9125335.
[16] J. Venable, J. Pries-Heje, and R. Baskerville, ‘FEDS: a Framework for Evaluation in Design Science Research’, *European Journal of Information Systems*, vol. 25, no. 1, pp. 77–89, 2016, doi: 10.1057/ejis.2014.36.
[17] A. Sharma, S. Sharma, and M. Dave, ‘Identity and access management- a comprehensive study’, in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, IEEE, Oct. 2015, pp. 1481–1485. doi: 10.1109/ICGCIoT.2015.7380701.
[18] Y. Yang, X. Chen, G. Wang, and L. Cao, ‘An Identity and Access Management Architecture in Cloud’, in *2014 Seventh International Symposium on Computational Intelligence and Design*, IEEE, Dec. 2014, pp. 200–203. doi: 10.1109/ISCID.2014.221.
[19] A. Cameron and O. Grewe, ‘An Overview of the Digital Identity Lifecycle (v2)’, *IDPro Body of Knowledge*, vol. 1, no. 7, Feb. 2022, doi: 10.55621/idpro.31.
[20] H. Rasouli and C. Valmohammadi, ‘Proposing a conceptual framework for customer identity and access management’, *Global Knowledge, Memory and Communication*, vol. 69, no. 1/2, pp. 94–116, Jul. 2019, doi: 10.1108/GKMC-02-2019-0014.
[21] M. Kuppinger, ‘How to Build the Modern CIAM: For Customers, Consumers, and Citizens’, Dec. 2022. Accessed: Jan. 18, 2024. [Online]. Available: https://www.kuppingercole.com/research/wp81239/how-to-build-the-modern-ciam-for-customers-consumers-and-citizens
[22] A. Koot, ‘Introduction to Access Control (v4)’, *IDPro Body of Knowledge*, vol. 1, no. 6, Jun. 2020, doi: 10.55621/idpro.42.
[23] B. Jayant.D, U. Swapnaja A, A. Sulabha S, and M. Dattatray G, ‘Analysis of DAC MAC RBAC Access Control based Models for Security’, *Int J Comput Appl*, vol. 104, no. 5, pp. 6–13, Oct. 2014, doi: 10.5120/18196-9115.
[24] V. C. Hu, D. F. Ferraiolo, and D. R. Kuhn, ‘Assessment of access control systems’, Gaithersburg, MD, 2006. doi: 10.6028/NIST.IR.7316.
[25] R. Sandhu, D. Ferraiolo, and R. Kuhn, ‘The NIST model for role-based access control’, in *Proceedings of the fifth ACM workshop on Role-based access control*, New York, NY, USA: ACM, Jul. 2000, pp. 47–63. doi: 10.1145/344287.344301.
[26] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, ‘Proposed NIST standard for role-based access control’, *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, Aug. 2001, doi: 10.1145/501978.501980.
[27] ‘ANSI INCITS 359-2004 Role Based Access Control’. American National Standards Institute, Inc., 2024.
[28] V. C. Hu, D. F. Ferraiolo, and D. R. Kuhn, ‘Assessment of access control systems’, Gaithersburg, MD, 2006. doi: 10.6028/NIST.IR.7316.
[29] E. Coyne and T. R. Weil, ‘ABAC and RBAC: Scalable, Flexible, and Auditable Access Management’, *IT Prof*, vol. 15, no. 3, pp. 14–16, May 2013, doi: 10.1109/MITP.2013.37.

[30] L. Wang, D. Wijesekera, and S. Jajodia, ‘A logic-based framework for attribute based access control’, in *Proceedings of the 2004 ACM workshop on Formal methods in security engineering*, New York, NY, USA: ACM, Oct. 2004, pp. 45–55. doi: 10.1145/1029133.1029140.
[31] E. Yuan and J. Tong, ‘Attributed based access control (ABAC) for Web services’, in *IEEE International Conference on Web Services (ICWS’05)*, IEEE, 2005. doi: 10.1109/ICWS.2005.25.
[32] V. C. Hu *et al.*, ‘Guide to Attribute Based Access Control (ABAC) Definition and Considerations’, Gaithersburg, MD, Jan. 2014. doi: 10.6028/NIST.SP.800-162.
[33] B. Jayant.D, U. Swapnaja A, A. Sulabha S, and M. Dattatray G, ‘Analysis of DAC MAC RBAC Access Control based Models for Security’, *Int J Comput Appl*, vol. 104, no. 5, pp. 6–13, Oct. 2014, doi: 10.5120/18196-9115.
[34] M. Kunz, A. Puchta, S. Groll, L. Fuchs, and G. Pernul, ‘Attribute quality management for dynamic identity and access management’, *Journal of Information Security and Applications*, vol. 44, pp. 64–79, Feb. 2019, doi: 10.1016/j.jisa.2018.11.004.
[35] D. Servos and S. L. Osborn, ‘Current Research and Open Problems in Attribute-Based Access Control’, *ACM Comput Surv*, vol. 49, no. 4, pp. 1–45, Dec. 2017, doi: 10.1145/3007204.
[36] A. J. Lee and M. Winslett, ‘Open Problems for Usable and Secure Open Systems’, in *ACM Conference on Human Factors and Usability (CHI)*, Montréal, Canada, Apr. 2006. Accessed: Feb. 05, 2024. [Online]. Available: http://d-scholarship.pitt.edu/id/eprint/16537
[37] D. R. Kuhn, E. J. Coyne, and T. R. Weil, ‘Adding Attributes to Role-Based Access Control’, *Computer (Long Beach Calif)*, vol. 43, no. 6, pp. 79–81, Jun. 2010, doi: 10.1109/MC.2010.155.
[38] A. K. Y. S. Mohamed, D. Auer, D. Hofer, and J. Küng, ‘A systematic literature review for authorization and access control: definitions, strategies and models’, *International Journal of Web Information Systems*, vol. 18, no. 2/3, pp. 156–180, Oct. 2022, doi: 10.1108/IJWIS-04-2022-0077.
[39] ‘What Is Dynamic Authorization?’, Ping Identity. Accessed: Apr. 21, 2024. [Online]. Available: https://www.pingidentity.com/en/resources/identity-fundamentals/authorization/dynamic-authorization.html
[40] ‘How Dynamic Authorization Enables Real-Time Policy Enforcement’, Jul. 2023. Accessed: Mar. 21, 2024. [Online]. Available: https://www.nextlabs.com/how-dynamic-authorization-enables-real-time-policy-enforcement/
[41] M. K. McKee, ‘Policy-Based Access Controls’, *IDPro Body of Knowledge*, vol. 1, no. 4, Apr. 2021, doi: 10.55621/idpro.61.
[42] M. von Mandel and A. Panday, ‘Policy-based access control in application development with Amazon Verified Permissions’. Accessed: Mar. 08, 2024. [Online]. Available: https://aws.amazon.com/blogs/devops/policy-based-access-control-in-application-development-with-amazon-verified-permissions/
[43] G. Williamson and M. Kuppinger, ‘Policy Based Access Management’, Feb. 2024. Accessed: Mar. 06, 2024. [Online]. Available: https://www.kuppingercole.com/research/lc80819/policy-based-access-management

[44] R. Soltani, U. T. Nguyen, and A. An, ‘A Survey of Self-Sovereign Identity Ecosystem’, *Security and Communication Networks*, vol. 2021, pp. 1–26, Jul. 2021, doi: 10.1155/2021/8873429.
[45] ‘Verifiable Credentials Data Model v1.1’. World Wide Web Consortium (W3C), Mar. 03, 2022. Accessed: Feb. 23, 2024. [Online]. Available: https://www.w3.org/TR/vc-data-model/
[46] ‘OpenID for Verifiable Credentials’. OpenID Foundation, 2024. Accessed: Mar. 30, 2024. [Online]. Available: https://openid.net/sg/openid4vc/
[47] K. Yasuda, T. Lodderstedt, D. Chadwick, K. Nakamura, and J. Vercammen, ‘OpenID for Verifiable Credentials A Shift in the Trust Model Brought by Verifiable Credentials’, Jun. 2022. Accessed: Mar. 24, 2024. [Online]. Available: https://openid.net/wordpress-content/uploads/2022/06/OIDF-Whitepaper_OpenID-for-Verifiable-Credentials-V2_2022-06-23.pdf
[48] *REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Da*. the European Parliament and the Council, 2016. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/2016-05-04
[49] N. Keitaanpää, ‘Regulations in Identity and Access Management’, 2022, Accessed: Dec. 16, 2023. [Online]. Available: https://urn.fi/URN:NBN:fi:amk-202202142453
[50] ‘Final Report on amending RTS on SCA and CSC under PSD2’. Apr. 05, 2022. Accessed: Apr. 20, 2024. [Online]. Available: https://www.eba.europa.eu/legacy/regulation-and-policy/regulatory-activities/payment-services-and-electronic-money-0#activity-versions
[51] ‘Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market’. 2015. Accessed: Apr. 20, 2024. [Online]. Available: http://data.europa.eu/eli/dir/2015/2366/2024-04-08
[52] ‘Commission Delegated Regulation (EU) 2018/389 of 27 November 2017 supplementing Directive (EU) 2015/2366 of the European Parliament and of the Council with regard to regulatory technical standards for strong customer authentication and common and secure open standards of communication’. Nov. 27, 2017. Accessed: Apr. 20, 2024. [Online]. Available: http://data.europa.eu/eli/reg_del/2018/389/2023-09-12
[53] C. Horwood, N. Sanghani, and T. Pearce, ‘Competitiveness of European financial services’, 2024. Accessed: Apr. 20, 2024. [Online]. Available: https://www.luxembourgforfinance.com/wp-content/uploads/2024/01/OMFIF-LFF-report-2024.pdf
[54] Si. Hansen, ‘How will Open Finance and the Financial Data Access Regulation impact the Financial Sector?’, EY Belgium. Accessed: Apr. 20, 2024. [Online]. Available: https://go.ey.com/44NvQ8m
[55] ‘eIDAS Regulation (Regulation (EU) N°910/2014)’. 2016. Accessed: Apr. 20, 2024. [Online]. Available: https://ec.europa.eu/futurium/en/content/eidas-regulation-regulation-eu-ndeg9102014.html
[56] S. Lips, N. Bharosa, and D. Draheim, ‘eIDAS Implementation Challenges: The Case of Estonia and the Netherlands’, 2020, pp. 75–89. doi: 10.1007/978-3-030-67238-6_6.

[57] ‘Press release: European digital identity (eID): Council adopts legal framework on a secure and trustworthy digital wallet for all Europeans’. Mar. 26, 2024. Accessed: Apr. 20, 2024. [Online]. Available: https://www.consilium.europa.eu/en/press/press-releases/2024/03/26/european-digital-identity-eid-council-adopts-legal-framework-on-a-secure-and-trustworthy-digital-wallet-for-all-europeans/
[58] S. Lips, N. Vinogradova, R. Krimmer, and D. Draheim, ‘Re-Shaping the EU Digital Identity Framework’, in *DG.O 2022: The 23rd Annual International Conference on Digital Government Research*, New York, NY, USA: ACM, Jun. 2022, pp. 13–21. doi: 10.1145/3543434.3543652.
[59] ‘Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework’, *Official Journal of the European Union*. Apr. 30, 2024. Accessed: Apr. 30, 2024. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L_202401183
[60] M. Falk and W. Dolle, ‘NIS-2 Directive: How companies can improve their IT security’, KPMG. Accessed: Apr. 20, 2024. [Online]. Available: https://kpmg.com/de/en/home/services/advisory/consulting/services/cyber-security/nis-2-directive.html
[61] O. Barthoumi, ‘The NIS 2 directive: what impact for European companies?’, Jul. 2023. Accessed: Apr. 20, 2024. [Online]. Available: https://www.wavestone.com/en/insight/directive-nis-2-cybersecurity-impact-european-companies/
[62] *Regulation (EU) 2022/2554 of the European Parliament and of the Council of 14 December 2022 on digital operational resilience for the financial sector and amending Regulations (EC) No 1060/2009, (EU) No 648/2012, (EU) No 600/2014, (EU) No 909/2014 and (EU) 2016/1011*. the European Parliament and the Council, 2022. Accessed: Mar. 05, 2024. [Online]. Available: http://data.europa.eu/eli/reg/2022/2554/oj
[63] ‘The Digital Operational Resilience Act (DORA) - Regulation (EU) 2022/2554’. Accessed: Mar. 05, 2024. [Online]. Available: https://www.digital-operational-resilience-act.com/
[64] ‘E-identimise ja e-tehingute usaldusteenuste seadus’, *RT I, 25.10.2016, 1*, Oct. 2016, Accessed: May 08, 2024. [Online]. Available: https://www.riigiteataja.ee/akt/103032023003
[65] ‘Isikut tõendavate dokumentide seadus’, *RT I 1999, 25, 365*, Jan. 2000, Accessed: May 08, 2024. [Online]. Available: https://www.riigiteataja.ee/akt/126042024013
[66] ‘Personal Data Protection Act’, *RT I, 04.01.2019, 11*, Jan. 2019, Accessed: May 08, 2024. [Online]. Available: https://www.riigiteataja.ee/akt/111032023011
[67] ‘Rahapesu ja terrorismi rahastamise tõkestamise seadus’, *RT I, 17.11.2017, 2*, Nov. 2017, Accessed: May 08, 2024. [Online]. Available: https://www.riigiteataja.ee/akt/106072023071
[68] ‘Finantsinspektsiooni soovituslik juhend „Krediidi- ja finantseerimisasutuste organisatsiooniline lahend ning ennetavad meetmed rahapesu ja terrorismi rahastamise tõkestamiseks“’. Nov. 26, 2018. Accessed: Apr. 20, 2024. [Online]. Available: https://www.fi.ee/sites/default/files/2018-11/FI_AML_Soovituslik_juhend.pdf

[69] ‘Küberturvalisuse seadus’, *RT I, 22.05.2018, 1*, May 2018, Accessed: May 08, 2024. [Online]. Available: https://www.riigiteataja.ee/akt/106082022018
[70] ‘Krediidiasutuste seadus’, *RT I 1999, 23, 349*, Feb. 1999, Accessed: May 08, 2024. [Online]. Available: https://www.riigiteataja.ee/akt/117032023017
[71] M. Martins, ‘Juurdepääsuõiguste süsteemi väljatöötamine SEB ärikliendi internetipanga näitel’, Tallinna Tehnikaülikool, 2011.
[72] M. Geers, ‘Micro Frontends extending the microservice idea to frontend development’. Accessed: Mar. 29, 2024. [Online]. Available: https://micro-frontends.org/
[73] D. Taibi and L. Mezzalira, ‘Micro-Frontends: Principles, Implementations, and Pitfalls’, *ACM SIGSOFT Software Engineering Notes*, vol. 47, no. 4, pp. 25–29, Sep. 2022, doi: 10.1145/3561846.3561853.
[74] ‘Existing system documentation’. SEB, 2024.
[75] J. Lindbakk, ‘Authorisation Patterns for Monoliths and Microservices’. Accessed: Apr. 01, 2024. [Online]. Available: https://lindbakk.com/blog/authorisation-patterns-for-monoliths-and-microservices#native-authorization-pattern
[76] G. Neray, ‘Best Practices for Authorization in Microservices’. Accessed: Apr. 01, 2024. [Online]. Available: https://www.osohq.com/post/microservices-authorization-patterns
[77] N. Sänger and S. Abeck, ‘User Authorization in Microservice-Based Applications’, *Software*, vol. 2, no. 3, pp. 400–426, Sep. 2023, doi: 10.3390/software2030019.
[78] R. Pang *et al.*, ‘Zanzibar: Google’s Consistent, Global Authorization System’, in *2019 USENIX Annual Technical Conference (USENIX ATC ’19)*, Renton, WA, 2019.
[79] ‘Cedar Language’. Accessed: Apr. 12, 2024. [Online]. Available: https://www.cedarpolicy.com/en
[80] ‘Oso: Authorization as a Service’. Accessed: Apr. 12, 2024. [Online]. Available: https://www.osohq.com/
[81] ‘Open Policy Agent’. Accessed: Apr. 12, 2024. [Online]. Available: https://www.openpolicyagent.org/
[82] ‘ISO/IEC 25010:2023(E): Systems and software engineering —Systems and software QualityRequirements and Evaluation(SQuaRE) — Product quality model’, *International Organization for Standardization*. Nov. 2023. Accessed: Apr. 14, 2024. [Online]. Available: https://www.iso.org/standard/78176.html

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[2]

I

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Architecture Design for the Customer Access Management of Digital Channels in a Financial Services Company Based on SEB Pank AS Example”, supervised by Silvia Lips and Henrik Leinola
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

2 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – Interview with Identity and Access Management Expert at SEB Baltics

Interview date: 16.01.2024.

Questions:

1. What are the key features and technologies that are considered essential in modern customer IAM solutions?
2. What are the most significant challenges organizations' face when adopting customer IAM solutions?
3. What emerging technologies (like AI, blockchain, etc.) do you see having the most significant impact on customer IAM solutions?
4. What challenges do you foresee for customer IAM solutions in the next five years?
5. How do you see the user experience evolving in customer IAM solutions?
6. Can you share some best practices for implementing and managing customer IAM solutions?
7. Are there any specific features or considerations for customer IAM solutions in retail banking?
8. Are there any specific features or considerations for customer IAM solution that SEB needs to consider?
9. Do you want to add anything else?

Appendix 3 – Interview with Small and Medium Enterprise Service Experts

Group interview date: 19.01.2024.

Units of the interviewees.

Unit	**Country**
Customer Centre Department	Estonia
Customer Centre Department	Estonia
Business Customers Department	Estonia
Large Customers Department	Estonia
Customer Centre Department	Estonia
Financial Centre of Ridzene	Latvia
Segment Management	Latvia
Real Estate Financing Department	Latvia
Business Customers Service Department	Latvia
Business Customers Daily Banking Group	Lithuania
Business Customers Daily Banking Group	Lithuania

Questions:

1. Please describe the current SEB customer IAM solution.
2. What would you consider as the strong points in the current solution?
3. What are the most significant challenges that the customers are facing in relation to the current cIAM solution?
4. What are the most significant challenges that you, as a front-line employee, are facing in relation to the current cIAM solution?
5. What do customers expect from the cIAM solution?
6. Are there any specific features or considerations for customer IAM solution that SEB needs to consider?
7. Do you want to add anything else?

Appendix 4 – Interview with Corporate Customer Service Experts

Interview date: 22.01.2024.

Units of the interviewees.

Unit	**Country**
Corporate banking	Estonia
Corporate banking	Estonia
Corporate services and financial markets	Latvia
Financing Products Division	Latvia
Business service department	Latvia
Real estate financing department	Latvia
Real estate financing department	Latvia
Cash Management and Trade Finance unit	Latvia
Cash Management and Trade Finance unit	Latvia
Corporate banking	Lithuania
Corporate banking	Lithuania
Business Clients Department	Lithuania
Funds Management Department	Lithuania
Funds Management Department	Lithuania
Corporate Segment unit	Lithuania

Questions:

1. Please describe the current SEB customer IAM solution.
2. What would you consider to be the strong points in the current solution?
3. What are the most significant challenges that the customers are facing about the current cIAM solution?
4. What are the most significant challenges that you as a front-line employee are facing in relation to the current cIAM solution?
5. What do customers expect from the cIAM solution?

6. Are there any specific features or considerations for customer IAM solution that SEB needs to consider?
7. Do you want to add anything else?

Appendix 5 – Interview with Solution Experts

Group interview date: 8.02.2024

Units of the interviewees.

Position	**Country**
Product owner Internet bank	Estonia
Business developer	Estonia
Product owner API channel	Estonia
Senior Analyst	Estonia
Senior Business Developer	Lithuania
Enterprise area architect	Estonia
Product owner Internet bank	Lithuania
Product owner Mobile	Lithuania
Solution Architect Internet bank	Lithuania
Solution Architect Mobile channel	Lithuania
Solution Architect API channel	Lithuania

Questions:

1. Please describe the current SEB customer IAM solution.
2. What would you consider as the strong points in the current solution?
3. What are the most significant challenges that the customers are facing in relation to the current cIAM solution?
4. What are the most significant challenges that the front-line employees are facing in relation to the current cIAM solution?
5. What are the most significant challenges that the development organization is facing in relation to the current cIAM solution?
6. What do customers expect from the cIAM solution?
7. What challenges do you foresee for customer IAM solution in the next five years?
8. Are there any specific features or considerations for customer IAM solution that SEB needs to consider?
9. Do you want to add anything else?

Appendix 6 – Stakeholder requirements

SHR1 “What are the key success factors of a good implementation?”

Category ID	Category Name	Absolute Count
SHR1-1	System should be easy to use	8
SHR1-2	Roles should be available	1
SHR1-5	Challenges in getting priority must be solved	2
SHR1-6	Important to follow proven models	1
SHR1-7	Artificial intelligence should be avoided for now	1
SHR1-8	Okta should be used as the best practice examples	1
SHR1-10	Securing support from senior management is important	1
SHR1-11	There should be migration plan from old to new system	1
SHR1-12	Decentralised identities are possible future trend	2

SHR2 “What are the challenges in the current solution?”

	Category ID	Category Name	Absolute Count
Problems in current IAM			54
	SHR2-1	Changing limits and access rights should be simple	1
	SHR2-2	Maximum daily limits in Latvia are low and it is difficult to change	6
	SHR2-3	It should be possible to provide power of attorney to change limits so that the admin should not be management board member.	1
	SHR2-4	Structure of access rights for small companies should be simpler	1
	SHR2-5	Opening new account should provide limits to the newly opened account.	2
	SHR2-8	Available limits should be visible to the user	3
	SHR2-9	It should be easy for employees to manage user rights	2
	SHR2-14	Number of different access rights should be reduced	1
	SHR2-11	Access rights should be easy to understand	9
	SHR2-10	Access rights administration must be optimized for mobile screens	1

	Category ID	Category Name	Absolute Count
	SHR2-12	It should be possible to add administrators via self service	6
	SHR2-19	After opening new deposit, access rights need to be added for that deposit	1
	SHR2-22	When added user is not SEB customer, then access rights should become available without the need to re-login, for that that user to gain access	2
	SHR2-24	Used limit amount should be adjusted in case payment is cancelled	1
	SHR2-35	It should be easy to track and maintain logic related to checking access rights	2
	SHR2-36	User access data should be available via single API service	3
	SHR2-38	Some users should have restricted right to use only business internet bank	2
Not IAM related			8
	SHR2-7	Account statement should be visible longer than 1 year period	1
	SHR2-6	KYC data update should be possible to non-board member	1
	SHR2-20	It should be clear for customers how to order additional vs new credit card	1
	SHR2-21	It should be possible to attach additional documents to applications	2
	SHR2-18	It should be possible to have multiple confirmation for applications and agreements	5
	SHR2-26	It should be possible to sign credit card agreement after the application is submitted	2
	SHR2-29	If payment is rejected, the rejection reason needs to be given to customer	2
Future ideas			20
	SHR2-13	It must be possible to have periodic review of access rights	3
	SHR2-23	It must be possible to set up temporary limits for several different date ranges	1
	SHR2-25	It should be possible to have separate right to upload and work with confidential documents	2
	SHR2-15	It must be possible to accept foreign digital signatures	1
	SHR2-16	It must be possible to use external registry data for board members access	5

	Category ID	Category Name	Absolute Count
	SHR2-28	It must be possible to audit, who has done which action	1
	SHR2-30	It must be possible to limit what administrator can do	1
	SHR2-27	It must be possible to add access rights to a group of companies at once	1
	SHR2-31	It must be possible to have role-based access rights for simpler setup	3
	SHR2-32	It should be possible to copy access rights for easier setup	1
	SHR2-33	It must be possible to separate login credentials to business and private accounts	1
	SHR2-34	It must be possible to have power of attorney over private person	1
	SHR2-37	It must be possible to support special type of customers (bailiffs, etc)	3

Appendix 7 – Internet Bank for Business user rights

User based rights

Administrator rights	Right to add new and remove existing Internet Bank users, their rights, and limits. Administrator rights enter into force when the user has been entered into the list of administrators in the agreement. Can be set only in the branch.
General rights of the company	Right to use company-related services, which are not account-based: information on leasing and factoring, additional services, and other services, which may be added in the future
Access to consolidated payment report	The right gives access to viewing the list of beneficiaries of consolidated payment. Consolidated order report can be viewed if in addition to the right to consolidated payment report, the user has been assigned also the rights to make predetermined payments and view outgoing payments.
Right to apply for the trade financing products	The right to submit to the Bank applications for trade financing products (guarantees, letters of credit, documentary collection).
Right to apply for loan disbursement	The right to apply for the disbursement of loans with unused loan limit. For payments made under the loan disbursement application, the Internet Bank limits are not considered.
Access to e-documents portal	User can see and download all digitally signed documents on behalf of the company and upload documents to be sent to the Bank. Management board members of the company and the business Internet Bank administrator always have access to digital documents, regardless of whether this right has been granted or not.
Access to the data on the legal entity	The right to view and prepare changes in the customer data questionnaire (Know Your Customer data sheet). The changes can be confirmed only by the management board member, regardless of whether this right has been granted or not.

Current account rights

Right to view account balance	User can see account balance
Right to view incoming payments	User can see credit transaction in statement
Right to view outgoing payments	User can see debit transactions in statement
Access to products and services	Right to view information on account-related banking services. User cannot make amendments to agreements with only access right to products and services.
Right to conclude basic agreements	Right to conclude basic agreements: ordering e-invoices and conclusion of e-invoice with automated standing order agreement; conclusion and amendment of debit card agreements, incl. blocking; deposit and current account agreement conclusion; standing order agreement conclusion; notification service agreement conclusion; conclusion of account-related agreements to be added in the future. Agreements can be amended, if the user has also been granted the right of “Access to products and services” for the same account.
Right to prepare and change payment orders	This user has the right to prepare and change payment orders from the account. Daily and monthly limits are not applied for this right.
Right to confirm payments	If the user is authorized to confirm payments from the account
Daily transfer limit (EUR)	
Monthly transfer limit (EUR)	
Signing weight for confirming payments	If company uses multiple signing of payments, the signing weights of payments for the users. The following signing weights can be used: 0%; 25%; 50%; 75%; 100%

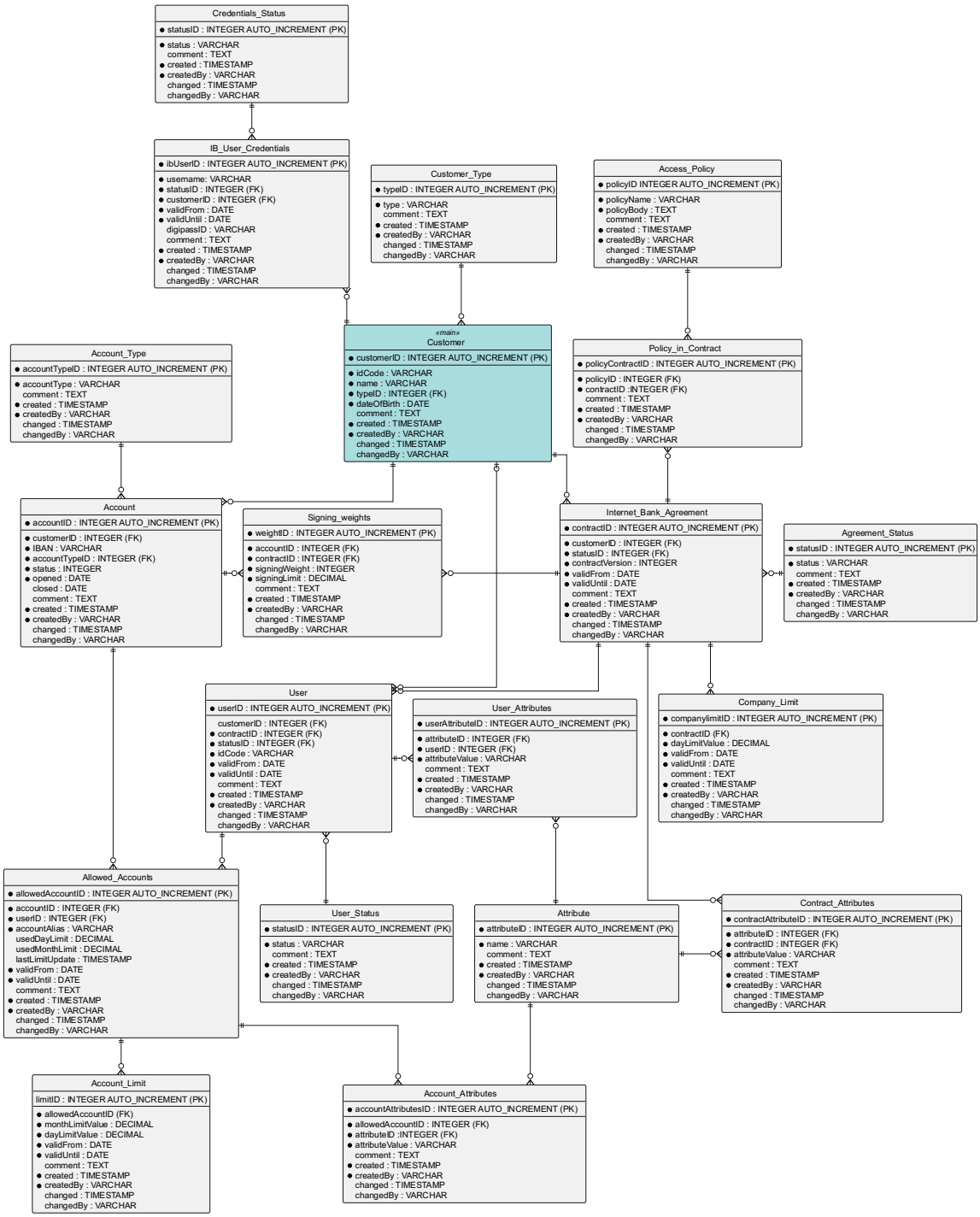
Securities account rights

Right to view information	Securities information is shown to user
Right to execute transactions	User can execute securities transactions

Rights related to accounts for forwarding e-invoices

Current account No	Account number of the e-invoice forwarding service
Administrator of the e-invoice forwarding program	Whether user is allowed to administrate e-invoice forwarding service

Appendix 8 – Detailed description of entities in the access control system



Detailed entity relationship diagram of the access rights model.

Table Customer

Field Name	Type	Mandatory	Comment
customerID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the customer record. Primary Key for table Customer.
idCode	VARCHAR	Yes	Identification code of the customer. Registry code for legal entity, national identification number for private customer
name	VARCHAR	Yes	Name of the customer
typeID	INTEGER (FK)	Yes	Reference to the customer's type. Foreign key to entity Customer_Type
dateOfBirth	DATE	Yes	Date of birth of the customer of date of first registration of the company.
created	TIMESTAMP	Yes	Timestamp when the record was created
createdBy	VARCHAR	Yes	User who created the record
changed	TIMESTAMP	No	Timestamp when the record was last updated
changedBy	VARCHAR	No	User who last updated the record

Table Customer_Type

Field Name	Type	Mandatory	Comment
typeID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the customer type. Primary Key for table Customer_Type
type	VARCHAR	Yes	Description of the customer type
comment	TEXT	No	Additional information about the customer type
created	TIMESTAMP	Yes	Timestamp when the type was created
createdBy	VARCHAR	Yes	User who created the type
changed	TIMESTAMP	No	Timestamp when the type was last updated
changedBy	VARCHAR	No	User who last updated the type

Table Agreement_Status

Field Name	Type	Mandatory	Comment
statusID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the status. Primary Key for table Agreement_Status
status	VARCHAR	Yes	Description of the agreement status
comment	TEXT	No	Additional comments on the status
created	TIMESTAMP	Yes	Timestamp when the status was created
createdBy	VARCHAR	Yes	User who created the status
changed	TIMESTAMP	No	Timestamp when the status was last updated
changedBy	VARCHAR	No	User who last updated the status

Table Internet_Bank_Agreement

Field Name	Type	Mandatory	Comment
contractID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the agreement. Primary Key for table Internet_Bank_Agreement
customerID	INTEGER (FK)	Yes	Link to the customer whom the agreement belongs to. Connects to table Customer
statusID	INTEGER (FK)	Yes	Status of the agreement, links to table Agreement_Status
contractVersion	INTEGER	Yes	Version number of the agreement. If new version of the contact is created, then the version number is incremented.
validFrom	DATE	Yes	Start date of the agreement
validUntil	DATE	Yes	End date of the agreement. If the contract does not have agreed end date, then the value of the record is 01-01-2100.
comment	TEXT	No	Additional comments on the agreement record
created	TIMESTAMP	Yes	Timestamp when the agreement was created
createdBy	VARCHAR	Yes	User who created the agreement
changed	TIMESTAMP	No	Timestamp when the agreement was last updated
changedBy	VARCHAR	No	User who last updated the agreement

Table User

Field Name	Type	Mandatory	Comment
userID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the user Primary Key for table User
customerID	INTEGER (FK)	No	Link to the associated customer in the table Customer. Field can be left empty in case there is not yet customer record created at the time when user is added to the contract.
contractID	INTEGER (FK)	Yes	Link to the table Internet_Bank_Agreement
statusID	INTEGER (FK)	Yes	Status of the user, links to table User_Status
idCode	VARCHAR	Yes	National Identification Code for the user. This is used to find corresponding customer in the bank’s customer list.
validFrom	DATE	Yes	Start date of the user's validity
validUntil	DATE	Yes	End date of the user's validity
comment	TEXT	No	Additional comments on the record
created	TIMESTAMP	Yes	Timestamp when the user was created
createdBy	VARCHAR	Yes	User who created the user record
changed	TIMESTAMP	No	Timestamp when the user record was last updated
changedBy	VARCHAR	No	User who last updated the user record

Table User_Status

Field Name	Type	Mandatory	Comment
statusID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the status. Primary Key for table User_Status
status	VARCHAR	Yes	Description of the user status
comment	TEXT	No	Additional comments on the status record
created	TIMESTAMP	Yes	Timestamp when the status was created
createdBy	VARCHAR	Yes	User who created the status
changed	TIMESTAMP	No	Timestamp when the status was last updated
changedBy	VARCHAR	No	User who last updated the status

Table Credentials_Status

Field Name	Type	Mandatory	Comment
statusID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the status. Primary Key for table Credentials_Status
status	VARCHAR	Yes	Description of the credential's status
comment	TEXT	No	Additional comments on the status
created	TIMESTAMP	Yes	Timestamp when the status was created
createdBy	VARCHAR	Yes	User who created the status
changed	TIMESTAMP	No	Timestamp when the status was last updated
changedBy	VARCHAR	No	User who last updated the status

Table IB_User_Credentials

Field Name	Type	Mandatory	Comment
ibUserID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the credentials. Primary Key for table IB_User_Credentials
username	VARCHAR	Yes	Username for internet banking access
statusID	INTEGER (FK)	Yes	References the status of the credentials
customerID	INTEGER (FK)	Yes	Link to the associated customer in the table Customer
validFrom	DATE	Yes	Start date of the credential's validity
validUntil	DATE	Yes	End date of the credential's validity
digipassID	VARCHAR	No	Number of the Digipass device issued to the customer
comment	TEXT	No	Additional comments on the credentials record
created	TIMESTAMP	Yes	Timestamp when the credentials were created
createdBy	VARCHAR	Yes	User who created the credentials
changed	TIMESTAMP	No	Timestamp when the credentials were last updated
changedBy	VARCHAR	No	User who last updated the credentials

Table Account

Field Name	Type	Mandatory	Comment
accountID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the account. Primary Key for table Account
customerID	INTEGER (FK)	Yes	Link to the account owner in table Customer
IBAN	VARCHAR	Yes	International Bank Account Number
accountTypeID	INTEGER (FK)	Yes	References the type of the account
status	INTEGER	Yes	Status of the account
opened	DATE	Yes	Date when the account was opened
closed	DATE	No	Date when the account was closed, if applicable
comment	TEXT	No	Additional comments on the account record
created	TIMESTAMP	Yes	Timestamp when the account was created
createdBy	VARCHAR	Yes	User who created the account
changed	TIMESTAMP	No	Timestamp when the account was last updated
changedBy	VARCHAR	No	User who last updated the account

Table Account_Type

Field Name	Type	Mandatory	Comment
accountTypeID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the account type. Primary Key for table Account_Type
accountType	VARCHAR	Yes	Description of the account type
comment	TEXT	No	Additional information about the account type record
created	TIMESTAMP	Yes	Timestamp when the account type was created
createdBy	VARCHAR	Yes	User who created the account type
changed	TIMESTAMP	No	Timestamp when the account type was last updated
changedBy	VARCHAR	No	User who last updated the account type

Table Allowed_Accounts

Field Name	Type	Mandatory	Comment
allowedAccountID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the allowed account link. Primary Key for the table.
accountID	INTEGER (FK)	Yes	Link to the associated account
userID	INTEGER (FK)	Yes	Link to the associated user
accountAlias	VARCHAR	Yes	Alias name for the account
usedDayLimit	DECIMAL	No	Daily limit usage
usedMonthLimit	DECIMAL	No	Monthly limit usage
lastLimitUpdate	TIMESTAMP	No	Timestamp of the last limit update
validFrom	DATE	Yes	Start date of allowance
validUntil	DATE	Yes	End date of allowance
comment	TEXT	No	Additional comments on the allowance
created	TIMESTAMP	Yes	Timestamp when the link was created
createdBy	VARCHAR	Yes	User who created the link
changed	TIMESTAMP	No	Timestamp when the link was last updated
changedBy	VARCHAR	No	User who last updated the link

Table Account_Attributes

Field Name	Type	Mandatory	Comment
accountAttributesID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the account attribute. Primary Key for the table.
allowedAccountID	INTEGER (FK)	Yes	Link to the account in the Allowed_Accounts table.
attributeID	INTEGER (FK)	Yes	Link to the attribute name
attributeValue	VARCHAR	Yes	Value of the attribute
comment	TEXT	No	Additional comments on the attribute record
created	TIMESTAMP	Yes	Timestamp when the attribute was created
createdBy	VARCHAR	Yes	User who created the attribute
changed	TIMESTAMP	No	Timestamp when the attribute was last updated
changedBy	VARCHAR	No	User who last updated the attribute

Table Attribute

Field Name	Type	Mandatory	Comment
attributeID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the attribute. Primary Key for the table.
name	VARCHAR	Yes	Name of the attribute
comment	TEXT	No	Additional comments on the attribute
created	TIMESTAMP	Yes	Timestamp when the attribute was created
createdBy	VARCHAR	Yes	User who created the attribute
changed	TIMESTAMP	No	Timestamp when the attribute was last updated
changedBy	VARCHAR	No	User who last updated the attribute

Table Account_Limit

Field Name	Type	Mandatory	Comment
limitID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the account limit. Primary Key for the table.
allowedAccountID	INTEGER (FK)	Yes	Link to the account in the Allowed_Accounts table
monthLimitValue	DECIMAL	Yes	Monthly limit value
dayLimitValue	DECIMAL	Yes	Daily limit value
validFrom	DATE	Yes	Start date of the limit validity
validUntil	DATE	Yes	End date of the limit validity
comment	TEXT	No	Additional comments on the limit
created	TIMESTAMP	Yes	Timestamp when the limit was created
createdBy	VARCHAR	Yes	User who created the limit
changed	TIMESTAMP	No	Timestamp when the limit was last updated
changedBy	VARCHAR	No	User who last updated the limit
changed	TIMESTAMP	No	Timestamp when the attribute was last updated
changedBy	VARCHAR	No	User who last updated the attribute

Table Access_Policy

Field Name	Type	Mandatory	Comment
policyID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the policy. Primary Key for the table.
policyName	VARCHAR	Yes	Name of the policy
policyBody	TEXT	Yes	Policy code in the agreed policy language format – Rego, Cedar or similar
comment	TEXT	No	Additional comments on the policy
created	TIMESTAMP	Yes	Timestamp when the policy was created
createdBy	VARCHAR	Yes	User who created the policy
changed	TIMESTAMP	No	Timestamp when the policy was last updated
changedBy	VARCHAR	No	User who last updated the policy

Table User_Attributes

Field Name	Type	Mandatory	Comment
userAttributeID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the user attribute. Primary Key for the table.
attributeID	INTEGER (FK)	Yes	Link to the attribute definition
userID	INTEGER (FK)	Yes	Link to the user
attributeValue	VARCHAR	Yes	Value of the attribute
comment	TEXT	No	Additional comments on the attribute
created	TIMESTAMP	Yes	Timestamp when the attribute was created
createdBy	VARCHAR	Yes	User who created the attribute

Table Contract_Attributes

Field Name	Type	Mandatory	Comment
contractAttributeID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the user attribute. Primary Key for the table.
attributeID	INTEGER (FK)	Yes	Link to the attribute definition
contractID	INTEGER (FK)	Yes	Link to the Internet Bank agreement
attributeValue	VARCHAR	Yes	Value of the attribute
comment	TEXT	No	Additional comments on the attribute
created	TIMESTAMP	Yes	Timestamp when the attribute was created
createdBy	VARCHAR	Yes	User who created the attribute

Table Policy_in_Contract

Field Name	Type	Mandatory	Comment
policyContractID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the policy assignment in a contract. Primary Key for the table.
policyID	INTEGER (FK)	Yes	Link to the policy definition
contractID	INTEGER (FK)	Yes	Link to the internet bank contract
comment	TEXT	No	Additional comments on the policy assignment record
created	TIMESTAMP	Yes	Timestamp when the policy assignment was created
createdBy	VARCHAR	Yes	User who created the policy assignment
changed	TIMESTAMP	No	Timestamp when the policy assignment was last updated
changedBy	VARCHAR	No	User who last updated the policy assignment

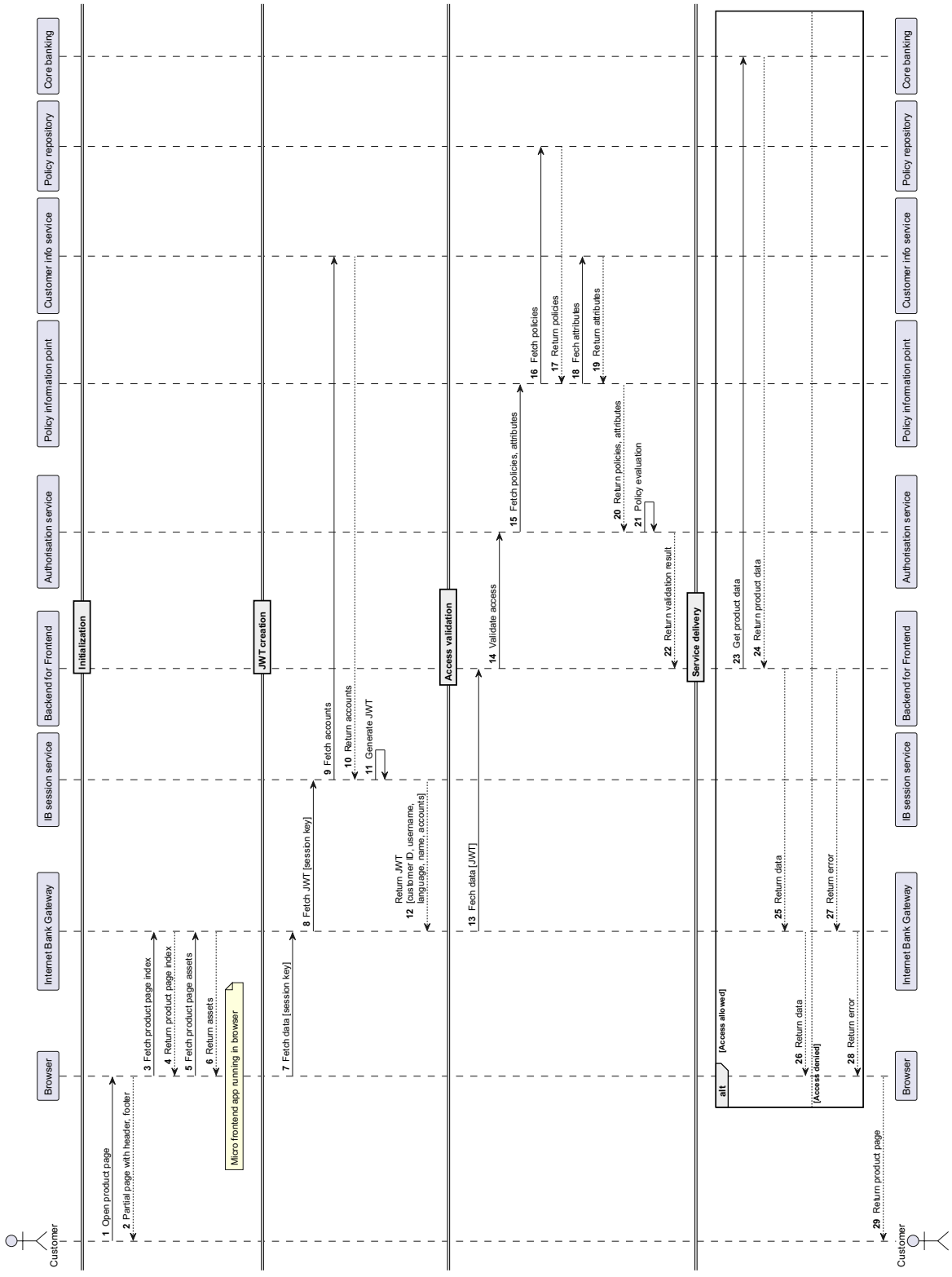
Table Company_Limit

Field Name	Type	Mandatory	Comment
companylimitID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the company limit. Primary Key for the table.
contractID	INTEGER (FK)	Yes	Link to the agreement on the Internet_Bank_Agreement table
dayLimitValue	DECIMAL	Yes	Daily limit value
validFrom	DATE	Yes	Start date of the limit validity
validUntil	DATE	Yes	End date of the limit validity
comment	TEXT	No	Additional comments on the limit
created	TIMESTAMP	Yes	Timestamp when the limit was created
createdBy	VARCHAR	Yes	User who created the limit
changed	TIMESTAMP	No	Timestamp when the limit was last updated
changedBy	VARCHAR	No	User who last updated the limit
changed	TIMESTAMP	No	Timestamp when the attribute was last updated
changedBy	VARCHAR	No	User who last updated the attribute

Table Signing_weights

Field Name	Type	Mandatory	Comment
weightID	INTEGER AUTO_INCREMENT (PK)	Yes	Unique identifier for the record. Primary Key for the table.
accountID	INTEGER (FK)	Yes	Link to the associated account
contractID	INTEGER (FK)	Yes	Link to the internet bank contract
signingWeight	INTEGER	Yes	Required signing weight for the multiple confirmation setup. Can be value from 0 to 100
signingLimit	DECIMAL	Yes	Required signing limit for the multiple confirmation setup. Specifies from which amount the required weight is required for payment confirmation.
comment	TEXT	No	Additional comments on the policy assignment record
created	TIMESTAMP	Yes	Timestamp when the policy assignment was created
createdBy	VARCHAR	Yes	User who created the policy assignment
changed	TIMESTAMP	No	Timestamp when the policy assignment was last updated
changedBy	VARCHAR	No	User who last updated the policy assignment

Appendix 9 – Full flow for the to-be authorisation sequence.



To-be sequence diagram. Full flow.