Design and Evaluation of a Proof-of-Concept for an Electronic Personnel Security Clearance (e-PSP) on the Swiss e-ID Infrastructure

University of Turku
Department of Computing

Master of Science (Tech) Thesis
May 2026

Mátyás Szikra

Supervisors:
Dr. Kaitai Liang
Petri Sainio

UNIVERSITY OF TURKU
Department of Computing

Mátyás Szikra: Design and Evaluation of a Proof-of-Concept for an Electronic Personnel Security Clearance (e-PSP) on the Swiss e-ID Infrastructure

Master of Science (Tech) Thesis, 88 p., 1 app. p.
May 2026

Personnel Security Clearances (PSPs) are high-trust credentials vital to Swiss national security. Despite the comprehensive background checks and regulations they are based on, their real-life verification process is often lacking or vulnerable. Current practices partially rely on ad-hoc information exchange, the manual checks of physical notification letters, or in extreme cases, simply asking subjects about their clearance status. The emergence of SWIYU, the Swiss national trust infrastructure, presents a timely opportunity to modernize this imperfect system by transitioning to verifiable digital credentials (e-PSPs) built upon hybrid Self-Sovereign Identity (SSI) principles.

Using a design-science methodology, this thesis designs, implements, and evaluates a complete proof-of-concept (PoC) for such a solution. To enforce data minimization in verifiable credentials, the research introduces a reusable three-layer model isolating identity, authorization, and lifecycle attributes, which guides the design of e-PSPs. Furthermore, recognizing that the ultimate security of SWIYU-based systems hinges on the application layer of the verifier to a much greater extent than one might assume, this component of the PoC is presented as a security-hardened reference implementation. This Backend-for-Frontend (BFF) architecture advocates for strict server-side enforcement policies, a formally defined state machine, and session isolation mechanisms.

The evaluation of the completed artifact and the analysis of the trust infrastructure validate that SWIYU's hybrid approach between SSI-inspired decentralization and centralized, government-backed trust guarantees can serve as a solid technical foundation for high-trust credentials. The developed prototype can execute full credential lifecycles, covering issuance, revocation, and verification. The development and testing of the PoC also served as an active battle-test for SWIYU's public beta, not only by generating actionable bug reports but by also being the first project to try multiple security-critical features in production.

In spite of the above, a broader socio-technical assessment reveals that Switzerland is not immediately ready for the universal adoption of SWIYU-based e-PSPs. Resolving open legal and operational questions, clarifying organizational responsibilities, and lowering public skepticism towards the technology are important prerequisites. Nevertheless, these current blockers do not invalidate the general idea. This research provides the analytical and engineering foundations required once Switzerland is ready to transfer its high-security credentials to a new era.

Keywords: Self-Sovereign Identity (SSI) – SWIYU – e-PSP – Digital Credentials – Cybersecurity Infrastructure

Acknowledgements

The present thesis was created during and is predominantly based on the work I did for my internship at the Swiss Cyber Defence Campus in Zürich. Therefore, I would like to thank the entire CYD Campus team and especially Dr. Martin Burkhart for the opportunity to work on something this relevant and interesting. Martin's insights and expertise on self-sovereign identity and the SWIYU trust infrastructure were above all key both to the development of the prototype and to the preparation of the thesis.

I would also like to thank Nico Grandjean of ASTAB, who is a dedicated advocate for innovation and digitalization within the Swiss Armed Forces. Without his efforts, this project would not have been possible at all, and he also played a key part in enabling us to receive crucial feedback from all the different stakeholders on the prototype.

I am similarly thankful to all my academic supervisors: Petri Sainio and Dr. Kaitai Liang of the University of Turku and Dr. Viktória Ildikó Villányi of Eötvös Loránd University for their support and the feedback they gave on the thesis draft. This was a much-needed final piece of the puzzle.

And last but not least, I would like to thank the European Institute of Innovation and Technology and the entire team of the EIT Digital Master School and SPECTRO for creating such an unparalleled degree program to begin with. Their work enabled me to spend my master's studies in not less than five European countries, with

potentially a sixth one upcoming for the graduation, and their support was an important motivation for always pushing for the next level.

Contents

List of Figures

List of abbreviations

AHV Alters- und Hinterlassenenversicherung (Old-Age and Survivors' Insurance)

API Application Programming Interface

BFF Backend-for-Frontend

CYD Cyber-Defence (as in CYD Campus)

DB Database

DCQL Digital Credentials Query Language

DID Decentralized Identifier

ECDSA Elliptic Curve Digital Signature Algorithm

e-ID Electronic Identity Document / Identification

e-PSP Electronic Personnel Security Clearance

ES256 ECDSA using P-256 and SHA-256 (Elliptic Curve Digital Signature Algorithm)

ESP Erweiterte Personensicherheitsprüfung (Extended Security Clearance)

Fedpol Bundesamt für Polizei (Federal Office of Police)

FOITT Federal Office for Information Technology, Systems and Telecommunication

FOJ Federal Office of Justice

GSP Grundsicherheitsprüfung (Basic Security Clearance)

HR Human Resources

HTTPS Hypertext Transfer Protocol Secure

ID Identity Document / Identification

iOS iPhone Operating System

ISG Informationssicherheitsgesetz (Federal Act on Information Security)

IT Information Technology

JSON JavaScript Object Notation

JWE JSON Web Encryption

JWT JSON Web Token

KB-JWT Key-Bound JSON Web Token

MVP Minimum Viable Product

OAuth2 Open Authorization 2.0

OID4VCI / OpenID4VCI OpenID for Verifiable Credential Issuance

OID4VP / OpenID4VP OpenID for Verifiable Presentations

OPM Office of Personnel Management (United States)

OSINT Open-Source Intelligence

PoC Proof-of-Concept

PSP Personensicherheitsprüfung (Personnel Security Clearance)

PWA Progressive Web App

QR Quick Response (code)

SD-JWT Selective Disclosure JSON Web Token

SEPOS Staatssekretariat für Sicherheitspolitik (State Secretariat for Security Policy)

SHA-256 Secure Hash Algorithm 256-bit

SSI Self-Sovereign Identity

SWIYU Swiss National Trust and e-ID Infrastructure (branded term)

tdw Trust DID Web

TLS Transport Layer Security

TSL Token Status List

UI User Interface

URL Uniform Resource Locator

U.S. United States

UX User Experience

VC Verifiable Credential

vct Verifiable Credential Type

VPSP Verordnung über die Personensicherheitsprüfungen (Ordinance on Personnel Security Screenings)

W3C World Wide Web Consortium

webvh Web Verifiable History

1 Introduction

1.1 Background and motivation

The Swiss government is currently developing a national trust infrastructure called SWIYU [1], which is expected to go live in late 2026 as the underlying ecosystem of Swiss e-IDs [2]. SWIYU's design is strongly inspired by self-sovereign identity (SSI) principles, an emerging approach aiming to make centralized identity providers obsolete by promising individuals secure and privacy-preserving control over how they store and share their own digital credentials [3].

Within this evolving environment of digital identities, Personnel Security Clearances (PSPs) represent a specific and critical domain. These clearances are used by and trusted at all levels of the Swiss administration and armed forces, and they act as important gatekeepers, controlling access to the country's essential facilities, classified information, high-security IT systems, and critical infrastructure [4], [5].

Despite being a cornerstone of Swiss national security, the practical handling and verification of PSPs often suffer from operational-level vulnerabilities. While internal clearance systems and databases do exist, these are not generally accessible tools, and especially interorganizational verifications in practice are often dominated by ad-hoc e-mail exchanges, subjects verbally confirming their own clearance status, or the acceptance of paper-based notification documents which are not meant to be used as official PSP credentials.

This contrast between the importance of PSPs and the operational realities makes it clear that modernization efforts are necessary, which is also argued by internal audits, even if from a slightly different perspective [6]. The emergence of the SWIYU trust infrastructure provides an ideal opportunity for re-thinking current processes. Integrating PSPs into this ecosystem has the potential to fundamentally improve clearance verifications by basing them on tamper-proof, electronic verifiable credentials (e-PSPs) [7] instead of partially ad-hoc and paper-based practices.

Besides improving security and efficiency, the prototype of an e-PSP system can also act as an important test environment. It can help answer the question of whether SSI-like frameworks, built mostly on the ideas of user control and decentralization, can successfully be adapted to support the strict security and usability requirements of high-trust, inherently authoritative clearance credentials.

1.2 Problem statement and research gap

While SWIYU presents an opportunity to improve current practices, implementing such a system is not straightforward. From an engineering perspective, the challenge is designing a tool that solves the above issues in a secure and efficient way. However, this, on its own, is not sufficient. Rather than just proposing a new tech stack, it needs to be assessed from various perspectives whether Switzerland is ready for a completely innovative approach in a security-critical area, and if it is, how the solution can replace or co-exist with legacy systems. Therefore, the focus is not only on whether SWIYU-based e-PSPs are implementable and secure, but also on legal, societal, and operational concerns.

Besides the above, the implementation of the PoC also shed light on a practical research gap. As both a handful of technologies it is built on and SWIYU itself are brand new [7], several security-relevant features of the infrastructure had never been tested in a production environment before this work.

Another, quite specific gap was identified concerning the security of the application layer of future verifier implementations. Though the ecosystem provides strong cryptographic guarantees regarding the validity, integrity, and privacy of credentials [7], strong cryptography, on its own, is not sufficient to guarantee the correct outcome of verifications [8], and the ultimate decision on acceptance or rejection is always going to be made by upper layers. This is important because SWIYU's design can make it tempting to view a verifier's application layer as a simple interface, while also leaving this critical responsibility to it. Since their role is so critical, rigorous security measures and a robust design are needed for verifier implementations to keep SWIYU-based solutions secure. With a structured approach and a reference architecture, this research aims to give first-of-its-kind recommendations.

1.3 Research goal and research questions

The primary goal of this thesis is the design, implementation, and analysis of a proof-of-concept solution for an electronic Personnel Security Clearance system. By designing the credentials and developing a functional prototype with issuance, revocation, and verification capabilities, the work can assess whether SWIYU is capable of securely and effectively supporting the complete lifecycle of state-level clearances in a practical setting. It also aims to address the verification layer gap by formalizing responsibilities and presenting a security-hardened reference implementation. From a higher-level perspective, the results are expected to provide insights into the applicability of SSI-like frameworks in general for high-trust use cases.

To address these goals systematically, the thesis aims to answer the following research questions:

1. How suitable is the SWIYU trust infrastructure to host an electronic Personnel Security Clearance system?

2. What technical design, architecture, and implementation practices enable the secure issuance, lifecycle management, and verification of credentials, with a particular focus on the verifier's application-level responsibilities?

3. How viable is the completed prototype from a technological perspective and what does stakeholder feedback indicate about its operational and organizational potential?

4. Beyond technological and operational feasibility, how ready is Switzerland to host security-critical credentials on an SSI-like infrastructure from a broader legal and societal perspective?

1.4 Scope and delimitations

In order to keep the research focused and feasible, this thesis defines certain boundaries and a concrete scope for the work.

First, though the legal framework in Switzerland enables multiple possible outcomes for personnel security screenings, including clearances with certain reservations and the opposite of clearances, so-called "risk declarations" [5], [9], this proof-of-concept only models the positive outcome. These positive "security declarations" are what this work refers to as e-PSP credentials. Modeling negative or conditional clearances in an SSI-like framework raises both legal and operational questions, including how likely it is that an individual willingly stores and presents an official unfavorable recommendation of themselves.

Second, while the issuance and revocation flows of the prototype are strictly treated on a proof-of-concept level, with the primary goal of demonstrating that implementing these is feasible within SWIYU, verification is given special attention. The verifier component of the PoC is presented as a security-hardened reference implementation, which also serves as a core technical contribution of the thesis.

Third, the final scope of the work has slightly changed when compared strictly to the original topic proposal. All of these changes come from the agile nature of the research, and they are all based on either the current maturity state of SWIYU and the respective limitations, or the realization that certain areas deserve more attention than others. Since even at the time of writing, both the e-PSP prototype and the trust infrastructure are actively being developed further, guaranteeing a "scope freeze" is not realistic. Overall, the changes are minimal, and the focus stays on creating a SWIYU-based proof-of-concept and analyzing the solution.

1.5 Methodology and thesis approach

In order to address the research questions systematically, this thesis applies a design-science methodology [10]. This approach is a natural fit for this use case, since its goal is building and assessing an artifact aiming to solve a real-world problem. The concrete research and development process is structured into the following phases:

1. Problem identification and technical background research: The work starts with information gathering on current PSP-related practices and the systematic analysis of SWIYU, its APIs, underlying technologies, issuance, and verification flows. This phase provides the technological basis for the work as well as the specific operational vulnerabilities of current processes that the thesis aims to address.

2. Requirements gathering: The next step consists of collecting additional direct input from stakeholders and employees of the Swiss administration, including operational and legal requirements for a SWIYU-based e-PSP system. Then the technical requirements for the prototype are also formalized.

3. Artifact design and implementation: Based on the collected information, the core engineering artifacts are designed and implemented. The output includes

the whole functional proof-of-concept, conceptual modeling and design recommendations, the general, security-hardened SWIYU reference verifier, and a live deployment of the system for demonstrative purposes.

4. Evaluation and analysis: The implemented system is analyzed along different dimensions, including reflections on technical feasibility, the maturity of SWIYU and its ability to host high-trust credentials, and Switzerland's societal and legal readiness for such a solution from a broader perspective. The evaluation is based on a mixture of formal analysis of components, the impressions made while actively battle-testing the trust infrastructure through the development phases, feedback from stakeholders, and observations on the general attitude towards such solutions.

1.6 Thesis structure

The remaining parts of this work are structured in the following way.

Chapter 2 provides the necessary context by introducing what PSPs are, their legal framework, their security implications for Switzerland, and their exact roles as critical, gatekeeping credentials. Building on this, Chapter 3 analyzes current verification practices, existing trust assumptions, and presents the identified operative vulnerabilities and limitations.

Chapter 4 examines the base principles of SSI and introduces the SWIYU trust infrastructure, including its architecture, components, and cryptographic protocols. Chapter 5 then narrows the gap between theory and technical implementation by formalizing the requirements for an e-PSP system.

The discussion of the practical design and implementation of the proof-of-concept is split into two separate parts. Chapter 6 focuses on the modeling and design of e-PSP credentials and on the issuance and revocation features of the prototype.

Chapter 7 introduces the verifier of the PoC more deeply, positioning its application layer as a security-critical component, and presenting a security-hardened reference architecture.

After the development phase, Chapter 8 assesses the created artifact, confirming its technical feasibility, and collecting the takeaways of this real-life battle test of SWIYU as well as stakeholder feedback. Chapter 9 then widens the focus again, discussing Switzerland's general, legislative, and societal readiness for such a solution, while also briefly touching upon the international landscape of similar projects. After that, Chapter 10 closes the thesis by directly answering the research questions, summing up the main contributions, and outlining future implications.

2 Personnel Security Clearances in Switzerland

2.1 Purpose and security objectives of PSPs

Personnel Security Clearances, officially called "Personensicherheitsprüfungen" (PSPs), are preventive vetting mechanisms in Switzerland, aiming to protect the country's national security. The main goal of a PSP is to proactively assess whether an individual can become a security risk if put in a sensitive role or given access to critical facilities or information, and to thereby eliminate and mitigate potential insider threats [4]. These threats can be diverse, including espionage, sabotage, the unlawful disclosure of classified information, or any purposeful or negligent act that endangers Switzerland's general safety and stability.

To achieve these objectives, obtaining a PSP requires going through a comprehensive screening process. While the screening itself can have multiple outcomes, the positive one results in the issuance of a positive security declaration (Sicherheitserklärung).[1] This statement confirms that no substantial risk was identified through

[1]It is important to note that strictly speaking, a Personensicherheitsprüfung (PSP) refers to the screening process itself, not the resulting clearance credential. The official name of the positive outcome of the PSP process is the security declaration. Despite that, this thesis uses the terms PSP, security declaration, and security clearance interchangeably for the outcome, while PSP can still refer to the process itself too. The reason for that is that in everyday jargon, people almost always mean successfully having passed the screening process by having a PSP. While we try to stick to the more precise terminology when describing the legal background and official processes, overall the everyday jargon is prioritized. This is already made obvious in the title of the thesis,

the vetting process, and acts as a formal basis of trust required for security-critical access, roles, or responsibilities [9].

2.2 Legal framework

Switzerland's personnel security clearance system is based on a comprehensive legal framework, consisting mainly of the Federal Act on Information Security (Informationssicherheitsgesetz, ISG) and its implementing ordinances [5], [11]. The ISG in general regulates the protection of federal information and critical IT systems, which also includes the legal requirements of PSP screenings. The act formally defines the role of these vetting procedures as evaluating potential risks to information security originating from persons placed in sensitive roles, and explicitly mandates federal authorities to define which exact positions count as such. It also specifies data protection provisions and the rights of the subjects during the vetting procedure.

To operationalize the high-level directives of the ISG, the Swiss Federal Council enacted the Ordinance on Personnel Security Screenings (VPSP). The VPSP sets out PSP procedures in detail, regulates their practical implementation, and assigns concrete responsibilities to authorities. It also orders the creation of the specialized PSP units responsible for executing the screenings.

While the ISG and VPSP focus primarily on information security and roles with access to sensitive or classified information, Swiss law also provides the legal basis for vetting personnel in other critical domains. Legislation such as the Federal Personnel Act, the Military Act, the Asylum Act, and the Electricity Supply Act permits the screening of individuals who, in their different roles, could harm the fundamental interests of the state even without such access. Importantly, these provisions are all legally harmonized with the standard PSP procedure, enabling a unified, consistent vetting process across all federal and operational domains [12].

where the term e-PSP is meant as the digitalized version of a security declaration.

2.3 Scope and applicability

The obligation to undergo a personnel security screening applies to a wide range of governmental, defense, and private sector employees. The affected personnel can be categorized as follows [13]:

- Federal, cantonal, and municipal employees: All employees of the Swiss administration whose role was designated security-sensitive, generally those requiring access to federal confidential or secret information, or high-security IT systems. Certain cantonal or municipal roles with access can also require a PSP.

- Members of the Armed Forces: Beyond access to sensitive information, military personnel are also vetted if they require entry to certain restricted facilities or before promotion to certain ranks [14]. Unlike in civil cases, military background checks can even be performed without the prior consent of the subject.

- Third-party contractors: The scope of PSPs can extend to the private sector. Employees of certain contractors with sensitive access, such as in defense or IT, are also obligated to undergo a screening.

- Critical infrastructure personnel: The key staff of critical national infrastructure, such as employees of nuclear power plants or the national grid operator Swissgrid, are also vetted. These specific checks are also regulated by additional sector-specific laws.

2.4 PSP levels and screening procedure

Depending on the position, the required PSP is categorized into one of two separate levels: Basic Security Clearance (Grundsicherheitsprüfung, GSP) or Extended Per-

sonnel Security Clearance (erweiterte Personensicherheitsprüfung, ESP). Assigning a role to one of these levels is based on the severity of the potential damage that an untrustworthy or negligent individual could cause in it. A basic screening can suffice for positions where a breach or incident would do harm but not seriously endanger Swiss national interests, while an extended screening is strictly necessary for roles where that is the case [11], [15].

The clearance process itself follows a formal, well-defined procedure, consisting of several key phases and orchestrators [11], [16]. The main roles are those of the so-called initiating body (Einleitende Stelle), meaning the organization that initiates the screening, the deciding body (Entscheidende Stelle), which always makes a final decision based on the outcome of the checks, and the PSP body conducting the actual screening, which is either the specialized PSP unit within SEPOS or for certain cases a separate PSP unit of the Federal Chancellery (Bundeskanzlei) [4]. The process itself goes as follows:

1. Initiation: The process starts when an individual is assigned to or being actively considered for a certain sensitive role. In most cases, before the initiating body (often the future employer itself) can request the screening, the subject must give written consent to be checked. A notable exception is military personnel, for whom the law explicitly waives this right. Afterwards, the actual screening is performed by the competent PSP body, meaning, as mentioned before, either the PSP unit within SEPOS or in the Federal Chancellery. These can also collaborate with other agencies throughout the process [4], [15], [16].

2. Data collection and verification: In this phase, the PSP unit systematically collects all security-relevant information about the individual's background and lifestyle. A basic screening mainly relies on sources like criminal records, ongoing investigations, intelligence files, military security records, and basic information on financial stability. Open-source intelligence, such as social media

posts, may also be reviewed. An extended screening goes deeper, by contacting tax authorities, verifying residences, examining financial relationships, and conducting a mandatory interview with the candidate. In certain cases, interviews are also needed for basic clearances, such as when the collected data reveals suspicious gaps or inconsistencies [11], [16].

3. Analysis and risk assessment: After collecting all the above data, the PSP unit thoroughly analyzes it, looking for indicators of potential risks. Typical red flags can include serious criminal offenses, ties to extremist organizations, significant personal vulnerabilities, or any other signs of untrustworthiness. The severity and mitigability of the identified issues are then carefully assessed [16].

4. Outcomes and recommendations: As the outcome of the process, the PSP body issues one of four possible formal recommendations for the deciding body:

    - a Sicherheitserklärung (Security Declaration), indicating that no significant risk was identified. As declared in the scope, this positive outcome is what this thesis implements as an e-PSP credential.
    - a Sicherheitserklärung mit Vorbehalt (Security Declaration With Reservations), meaning that certain risks were identified, but they were deemed mitigatable.
    - a Risikoerklärung (Risk Declaration) formally declaring the subject a security risk, meaning that the PSP unit explicitly advises against employing the individual in sensitive roles.
    - or a Feststellungserklärung (Undetermined Status), when the investigation was unable to reach a clear conclusion, typically due to missing information.

5. Review and renewal: If the PSP unit intends to issue a negative recommendation, the subject has the right to appeal the decision. Positive recommendations are also not permanent, and must periodically be renewed. In the case of basic clearances, the validity period is 10 years, while extended clearances are valid for 5 years. An update may be required even sooner, for example when the subject shifts to a higher-sensitivity role [15], [16]. Recent cases of several high ranking officials failing PSP screenings also demonstrate how important these periodic checks are and how seriously the recommendations need to be taken [17].

2.5 PSPs as high-trust, high-risk credentials

PSPs, in practice, function as critical gatekeeper credentials. Cleared individuals often handle or are able to view governmental documents classified as confidential or secret, or have access to high-level security zones, for example, those housing critical infrastructure or intelligence agencies. They also often operate or use IT systems with the highest protection levels [16], and since a PSP may even be recognized as a formal declaration of trust by international partners, certain holders' access can extend to secrets and facilities of other nations or organizations [18].

Because of their critical preventive role in maintaining national security and stability, how PSPs are shared and verified in practice is a significant concern. While the legal framework regulating their use is robust and thorough, the operational realities, as mentioned in the introduction, do not always match this importance. In the upcoming chapter, this contrast is analyzed in more detail.

3 Current PSP Verification Practices and Limitations

3.1 Current verification practices

When a positive security declaration is issued, the cleared individual usually receives a confirmation in either paper or digital form. While this primarily serves as a formal written notice, the deciding body is also always notified, and the clearance status is recorded in internal systems [15], [19]. Within an organizational unit or command hierarchy, therefore, clearance status is known, and these internal databases allow designated security personnel, so-called information security officers, to verify employees' clearance statuses, for example, when the personnel of other organizations query them [5]. This designated personnel currently acts as the official channel for PSP checks.

However, in reality, these cross-organizational verifications often do not take place, or they do not take place correctly. This can stem from, for example, organizers of a sensitive meeting not wanting to go through the overhead of pre-contacting the responsible staff of every participant's organization to query their clearance status in advance, or simply from situations where this would not be realistic. Since existing internal systems are not designed to provide universally available, live verification services, in situations where pre-verification did not happen, actual verifi-

cations often fall back on ad-hoc and informal methods, such as meeting organizers simply asking participants if they are cleared or omitting verifications completely. Other prominent procedures include treating the paper-based notice of the clearance decision as the official clearance credential, and simply performing verification by inspecting that document.

Even when pre-verification happens, the practical implementation often also suffers from operational vulnerabilities. An example of this is the responsible staff exchanging fragmented emails, occasionally even sending a confirmation to the holder of the PSP only and trusting them to forward it to the verifier without making any changes in it.

And finally, even if a completely correct verification had been done in the past as a requirement for a classified project lasting years, it could easily be the case that the check was only conducted in the beginning and the subject's status has changed in the meantime.

While at the time of writing official data on what percentage of verifications is performed following the above practices is not available, the symptoms are clearly not occasional, and given the critical role of PSPs, even one imperfect verification is an operational risk. Conducting a survey to get a more exact number would certainly be an interesting task but it is out of scope for this research and is one of the recommendations for future work. For now, the above claims are based on non-representative conversations with several stakeholders within the Swiss administration.

3.2 Trust assumptions of current practices

Based on the observed shortcomings of current practices, we can identify the concrete trust assumptions that they inherently rely on. These are the following:

- Authenticity and integrity of the proof: For these practices to be secure, verifiers need to fundamentally trust that the presented proof of clearance is genuine and unaltered. In the case of relying on paper certificates, for example, authenticity can only be judged by visually inspecting letterheads, stamps, and signatures, assuming that forgeries would either be noticeable or too difficult for an adversary to make. When it comes to forwarded email confirmations, the verifiers must trust that these have not been changed by the subject or any intermediate party.

- Sufficient identity confirmation: In certain situations, the process also assumes that the verifier can and will correctly confirm the subject's identity. Since the physical PSP letters do not contain a photograph, they must manually and thoroughly be cross-checked with an identification document, and therefore trust is placed in the verifier's willingness and ability to do so.

- Awareness of clearance scope and validity: In the same scenario of paper-based verifications, it is also assumed that the verifier fully and correctly understands the clearance's scope and the validity period. This person must independently judge if the clearance level is appropriate in the specific context, and actively check for expiration based on a document, assuming that every guard or meeting organizer reliably spots invalid or out-of-scope clearances.

- Holder honesty: In extreme cases, where formal checks are completely omitted, the process relies either purely on the honesty of the subjects, assuming that none of them will attempt to lie about their clearance status, or on the transitive trust between employees, for example when somebody the verifier knows trusts the subject and has invited them to a meeting. However, even if verification is only omitted when the subject clearly works for a certain organization, they could easily be in a position not requiring a background check

or not be cleared for other reasons. This also leads to the next assumption.

- Compliance with recommendations: Since PSP outcomes are defined by law as recommendations and not decisive statements, the deciding body technically has the right to overrule them [4], [5]. The system therefore inherently trusts that decision-makers will act responsibly. If a deciding body overrules a risk declaration without a good reason, a person with a risky background could get into a trusted position, potentially without other organizations ever learning that they do not hold a clearance.

Formalizing the above trust assumptions makes it clear that with the observed shortcomings, the practical trust model of PSPs is heavily reliant not only on the alertness and competence of verifiers, but practically on the complete absence of malicious intent. The weaknesses and limitations of existing practices are further analyzed in the next section.

3.3 Weaknesses and limitations of current practices

Based on the above trust assumptions, the following weaknesses can be defined:

- Risk of forgery and misuse: The current operational reality is fundamentally vulnerable to the forgery or alteration of clearance proofs. A physical paper certificate could be faked convincingly enough by a sophisticated adversary to deceive an unsuspecting verifier, and ad-hoc practices based on email forwards are trivial opportunities for data manipulation. Beyond direct forgery, legitimate holders of lower level clearances could attempt to social engineer their way into higher-security contexts in both cases, for example by re-using email confirmations or relying on verifiers not properly checking the scope of a clearance document.

- Challenges in revocation: Another limitation is the difficulty of enforcing a clearance revocation across organizational boundaries. If a holder's clearance is revoked, the responsible authority can notify the direct employer and potentially confiscate the physical certificate, but due to the lack of a real-time, universally accessible revocation database, an external party could potentially remain uninformed of the invalidation, still treating the individual as cleared. A person secretly keeping a copy of their revoked certificate or forwarding an older email confirmation could also still attempt to prove having a clearance to new verifiers.

- High reliance on human alertness and honesty: Current practices inherently leave more responsibility to verifiers than they strictly need to have. A verifier checking a physical document might fail to properly cross-reference holder details with an ID, or miss that the clearance has expired. Verifications based on internal databases expect verifiers to make the extra effort of contacting responsible staff and querying the status through them every time a check is needed, while also ensuring that the right confirmation made its way to them in an unaltered way.

- High reliance on responsible personnel, and their role as a bottleneck: Since in the current system, up-to-date PSP status can only be accessed through designated security personnel, their role in the system is even more critical. Not only do they act as a bottleneck to important status information and thereby to the availability of the whole system, but one corrupted employee could potentially fake the outcome of a large number of PSP verifications.

3.4 Why modernization is justified

While to date, no catastrophic breaches exploiting PSP verification weaknesses appear to have been publicly reported, the above risks in the case of such security-critical credentials are concerning enough on their own to justify modernization.

It is also apparent that the majority of the above issues stem from a lack of a universally available, easy-to-use electronic verification service. Such a solution could significantly lower the administrative burden and responsibility of the verifiers, while also enforcing cryptographic authenticity and ensuring real-time status and revocation checks. It would also need to be designed in a way that while the system itself is accessible by every legitimate verifier, each of them can strictly only check for clearances that were meant to be shared with them.

Since a system built on verifiable credentials has the potential to offer exactly these advantages [20], this work's proposal of exploring whether currently emerging SSI-like infrastructures, and more concretely SWIYU, are a fit for PSP-like credentials, could not be more timely. The following chapter gives the necessary background to understand these frameworks.

4 Conceptual and Technical Background: SSI and the SWIYU Trust Infrastructure

4.1 SSI and its relevance for high-trust credentials

Self-Sovereign Identity (SSI) has recently emerged as the prominent paradigm of digital identity management. The general idea of SSI systems is replacing legacy identity providers with a decentralized so-called trust triangle composed of issuers, holders, and verifiers. In this decentralized model, users act as the sole guardian of their own identities and have full control over how they share their data. The credentials they hold are digitally signed claims stored locally on their own devices and presented only upon request [20], [21].

For high-trust, high-risk credentials like PSPs, the concept of SSI can definitely be appealing. The strong cryptography guarantees make credentials hard to forge or alter, and since these are stored locally by the holders themselves, the dependence on centralized databases and thereby the risk of major data breaches or single points of failure are heavily reduced [22]. These two are more than just theoretical concerns with national security clearances, as demonstrated by past incidents, such as the OPM hack in the USA [23]. SSI also offers decentralized verification capabilities by

design, which can be important when several different organizations need to be able to verify a certain credential without costly backend integrations.

At the same time, attempting to apply pure SSI principles for high-security, PSP-like use cases does not come without its own challenges. The most critical ones are the following [21], [22]:

- Secure key and wallet management: Since pure SSI makes holders the sole guardians of their credentials and the keys that secure them, while the single point of failure of a centralized database is eliminated, in a way, several others are created in its place. Since in a security clearance context even one stolen credential can be enough for a breach, this can get extremely dangerous. The designer of such a system would need to blindly trust that users will perfectly follow best security practices, without being able to do much to enforce them.

- The ownership of credentials as an information leak: When it comes to critical credentials like PSPs, even the sheer fact that a person holds that credential can be crucial and sensitive information, potentially making the owner an interesting target. This means that clearance data does not necessarily need to be stolen to cause issues. Even if all holders properly secure their keys and wallets, the decentralized nature of the system could still result in new ways that clearance statuses can leak. One example could be during border control, where foreign agents may be authorized to search personal devices.

- Issuer legitimacy and governance: Pure SSI does not formally restrict the possible pool of issuers, which puts the responsibility of judging an issuer's legitimacy entirely on holders and verifiers. In the case of high-trust credentials like PSPs, this is obviously unacceptable. In pure SSI nothing would theoretically stop an impostor from creating a credential falsely claiming to be an official clearance.

- The revocation dilemma: Since pure SSI generally opposes everything that is centralized, enforcing revocation is not a trivial problem. While distributed approaches like ledgers do exist, in the case of high-trust credentials, having constant access to up-to-date and reliable revocation data can be vital, possibly pushing designers towards hosting these in a somewhat centralized manner.

- Institutional oversight and power asymmetry: While the philosophy of pure SSI assumes that a holder is always free to decide what to disclose, high-trust use cases involve inherent power asymmetries that can make this unrealistic. A cleared individual may be legally obligated to present their credential to authorized verifiers, meaning that while SSI can ensure that the user technically mediates the process, that ultimate choice on what and when to share is not actually theirs. It is clear that security clearances by design remain somewhat authoritative credentials that likely require a certain level of institutional oversight, which can inherently conflict with pure SSI's ideology of complete decentralization.

While a pure SSI ecosystem may be unfit to host state-level security clearances, certain attributes of SSI remain highly desirable. To get the best of both worlds, a hybrid design can be a realistic solution. Such a system must retain the core advantages, such as tamper-proof credentials and decentralized local storage, while also mitigating the shortcomings, for example by relying on stricter issuer restrictions and additional institutional trust anchors.

The following sections are going to present how the SWIYU trust infrastructure follows exactly this kind of hybrid approach.

4.2 The SWIYU trust model and ecosystem roles

The SWIYU trust infrastructure implements the traditional SSI trust triangle but complements it with a fourth component of centralized, state-operated registries. The resulting roles of this ecosystem are described below [1].

- Holders: Potentially all citizens and residents of Switzerland. They interact with the ecosystem primarily through the official SWIYU app (or other compatible wallets), which enables them to receive, store and present their credentials. As per SSI principles, this is the only means of storage. No central credential database exists. Holders also have the technical capabilities for active control and selective disclosures, meaning they must not only give consent to share a credential, but approve the exact data fields to be shared [1], [7].

- Issuers: Any organization can be an issuer from government bodies to universities or private enterprises. Their primary function is to create, cryptographically sign, and offer verifiable credentials to holders. To ease joining SWIYU as an issuer, an open source Generic Issuer software is provided. This acts as an abstraction layer hiding the infrastructure's complexity, for example by simplifying the publication of cryptographic materials, the creation of credentials, or managing the lifecycle of these. [1], [24]

- Verifiers: Entities that request credentials (or specific claims within them) from holders for verification. Similar to the issuer side, SWIYU offers an open-source Generic Verifier to assist with certain tasks, such as the validation of cryptographic signatures or checking the status of credentials. Verifiers never directly communicate with issuers and instead rely on the centralized registries to query any required information. [1], [24]

- Registries: The aforementioned registries are operated by the Federal Office of Information Technology, Systems and Telecommunication (FOITT) of the Swiss Confederation. There are two of them, with well-defined and distinct roles. The Base Registry is responsible for establishing technical trust. It is open for anyone to onboard and acts as the public directory of the ecosystem. It hosts the technical identifiers (DIDs) of all issuers and verifiers as well as public cryptographic materials and the status lists needed for credential lifecycle management. The Trust Registry, on the other hand, is responsible for adding institutional trust to the ecosystem, directly addressing the issuer legitimacy problem of pure SSI. Onboarding here is more restricted. If an organization wants to be a recognized issuer or verifier, its real-world identity must first be formally confirmed by the Swiss Federal Office of Justice (FOJ). Once verified, the FOITT issues a special type of credential, called a Trust Statement, which is stored in the Trust Registry. This credential then cryptographically connects a DID to a verified, real-world institutional identity. [1], [25]

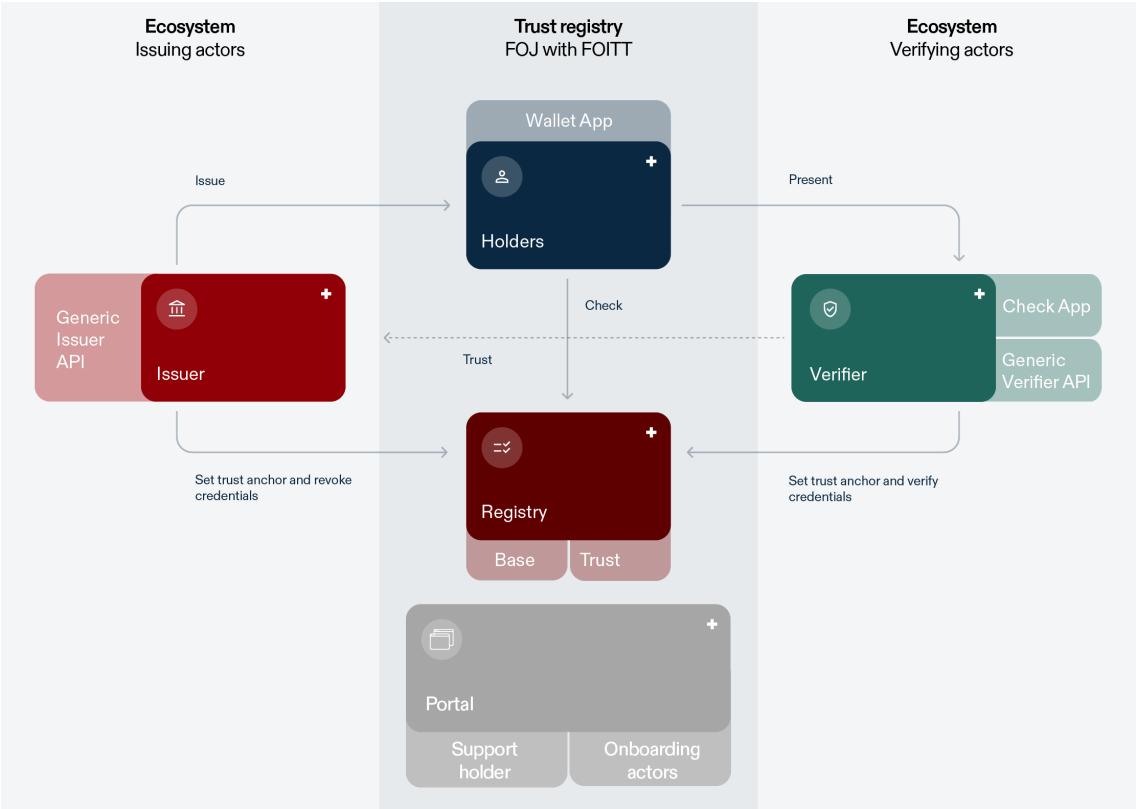

Figure 4.1: A high-level overview of the SWIYU trust infrastructure, illustrating the main ecosystem roles, including the central role of registries as trust anchors. Adapted from the SWIYU documentation.

4.3 Core SWIYU components and architecture

SWIYU provides a suite of software components, reference implementations, and helper tools. While the most important ones have already been mentioned in the last section, for the sake of completeness, the core elements of this infrastructure are the following [24]:

- SWIYU Wallet App: The official Swiss mobile application for holders. Its main functionalities are the above-mentioned storage and presentation of credentials in a secure and privacy-preserving manner [1], [26].

- Generic Issuer: The open-source reference implementation of core issuer tasks, also acting as an abstraction layer masking SWIYU's complexity. It is ready to be integrated into issuer implementations and guarantees full compatibility with the ecosystem out of the box. Services doing this already exist, an example is the Beta e-ID Issuer by the Federal Office of Police (Fedpol) [1], [24].

- Generic Verifier: The verification-side counterpart to the Generic Issuer, handling cryptographic validation and status checks. It can be integrated and customized the same way, allowing specific verifier implementations to be built on top of it [24].

- SWIYU Check App: While the Generic Verifier is designed to handle online, web-based verifications, the Check App is a mobile application being developed specifically for face-to-face and possibly offline use cases. Although initially it will only be able to be used for verifying e-IDs, its functionalities are planned to be extended to also support other credentials in the future [1].

- Base and Trust Registries: As established before, the Base Registry hosts technical information such as DIDs, DID documents, and status lists, while the Trust Registry stores FOJ-approved Trust Statements. The concrete registries are operated by the Swiss Federal Office of Information Technology, Systems and Telecommunication (FOITT) as API services [1], [25].

- DID Toolbox and DID Resolver: Supplementary utility tools provided to assist with generating key pairs and initial DID logs, and with resolving DIDs via the Base Registry. They are mostly useful during the infrastructure onboarding process [24], [27].

- Online Portals: Organizations can use these online web portals to register as

business partners and obtain the necessary API credentials required to write the Base Registry. They are also mainly used during onboarding [27].

Almost all of these components are open-source, allowing anyone to freely review or contribute to the codebase. Many of them are also optional to use, since SWIYU is designed to eventually allow third-party components, including alternative wallets, or proprietary verifier and issuer implementations. The one prerequisite is strict compliance with the infrastructure's protocols and specifications. In the following section, some of these are listed [1], [24], [28].

4.4 Credential formats, cryptographic primitives, and protocols

SWIYU relies on a robust stack of cryptographic primitives, standards, and protocols. While a deep presentation of these is beyond the scope of this work, the key ones are listed below [7], [28]:

- Credential formats and selective disclosure: SWIYU implements W3C Verifiable Credentials, specifically the SD-JWT VC (Selective Disclosure JSON Web Token) format. This standard enables selective disclosure by hashing each individual claim with a unique nonce. The resulting credential, which then includes these hashed values rather than any of the raw data, is digitally signed by the issuer. During verification, the holder simply sends the whole verifiable credential alongside only the specific plain-text values and corresponding salts that they wish to reveal. This design enables verifiers to cryptographically prove the integrity of a credential as a whole, while also learning nothing of the parts that are wished to be kept secret [7], [28].

- Cryptographic primitives and device binding: As a digital signature algorithm,

the infrastructure relies on ES256 (Elliptic Curve Digital Signature Algorithm on curve P-256), while hashing is performed using SHA-256. To ensure the confidentiality of the credentials in transit, all communication among components occurs strictly over HTTPS/TLS, and JSON Web Encryption (JWE) is about to be added as an extra layer of security before the go-live. Binding credentials to physical devices is also supported. This is possible by embedding a confirmation key within the credential, relying on the standard of Key-Bound JWTs. The future support for key attestations is going to make this mechanism even more secure by offering the possibility to ensure the use of hardware-backed keys, thereby making it significantly harder to replay or steal a credential [2], [7], [28].

- Issuance and presentation protocols: To orchestrate credential flows between components, SWIYU utilizes specialized OpenID protocols. Issuance is handled via OpenID4VCI (OpenID for Verifiable Credential Issuance), and verifiers request credentials using OpenID4VP (OpenID for Verifiable Presentations). The idea behind these two is taking something that has heavily been used and battle-tested in the last years for authentication flows and re-thinking them for the use case of SSI [7], [28].

While all of these building blocks are interesting topics on their own, deep comprehension of them is not required to understand how the ecosystem works. To maintain a focused narrative, this chapter goes on with the further exploration of SWIYU's practical working.

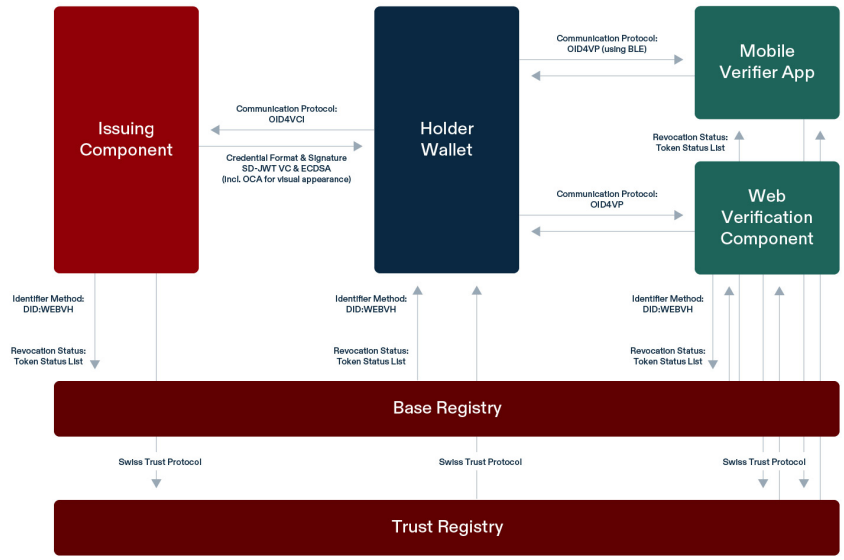

Figure 4.2: A high-level overview of SWIYU's most important formats and protocols in relation to the infrastructure's components. Adapted from the SWIYU documentation.

4.5 Onboarding to registries and credential lifecycle management

To participate in the ecosystem as an issuer or verifier, an organization must first onboard into the Base Registry. This process begins by registering as a formal business partner through the ePortal and acquiring the necessary API credentials. The organization then usually creates its Decentralized Identifier (DID) and generates the necessary cryptographic key pairs, unless these are already available. SWIYU relies on the did:tdw method (which has been renamed to "did:webvh" in newer versions). This method requires the generation of an initial DID log, which is a historical chain of all the versions of the public DID Document with one entry in

it as a starter. The organization then uploads the initial log to the Base Registry, thereby officially publishing its DID Document. The majority of this process can be done with the help of the DID Toolbox [27], [28].

If the organization wants to act as a recognized issuer or verifier, it must complete a secondary onboarding step into the Trust Registry. This currently requires submitting a request that includes the DID and contact details. Once the FOJ has successfully confirmed the real-world identity, the FOITT publishes the Trust Statement in the Trust Registry, linking the DID to the verified institution [25], [27].

Another important topic is how SWIYU's design enables credential lifecycle management. For this, we need to distinguish between automatic expiration and active revocation. Automatic, time-bound expiration is simply enabled by setting the respective "credential_valid_from" and "credential_valid_until" claims within credentials [29].

For active revocation capabilities, SWIYU's verifiers need real-time status updates without ever directly contacting issuers. This is achieved by the utilization of Token Status Lists (TSL). A TSL is essentially a bitset or bitmap, where each bit or group of bits represents the status of one specific credential. This data structure is then encoded as a whole as a separate JSON Web Token, signed by the owner of the list, who is always an issuer. Each TSL is published within the Base Registry, in a part also sometimes referred to as a separate Status Registry [7], [27].

Every issuer needs to maintain a distinct list for each of their issued credential types. During the issuance of a concrete credential instance, the issuer embeds a unique index number pointing to a specific bit within the list into the credential's metadata. If a credential later needs to be suspended or permanently revoked, the issuer updates the corresponding bit in the TSL, and re-publishes it [7], [27].

During a verification, the verifier fetches the entire latest TSL from the Base Registry and only looks up the corresponding bit(s) in it locally, based on the in-

dex number found in the presented credential. This mechanism is crucial, since as verifiers can download the entire list only, the registry never learns which specific credential was verified, while the issuer is completely bypassed during the check. This design effectively addresses the formerly introduced revocation dilemma, using only the necessary level of centralization needed for up-to-date and trusted status data while also not sacrificing the desired privacy properties [7].

4.6 SWIYU's strengths, limitations, and constraints for the use case of high-trust credentials

Having introduced the most important building blocks of SWIYU, we can evaluate how suitable the trust infrastructure is to host security-critical, high-trust credentials.

The hybrid SSI-like approach provides the advantages that were expected from it. Credentials are stored only locally without a central database, and there is a clear emphasis on privacy-by-design, for example through the use of Token Status Lists and the batch issuance of credentials. At the same time, institutional trust is integrated into the system, and even more importantly, state-operated registries add this authoritative credibility without compromising most advantages of decentralization, since even the hosting authorities cannot silently alter DID logs or status lists. The cryptographic standards relied upon are robust and battle-tested, and both hardware-binding mechanisms and the confidentiality of data in transit are given the attention required for security-critical use cases [1], [7], [25], [30].

Despite these strong properties, the system naturally still relies on certain trust assumptions. On an organizational level, the design assumes that the FOITT or future system owners will continuously and reliably operate the registries without deleting entries, and that the FOJ will perform perfectly thorough background

checks before any Trust Statement can be issued. Cryptographically, while the infrastructure relies on algorithms currently deemed secure, these are not post-quantum safe. Future research and development are already actively being planned to address this [2], [7], [25].

More pressingly for immediate implementations, SWIYU is still in a Public Beta stage. Several open questions and issues stem purely from the maturity state of the project rather than the fundamental design of the infrastructure. For example, certain security features, such as hardware-backed key attestations or JWE-encrypted credential payloads are not yet fully realized across all live components at the time of writing [2].

Though these are not security-related limitations, it is important to note that implementers also need to keep certain constraints in mind specific to the ecosystem. While SWIYU is predominantly built on open standards, it does introduce slight refinements to ensure consistency and minimize attack surfaces. As an example, the ecosystem's specification of did:tdw prohibits the renaming of DIDs and explicitly restricts the use of certain fields in DID Documents that could potentially be used to inject malicious links into the system. Similar minor but mandatory constraints also apply to the OpenID protocols and VC formats used within the infrastructure [28].

In summary, while current maturity-related limitations would require consideration in case of a live implementation, these limitations are expected to be mitigated before the go-live. The additional constraints to open protocols are not problematic at all, since these are included exactly to enhance security and consistency. Implementation-level security is also taken seriously, proven by for example, among others, the existence of a public bug bounty program [31]. Ultimately, the core architecture is robust and successfully finds the middle ground between pure SSI and the required authoritative governance. Even if the aforementioned risk of the fact

that a person holds a certain credential leaking is likely still higher than in legacy systems, the chances of this can likely be mitigated with precaution, for example by temporarily deleting clearances from a wallet when travelling to certain countries. All in all, we can conclude that the SWIYU infrastructure is a promising platform candidate not only for mass credential types, but also for security-critical, e-PSP-like use cases.

5 Requirements and Design Basis for an Electronic PSP System

5.1 Requirements derived from current practices

Before designing the e-PSP system, it is necessary to formally define what we expect of it. Based on the shortcomings of current verification practices outlined in Chapter 3, we can derive a set of core requirements for improving functionality and security. These are the following:

- Strong identity binding: As established previously, a clearance is meaningless if it cannot be securely tied to the presenting individual. The credential therefore must unambiguously belong to a specific person and contain sufficient yet minimized information to allow a verifier to reliably match the presentation to a real-world identity. This binding must support both manual verification against a physical ID card and fully digital checks utilizing the Swiss Beta e-ID.

- Clear authorization semantics: The e-PSP must include an explicit and clear representation of the clearance level. This representation must be directly interpretable by verifier systems so that these can enforce policies automatically.

- Real-time validity checks and revocation capabilities: The e-PSP system must

support both automated time-bound validities and provide a real-time revocation mechanism. Issuers must be able to instantly invalidate a clearance if a holder's risk profile changes, and verifiers must be able to automatically check for this status during verifications.

- Unambiguity during verifications: Since the final decision during verifications will ultimately be made by human operators, the aforementioned burden on them to understand scope and validity must be reduced in every respect. The system must provide clear, unambiguous outcomes that do not require or leave any room for interpretations.

- Data minimization: The system must follow strict data minimization and privacy-preserving principles as long as these do not conflict with security or usability. Verifiers must only receive the minimal data necessary for a decision, and verifications must be as unlinkable as possible.

5.2 e-PSPs as verifiable credentials and recap on design priorities

Since personnel security clearances are fundamentally just high-trust authoritative statements on an individual, they are a natural fit with the attribute-based attestations that are Verifiable Credentials. The above-listed operational requirements also directly map into concrete technical capabilities of either VCs or SWIYU itself.

The practical design and implementation of the e-PSP system is divided into two phases in this work: issuance and lifecycle management, and then verification. As defined in the project boundaries in Chapter 1, the phases are devoted to different levels of attention, with verification being prioritized. While the reasons for this choice are more deeply detailed in Chapter 7, the work now continues with Chapter

6, concentrating on the first phase.

6 Designing an e-PSP Credential, Issuance, and Revocation

6.1 Chapter positioning and goals

Designing a verifiable credential system requires answering four well-scoped engineering questions. What should the credential contain? How can it be issued? How can the credential lifecycle be managed? And finally, how can it be verified? This chapter aims to tackle the first three.

Since the issuance and lifecycle management processes are deliberately addressed at a proof-of-concept level, the primary purpose of this chapter is to demonstrate how e-PSP credentials can be defined, configured, issued, and revoked within the constraints of the SWIYU ecosystem. By proving that the initial creation and subsequent revocation of e-PSPs are technologically viable, the chapter sets the necessary stage for the security-hardened reference verifier implementation that follows.

6.2 Conceptual credential design model

The first concrete task of this chapter is the design of an e-PSP credential. To translate the formal requirements into a minimal yet functional VC, we must ensure that we only store what is strictly necessary for correct verification and lifecycle management. As an example, while we must embed a sufficient amount of data to support

confident identity matching, we must avoid overexposing personal information or making credentials trackable.

To structure these choices transparently and ensure that every included attribute has a well-justified reason to exist, this thesis introduces a three-layer model of credential design. This model classifies properties into the following categories:

- Identity Layer: The minimal attributes necessary to establish the identity of the credential holder.
- Authorization Layer: The decision-relevant properties that dictate the scope and privileges granted by a credential.
- Lifecycle Layer: The fields that enable the management of the credential from the issuer side and the validation of this status from the verifier side.

By consciously categorizing data into these three layers, we can guarantee that the e-PSP remains both lean and functional and can now start filling them up with a concrete set of claims.

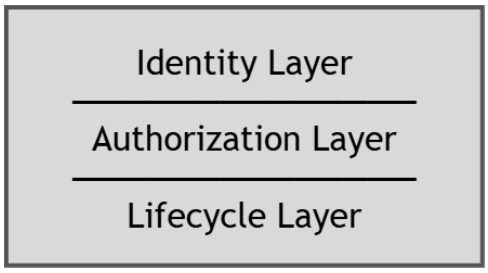

Figure 6.1: The conceptual three-layer model of credential design

6.3 The concrete e-PSP credential

In their concrete form, the layers of the e-PSP credential consist of the following claims:

- Identity Layer: This layer includes the "ahv_number", "family_name", "given_name", and "birth_date" fields. These attributes are enough to define the holder of the e-PSP. While the AHV number is already a unique identifier on its own, a name and date of birth are required to also support user-friendly physical ID based identity checks. It is also important to note that an AHV number is sensitive data and it is likely that in a future version of SWIYU only registered verifiers will be able to query it.

- Authorization Layer: True to the principle of data minimization, the authorization layer can be kept extremely thin and consists only of a "psp_level" claim. In the proof-of-concept, this field is strictly limited to the values of either "GSP" or "ESP", corresponding directly to the two clearance levels introduced in earlier chapters.

- Lifecycle Layer: The lifecycle layer is fulfilled by a single claim called "epsp_number". This is because SWIYU already offers built-in support for both setting a standard time-bound expiration by providing a "credential_valid_from" and a "credential_valid_until" field, and for including status list related information within the credential. These are part of the VC's metadata, not strictly the claim set itself. The extra e-PSP number included in our design is only needed as a unique, issuer-controlled reference ID, necessary to initiate revocation processes. While technically the holder's data could also be used for this purpose, this pseudo-randomized string enables an unambiguous revocation feature for the proof-of-concept. [7], [29]

As an implementation note, all of the above claims are currently represented as simple, string-valued fields within the PoC. While from a data-modeling perspective, utilizing typed fields and constrained enumerations would be a more robust approach, the support for this within SWIYU is limited at the time of writing. Nev-

ertheless, a string-based solution is more than sufficient to demonstrate that building e-PSPs on SWIYU is viable, and the responsibility to check for correct semantics and data formats is simply shifted to the issuer's and verifier's respective application logics.

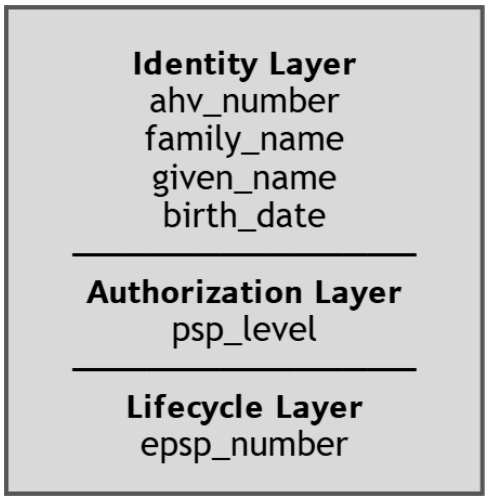

Figure 6.2: The concrete e-PSP claim set within the three-layer credential model

6.4 Configuring the SWIYU Generic Issuer

With the specific claim set established, the next step is to configure the SWIYU Generic Issuer to be able to produce this credential. In practice, this is achieved by defining the "credential_configurations_supported" field within the JSON file containing the issuer's metadata. This field explicitly dictates the structure and capabilities of the credentials the issuer is authorized to offer [28].

Our implementation does not just define a single credential type in this configuration, but three distinct ones named "psp-beta", "psp-beta-enhanced-sec", and "psp-beta-high-sec". From a semantic perspective, these three configurations are completely identical, sharing the exact same structure, claim sets, and value types defined in the previous section. The necessity for three separate credential types

arises solely from the way key attestations can be configured in the Generic Issuer. Since security-critical credentials demand the highest available level of device binding which varies from smartphone to smartphone, our PoC enables the issuer to explicitly select the right level, which requires this design. As an alternative, the wallet could be given the freedom to go with the highest available level [32].

Besides the redundancy stemming from explicit device binding, the rest of the configuration is quite straightforward. All three types follow the "vc+sd-jwt" format introduced in earlier chapters and share a common "vct" (Verifiable Credential Type) identifier of "beta-psp-sdjwt-v1". The fields in "claims" act as the direct configuration-level mapping of the six required fields defined in the previous section, while cryptographic and display-related settings are either self-explanatory or simply follow standard SWIYU defaults and suggested best practices [28].

It is also important to note again that even if that is out of the scope of this work, a deployment-ready issuer system would naturally require additional configuration, security hardening, and more advanced management of private keys. A good starting point for this can be the GitHub page of the Generic issuer, which discusses among others the topics of securing management endpoints and operations using Hardware Security Modules (HSMs) [32].

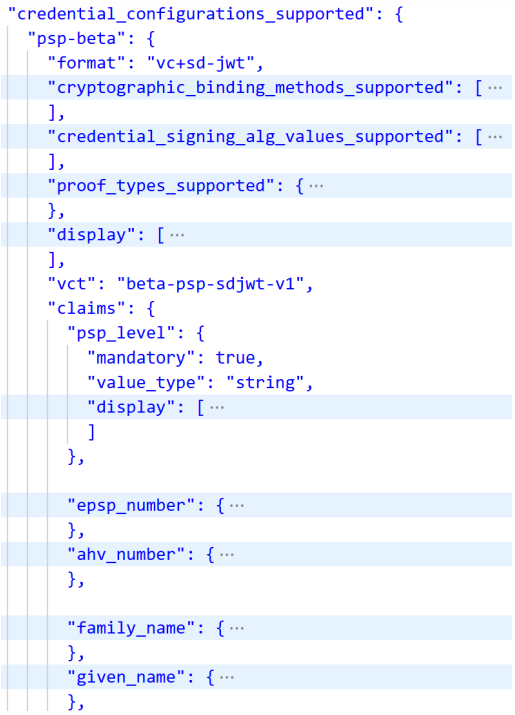

Figure 6.3: A snippet from the Generic Issuer configuration defining the e-PSP credential.

6.5 Issuance flow in the proof-of-concept

Since the Generic Issuer's management interface provides a higher level of abstraction, the business logic does not need to directly orchestrate underlying OpenID4VCI protocol interactions or navigate building the SD-JWT credentials themselves. It can instead concentrate on its main task, handling application-level, operator-facing workflows [29]. At the time of writing, the issuer component of the proof-of-concept is available for testing at `https://issue.epsp.ch`.

To issue a beta e-PSP through the PoC, the operator simply opens this web application, fills a form with the future holder's data, and selects the appropriate clearance level. The validity window of the credential and the aforementioned required level of wallet key protection can also be set. The operator then initiates the issuance process by clicking the appropriate button and thereby generating an

issuance QR code, which the holder can scan and accept or decline the credential.

Behind the scenes, the technical issuance process consists of the following steps:

1. The PoC issuer application gathers the input attributes mapping to future credential's fields, as well as the validity info for the metadata, and builds a request body. The status list data is also injected here.

2. The application submits the assembled payload via a POST request to the Generic Issuer's management API.

3. The Generic Issuer validates the request, creates an issuance session, and returns an offer deeplink.

4. The PoC application encodes this deeplink as a QR code and displays it on the screen.

5. The holder scans the QR code using the SWIYU wallet app. If they decide to accept the offer, the wallet and the Generic Issuer execute the rest of the OpenID4VCI protocol, securely transferring the e-PSP which is then stored locally in the wallet, ready to be presented to verifiers [7], [29].

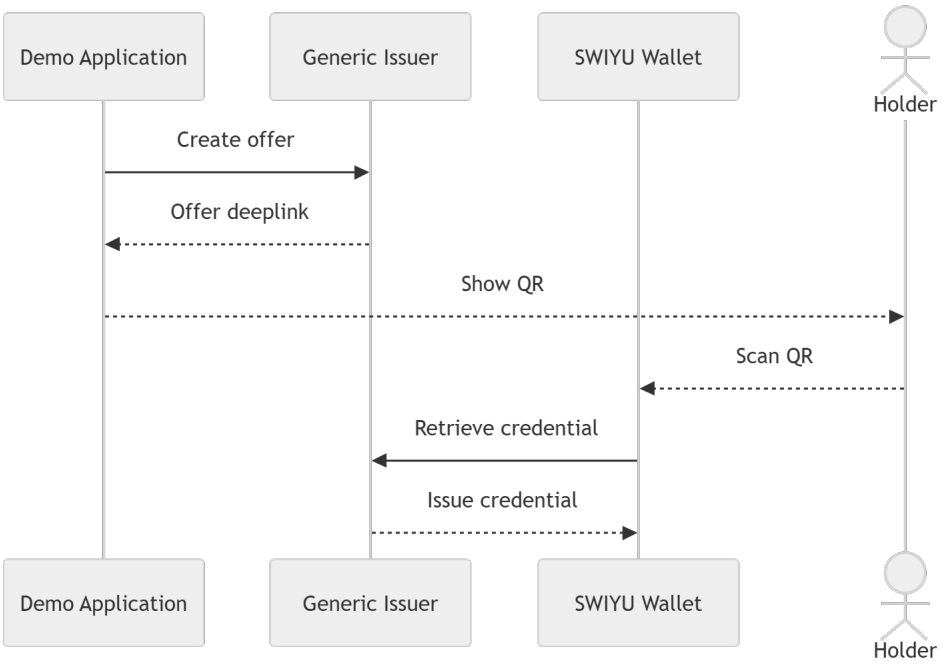

Figure 6.4: High-level issuance flow in the proof-of-concept implementation

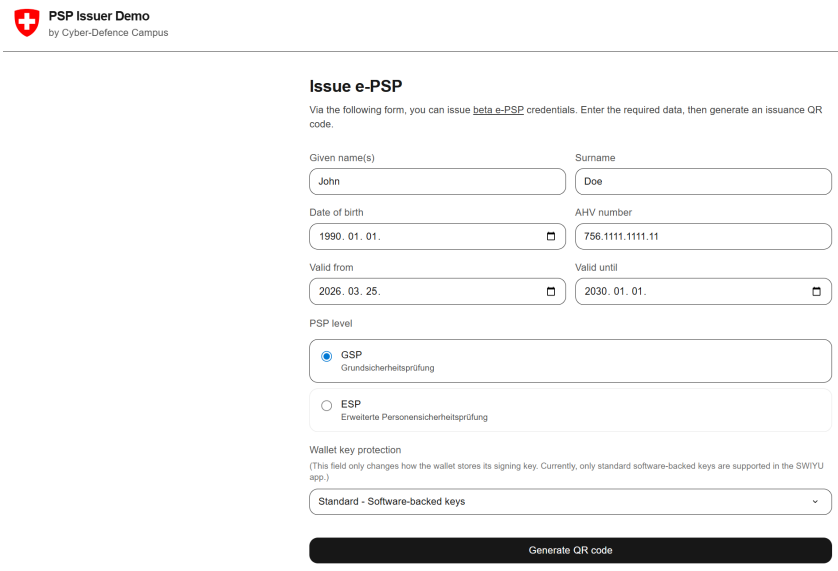

Figure 6.5: The e-PSP issuer interface developed for the proof-of-concept

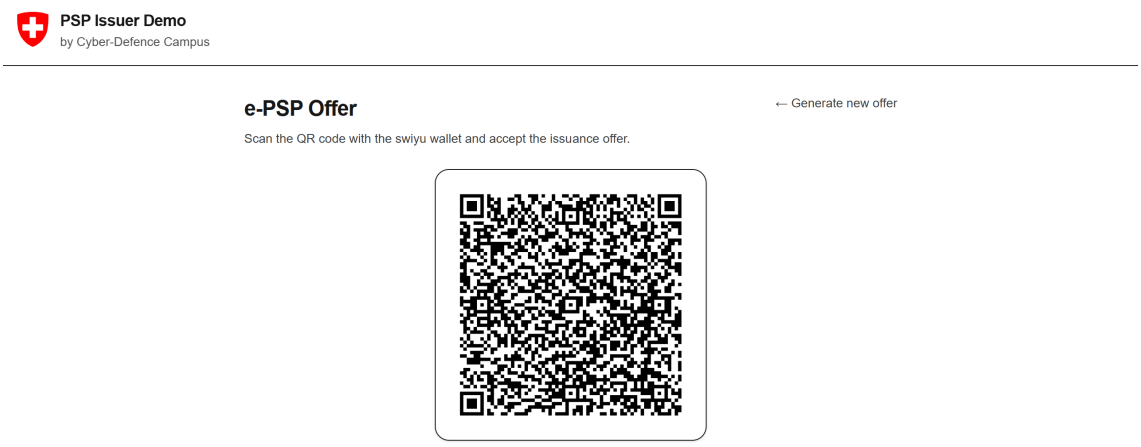

Figure 6.6: The e-PSP issuance offer encoded as a QR code in the proof-of-concept

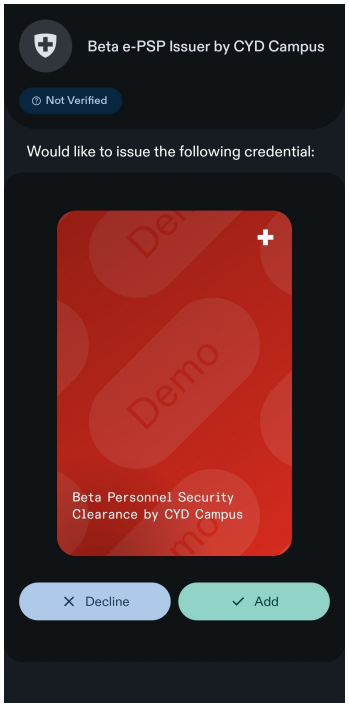

Figure 6.7: The e-PSP issuance offer as in the SWIYU wallet

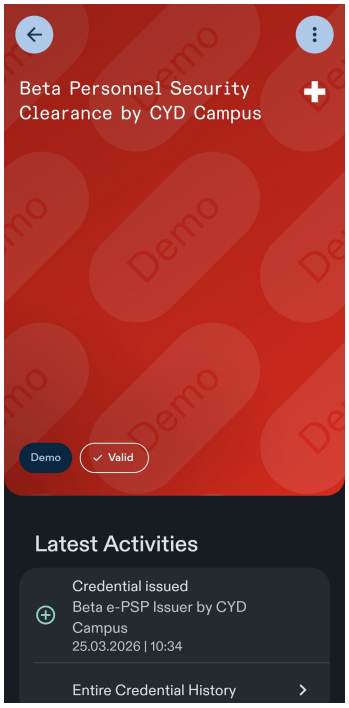

Figure 6.8: The issued e-PSP credential stored in the SWIYU wallet

6.6 Lifecycle handling and revocation

While Chapter 4 already covered the theory of how SWIYU utilizes Token Status Lists (TSL) to enable privacy-preserving revocation, this section presents how this mechanism is implemented within the proof-of-concept. At the time of writing, this part of the demonstration is available at `https://revoke.epsp.ch`.

A revocation feature in practice requires two things in SWIYU: embedding a link to an active status list and the corresponding index during issuance within each credential that verifiers can later rely on for status checks, and a way for issuers to update these lists if necessary. The complexity of these is again abstracted away by the Generic Issuer, since its management interface enables the initialization and initial upload of TSLs to the Status Registry, as well as the previously presented injection of status list data during the issuance flow. It is also possible to update credential statuses through it, however this requires the reliance on internal creden-

tial IDs that the Generic Issuer uses to keep track of credentials [7], [29]. This is where the "epsp_number" claim of the lifecycle layer comes in, acting as a user-facing counterpart to this internal ID. The mapping between these two is handled by the business logic of the proof-of-concept.

The concrete revocation workflow thus consists of the following steps in the e-PSP system:

1. During issuance, an "epsp_number" is generated and embedded into the credential, then securely mapped to the Generic Issuer's internal ID in the PoC application's backend. The status list information is injected into the metadata.

2. To revoke a clearance, the operator inputs the e-PSP number. The application queries its internal mapping to find the corresponding SWIYU credential ID.

3. The PoC application sends a revocation request using this internal ID to the Generic Issuer. The Generic Issuer updates the corresponding bit in the TSL, signs the new list, and pushes it to the Status Registry.

4. At this point, the credential is cryptographically invalidated, and any verifier or wallet fetching the status list in the future should correctly identify the credential as revoked [7].

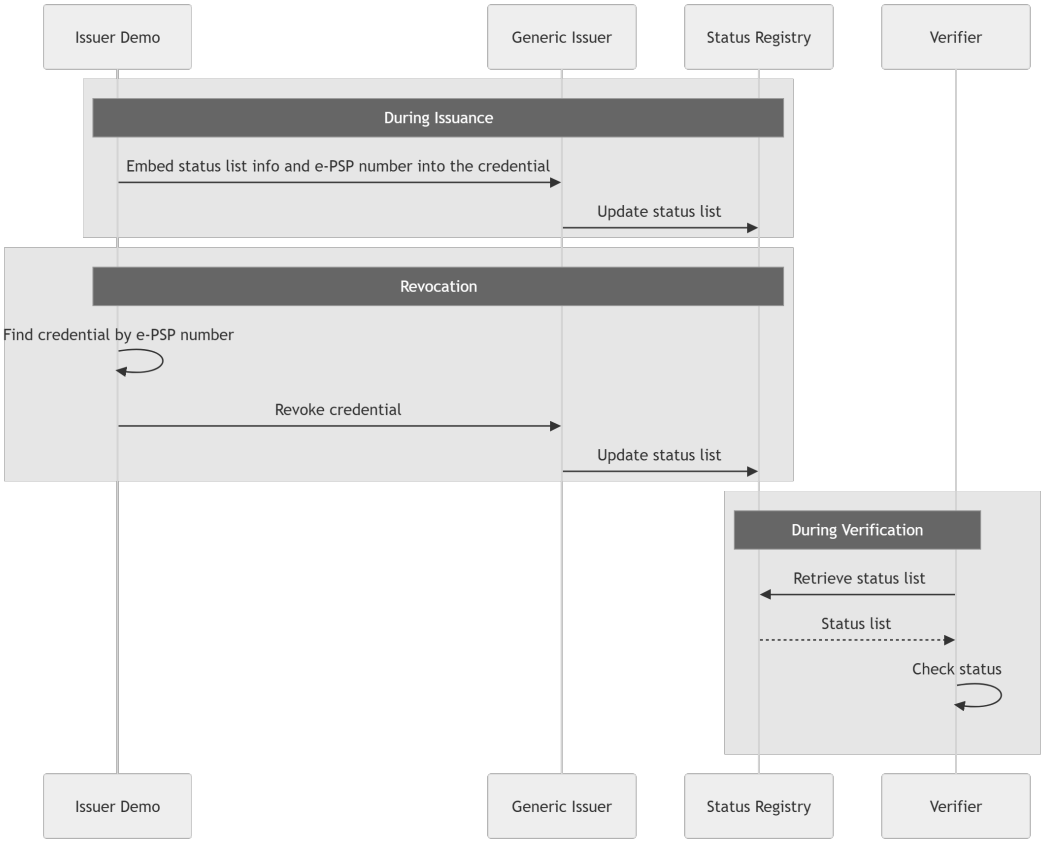

Figure 6.9: High-level revocation-related flows across the different lifecycle stages within the proof-of-concept

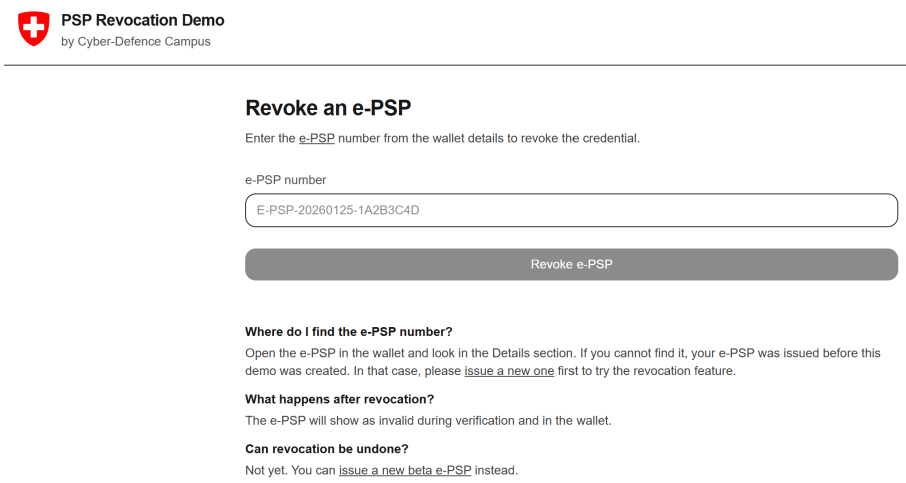

Figure 6.10: The e-PSP revocation interface within the proof-of-concept

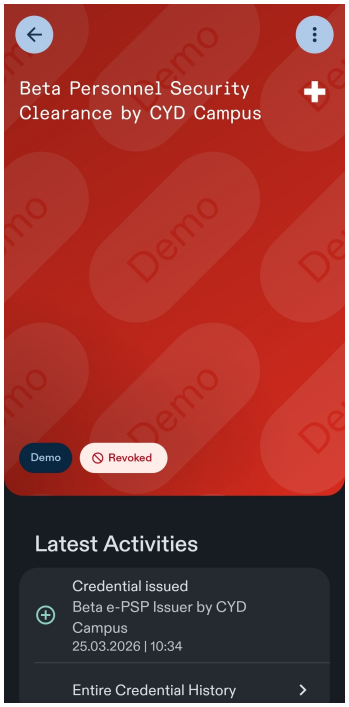

Figure 6.11: The revoked e-PSP credential as displayed in the SWIYU wallet

6.7 What this chapter demonstrates and what it does not

The above sections demonstrate that electronic Personnel Security Clearances can be designed, configured, issued, and revoked within the constraints of the SWIYU ecosystem. Going beyond sheer theory, the working proof-of-concept is publicly available for testing. Besides the demonstration, the proposed three-layer model separating identity, authorization, and lifecycle properties can act as a reusable pattern for turning similar credentials into VCs in a functional yet minimal manner.

However, it is still important to note that while the proof-of-concept successfully implements issuance and revocation features, in accordance with the established scope, this chapter deliberately avoided defining a production-ready reference architecture for an e-PSP issuer. The reasons for this decision are finally detailed in the upcoming section.

Ultimately, the chapter's goal was achieved, and the issued e-PSPs now reside securely in the SWIYU wallet. The focus of the thesis can now shift to how these can be verified.

7 A Reference Verifier for Security-critical Credentials on SWIYU

7.1 Why verifier implementations deserve special treatment

Since unlike issuance and revocation that were treated at a proof-of-concept level, this chapter presents a more deeply analyzed reference verifier implementation, it is time to finally answer where this asymmetry in the thesis stems from.

The reason is twofold. From an engineering perspective, verifier implementations simply have a higher potential for generalizability. Since issuers in most cases are not meant to be publicly available applications, they will likely often be integrated into existing backend infrastructures, where their interfaces, architectures, and desired security properties will vary significantly, completely depending on the internal technology stack, networks, and access control models of the organizations operating them. Even if they are implemented as public web applications with access for authorized users only, their concrete feature set will still highly be dependent on specific use cases and business goals, making the idea of defining a universally applicable reference architecture for an issuer one of little real-world value. Since even

around our concrete use case there are still open questions that will be presented in later chapters, attempting the creation of an e-PSP specific reference issuer would currently also be speculative.

On the other hand, when it comes to the characteristics of an e-PSP verifier, both the constraints of SWIYU and the general requirements leave us significantly less freedom. Even more importantly, this can also be stated about other similar use cases, and it is likely that a major group of SWIYU verifiers will possess similar well-defined attributes. This well-defined group is going to consist of web-based or mobile applications meant to be accessible by everyone or broader groups of people, implementing a similar core feature set of online verifications resulting in operator-facing messages, not automatic decisions. Since these similar systems will likely also share similar architectural patterns, workflows, and security properties regardless of whether they are checking an e-PSP or another credential, a reusable reference verifier implementation for this use case can provide direct real-life value not only for us but possibly also for other future implementers. The main goal of this chapter, thus, is not only to present the verifier of the e-PSP system, but to propose even more general best practices wherever possible. While future features, such as Check app based fully offline verifications, are expected to somewhat change the horizon of verifications, these are more likely to expand the portfolio of verifiers, not to make this first wave of implementations obsolete.

The second reason is that the security relevance of verifier implementations built on top of existing SWIYU components can easily be overlooked. Since the Generic Issuer actually handles an overwhelming amount of an issuer's responsibilities, the primary remaining critical task falling on upper layers is to ensure that only authorized people have access to the system, which is a responsibility hard to forget since no implementer would willingly expose the issuer interface of a security-critical credential to the public. However, the interface of a verifier, as stated above, is more

likely to be meant for the public. Even more importantly, in this case, the Generic Verifier only seemingly takes care of the most critical work. While the validation of signatures and registry statuses is undeniably the foundation of trust, it is important to recognize that secure cryptography on its own does not guarantee correct real-world verification outcomes. In practice, in SWIYU's design, the final verification decision is not solely based on the cryptographic validity of the credential, but also on how the upper layers of the verifier interpret its content and what expectations they define. Furthermore, even if the application logic is perfectly sound, as mentioned above, in several use cases, the ultimate decision is expected to be made by a human operator, which makes a robust and unambiguous interface even more important.

If future implementers fall into the trap of viewing the verifier's upper application layer simply as a thin interface built on top of an existing service, an army of vulnerable components within SWIYU could emerge. This army could also grow quite large, since it is also expected that there are going to be more verifiers than issuers in the ecosystem. Therefore, this thesis argues that the upper layer of these must be explicitly treated as security-critical components in their own right, and aims to demonstrate the right approach by presenting a security-hardened implementation.

Figure 7.1: Conceptual positioning of the identified verifier group's application layer within the SWIYU ecosystem

7.2 Responsibilities of verifier implementations

The fundamental responsibility of the verifier's application layer is to determine whether an otherwise valid credential should actually be accepted within a specific operational context, and (in the above identified set of use cases,) to securely and unambiguously communicate this to a human operator. To be a bit more precise, we can define five core tasks that the implementer and their implementation need to handle. These are the following:

- Policy enforcement: While the trust infrastructure can confirm a credential's technical validity, the application layer must determine if it is the right credential for the given context. In the case of e-PSPs, this means correctly configuring the DID of the e-PSP issuer as the only accepted source, checking the VC type, verifying that all expected attributes are in the presentation, and ensuring that a sufficient clearance level is presented.

- Workflow orchestration: The verification itself is rarely a single, atomic action, but rather a multi-step process. The implementation therefore must ensure that specific steps of a sequence such as initiating a session, requesting a credential, verifying it, and potentially binding it to a physical or digital identity are all executed in the correct order, and individual steps cannot be bypassed.

- Session and state management: Because verifications occur over time and consist of multiple requests and responses, the system must strictly manage verification session context. It is the implementation's responsibility to track ongoing verifications, prevent the reuse of old results or the mixing of verification states, and time out interactions if needed.

- Decision interpretation: Since in our use case, the final decision maker is a human operator, the implementation must accurately translate inner technical

failures such as invalid signatures, revoked credentials, or policy mismatches into clear, unambiguous operator-facing messages. The failure to clearly communicate results can significantly increase the risk of an incorrect human decision.

- Deployment security: Finally, the verifier implementation must protect its own integrity. This means as an example defining strict exposure boundaries and shielding the Generic Verifier from direct external manipulation.

Any SWIYU-based verifier in the identified group is ultimately dependent on implementers correctly handling all of these responsibilities, and neglecting even a single one can lead to the complete failure of an otherwise cryptographically sound system. The failure and attacker models presented in the next section further analyse what exactly can go wrong if that happens.

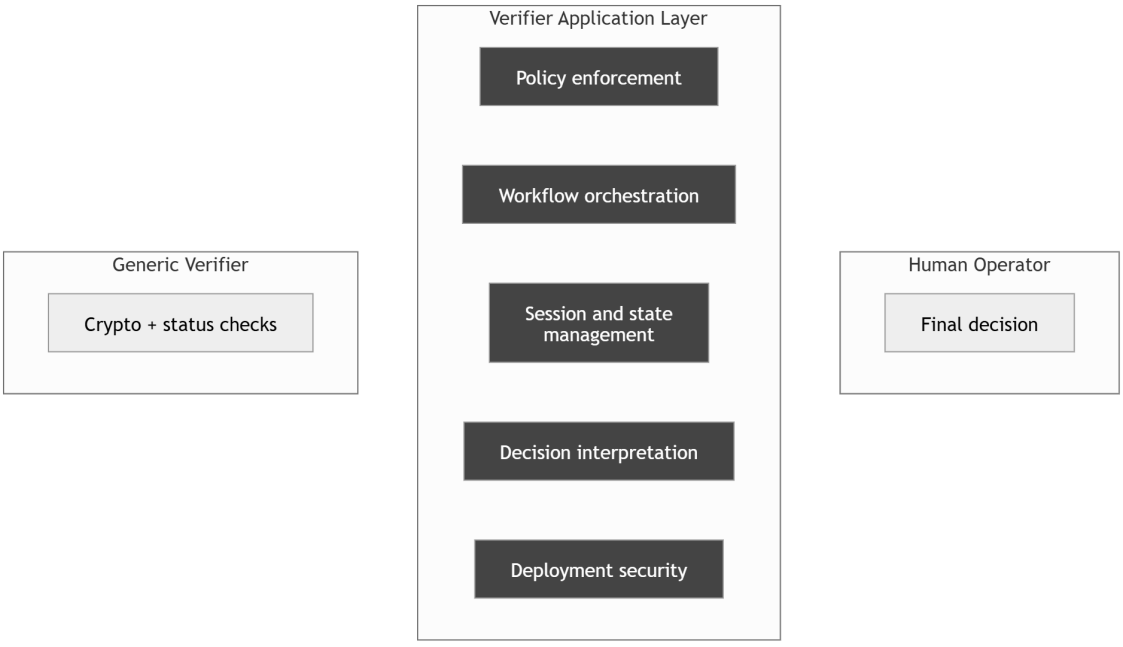

Figure 7.2: The core responsibilities of SWIYU verifier implementations within the formerly identified group of use cases

7.3 Failure and attacker models

A formal definition of what can go wrong during verifications is an important input for the design of the reference verifier. Although threat modeling for SWIYU itself has been performed by another work before [33], our needs are somewhat different. Because the implementation resides above the cryptography layer, we need to examine possible vulnerabilities on this specific abstraction level. In this context, a successful attack does not mean breaking the underlying cryptography, which is assumed to be perfectly secure, but rather a failure is defined as any situation in which the verifier produces or leads to an incorrect acceptance decision, even when the underlying trust infrastructure itself functions perfectly.

Based on the responsibilities listed in the previous section, the main types of failures can be defined as follows:

- False acceptance: Accepting a credential that is invalid, untrusted, or does not meet the necessary policy constraints. While the infrastructure below is assumed to be perfectly secure, a misconfiguration or incorrect implementation can still lead to this outcome.
- False binding: Accepting a perfectly valid credential that is presented by the wrong individual.
- Downgrade acceptance: Accepting a credential that satisfies weaker requirements than intended for the specific context.
- Revoked or expired credential acceptance: Failing to correctly enforce validity and status checks. As in the case of a false acceptance, an incorrect implementation can lead to this.
- Incorrect acceptance caused by the UI: Operator misinterpretation due to unclear or ambiguous system output.

- Result tampering and flow manipulation: Attempts to bypass workflow steps, reuse intermediate results, or manipulate client-side behavior to force an acceptance state.

While all the above issues are scenarios where something invalid is accepted, the opposite of this, false rejection, is of course also problematic. Since this work argues for a strict fail-closed principle in high-security use cases, this is not specifically listed, but following the later presented design principles naturally also lowers the risk of such attacks significantly. Attack types where the verifier as a whole is made unavailable is out of the scope of this work.

The possible attacker types who might cause the above listed failures can also be formalized. Assuming again that the trust infrastructure cannot be attacked directly, the adversaries are the following:

- Malicious holder: Someone attempting to get a credential accepted without meeting the requirements, for example, by presenting fake, stolen, or lower level credentials.
- Network attacker: An adversary attempting to manipulate or replay communication between the client and the verifier endpoints.
- Malicious or careless operator: An operator interacting with the system deliberately incorrectly or misinterpreting outcomes.
- External attacker: An adversary targeting exposed services or overall system availability without trying to be a man in the middle.
- Malicious or careless insider: An insider causing misconfiguration, incorrect deployment, or incorrect implementation carelessly or with malicious intent.

This formalized adversary model can help us establish the verifier's core security-related design principles in the next section.

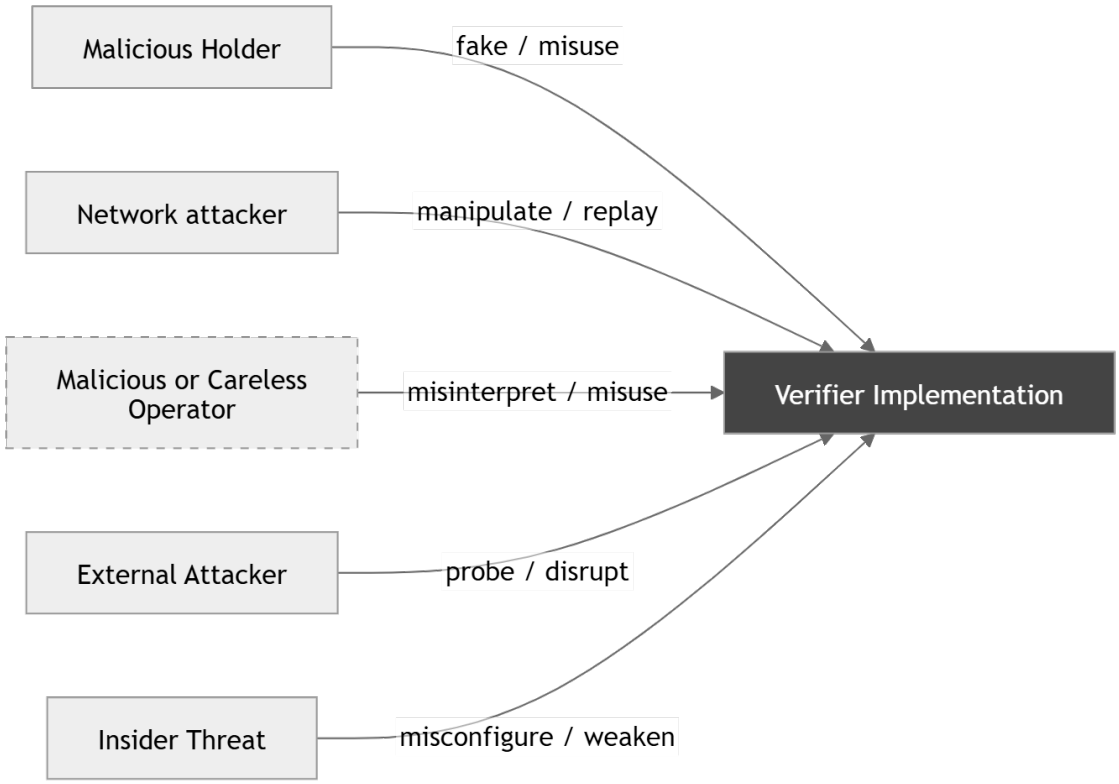

Figure 7.3: The threat model of verifier implementations above the layer of the Generic Verifier

7.4 Design principles for a secure verifier

While web-based SWIYU verifiers are of course recommended to follow classical web security best practices, their role and therefore security needs are somewhat different compared to those of many traditional applications. While availability and system integrity continue to be common requirements, traditional data protection may be less critical, simply because there is less data in a correctly implemented verifier to be protected. Instead, decision correctness and decision integrity emerge as critical requirements, which also somewhat shifts the focus of security engineering. To translate this and the previously defined threat model into a secure system design,

we can define the following core security principles:

- Server-based enforcement of decision integrity: All critical checks, policy validations, and acceptance conditions must be strictly enforced on the server side. A client must never be able to influence verification outcomes.

- Explicit state control and flow enforcement: A final trust decision must always depend on an entire verification context, never on the isolated validities of individual requests. Therefore, the verification process must be strictly stateful. All verification actions must follow a well-defined sequence, and any out-of-order steps or invalid state transitions must be explicitly rejected.

- Fail-closed principle and strict session isolation: The system must strictly follow a fail-closed principle meaning that any incomplete, ambiguous, or uncertain state must result in a definitive rejection and never in acceptance. Verifications must be strictly isolated, it must be impossible to mix or reuse them across sessions. Once a session reaches a terminal state, it must be impossible to continue and a completely new verification must be started from scratch.

- Explicit configuration as a policy: No configuration or trust assumption may ever be left implicit. Accepted credential types, valid issuers, and all context-specific rules must always explicitly be defined either in the configuration or the business logic.

- Unambiguity and clear outcomes: In any case where the final decision is made by a human, this operator must explicitly be considered an integral part of the security model. The verifier's interface and decision points must always be minimal, clear, and easy to understand. The system must never require an operator to infer hidden states or make decisions based on ambiguous output.

7.5 Architecture and trust boundaries

The reference implementation consists of three main components, and the architecture follows a Backend-for-Frontend (BFF) pattern, combined with a secure, single entrypoint. The three components are the following:

- The Reverse Proxy: The design exposes exactly one public interface through a reverse proxy, acting as the external entrypoint. This is responsible for features such as TLS termination, routing requests for inner components, and acting as a boundary between them and the public internet in general.

- The Web Verifier: The component containing the user interface layer and the backend control logic. It is responsible for orchestrating the verification workflow, enforcing policy constraints, managing the session state, and interpreting the verification results for the operator. It is also designed in a way that the client's browser is strictly never trusted to make verification-relevant decisions.

- The SWIYU Generic Verifier and its database: The Generic Verifier is integrated into the architecture as an internal service. It handles the formerly discussed responsibilities of protocol interactions, cryptographic validation, and trust resolutions [24], [34]. Both the Generic Verifier and its database are strictly only reachable through the Reverse Proxy.

The above architecture creates well-defined trust boundaries within the system by design, and communication between components and the outer world is explicit and tightly controlled. With the architecture and the trust boundaries established, the next step is to define how data and control logic flows through components.

Figure 7.4: The reference verifier components and trust boundaries

7.6 Verification workflow and state model

The correctness of a verification decision depends not on separate components, but on how all these components work together. In the reference implementation, a high-level verification flow consists of the main steps listed below. While this workflow and the following state model are naturally presented as they are in the concrete e-PSP-specific implementation, the general template can still easily be reused for the broader use case formerly identified.

1. Policy selection and session initialization: The human operator always initiates the verification process by selecting the required verification policy for a specific scenario. In our use case, this means specifying either GSP or ESP as the required clearance level. The Web Verifier then creates a new session and instructs the Generic Verifier to generate a presentation request.

2. QR code generation and presentation: The presentation request is encoded into a single-use QR code and displayed on the operator's interface.

3. Credential presentation: The holder scans the QR code using the SWIYU app, reviews the request, and submits the e-PSP.

4. Asynchronous verification by the Generic Verifier: The SWIYU Generic Verifier takes care of cryptographic validation, and signature and status checks.

5. Retrieval of the asynchronous verification's results: The Web Verifier retrieves the technical verification result from the Generic Verifier through internal API calls.

6. Application-level validation: The Web Verifier evaluates the retrieved content, checking whether the presented credential actually satisfies the selected policy. In the case of e-PSPs, this practically means checking for the existence and content of the "psp_level" claim.

7. Intermediate decision: Even if all the above checks pass, the decision on acceptance is not final yet, and the verifier transitions to an intermediate state. At this point, the e-PSP is cryptographically validated and meets the required level, but an identity binding step is still necessary for it to be accepted.

The need for this intermediate state directly stems from the failure model defined earlier, which explicitly states that a valid credential presented by the wrong individual is one of the possible breach attempts. The system therefore always enforces this identity binding, which the operator can either perform through a manual ID check, or through requesting the Swiss Beta e-ID. If the second option is chosen, a flow similar to the above will be orchestrated again, only with another credential type. This process is also presented on Figure 7.5. Since in the course of the first flow the identity layer attributes of the e-PSP are also requested, the two holder's data can easily be compared. Either way, the Web Verifier always ensures that this step occurs strictly after a successful PSP validation and that either a manual or automatic comparison of the attributes is conducted before a final acceptance state becomes reachable.

As mentioned before, states in the reference verifier play a crucial role and the entire workflow is enforced through an explicit state model validated on server side. The exact states in the reference implementation are defined as the following:

- “SELECT_POLICY”
- “PSP_VERIFICATION_PENDING”
- “PSP_VERIFIED”
- “IDENTITY_BINDING_REQUIRED”
- “IDENTITY_VERIFICATION_PENDING”
- “READY_FOR_DECISION”
- “FINAL_SUCCESS”
- additional non-success outcomes

While the model is explicitly defined on a theoretical level, the concrete implementation does not rely on an explicit monolithic finite state machine. Rather, a hybrid approach is used, meaning that correct transitions are enforced through a combination of server-side session and route validation, application-level workflow logic, and controlled UI progressions. Server-side API routes within the BFF layer act as controlled entry points for all actions, validating incoming requests against the current state to ensure only predefined transitions are possible. The concrete set of these allowed transitions can be observed in Figure 7.6. While this hybrid design decision stemmed mostly from the actively evolving, agile nature of the project, it is sufficient to guarantee correct workflow progression. Future implementers can of course consider a more centralized architecture.

It is also important to note that non-success outcomes are also treated as full states in their own right rather than mere edge cases. Any state can transition to “TERMINAL_INVALID” (if conditions are unmet), “TERMINAL_TIMEOUT” (if the session expires), or “TERMINAL_ERROR” (if unexpected issues occur). In accordance with the fail-closed principle, reaching any of these states immediately

results in the failure of the verification process and all intermediate results are disposed of. In this case, a restart is always necessary. Only once all required steps are successfully completed does the system transition to "FINAL_SUCCESS" and produce a final, clear decision of validity for the operator.

As a final implementation note, the verification workflow and all multi-credential presentations are also expected to become even more streamlined in the near future, when SWIYU fully supports the Digital Credentials Query Language (DCQL). Once that is the case, verifiers will be able to request fields from various credentials such as from an e-ID and an e-PSP within the same presentation, making two separate QR scans in our flow unnecessary [34].

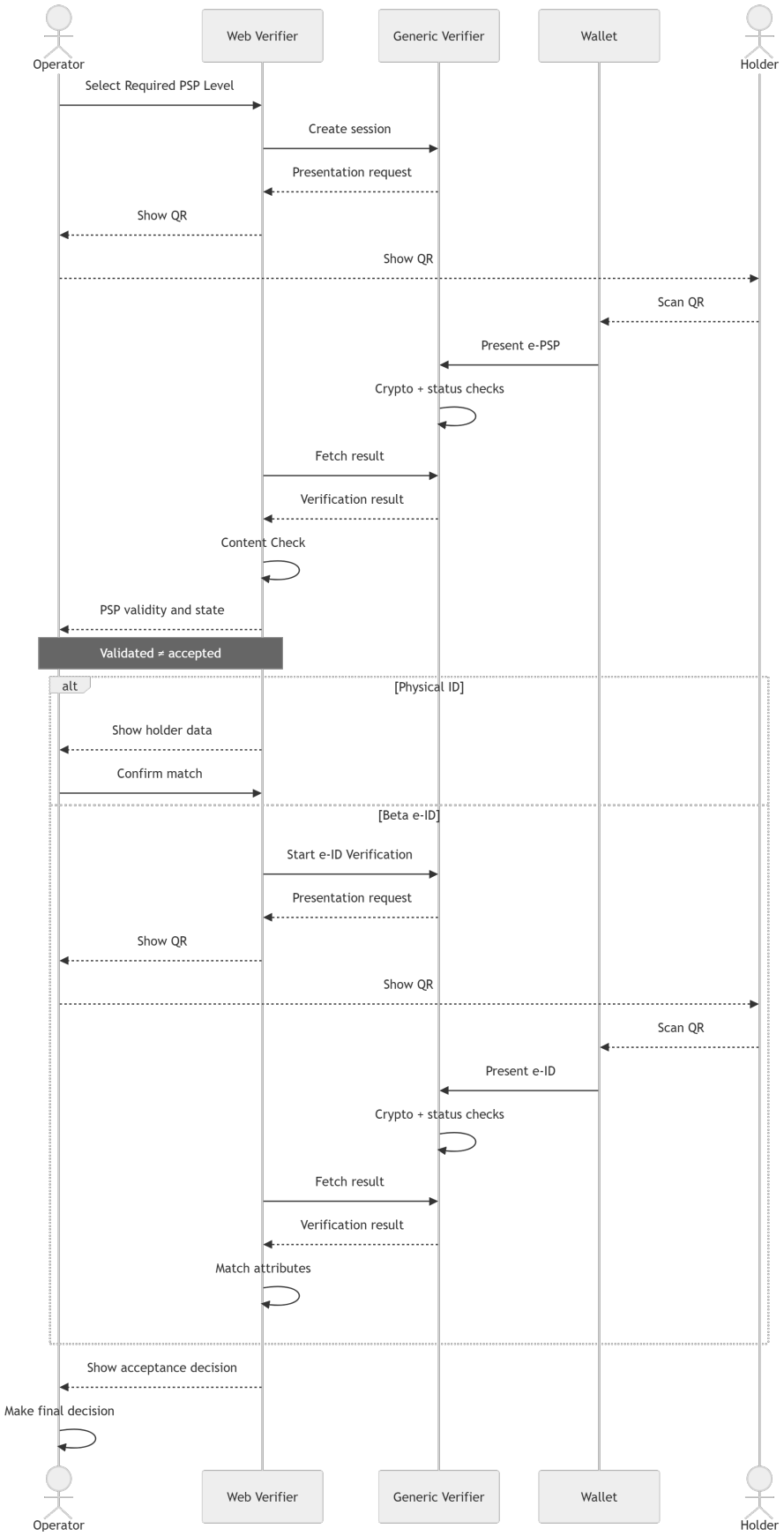

Figure 7.5: High-level verification workflow of the reference implementation

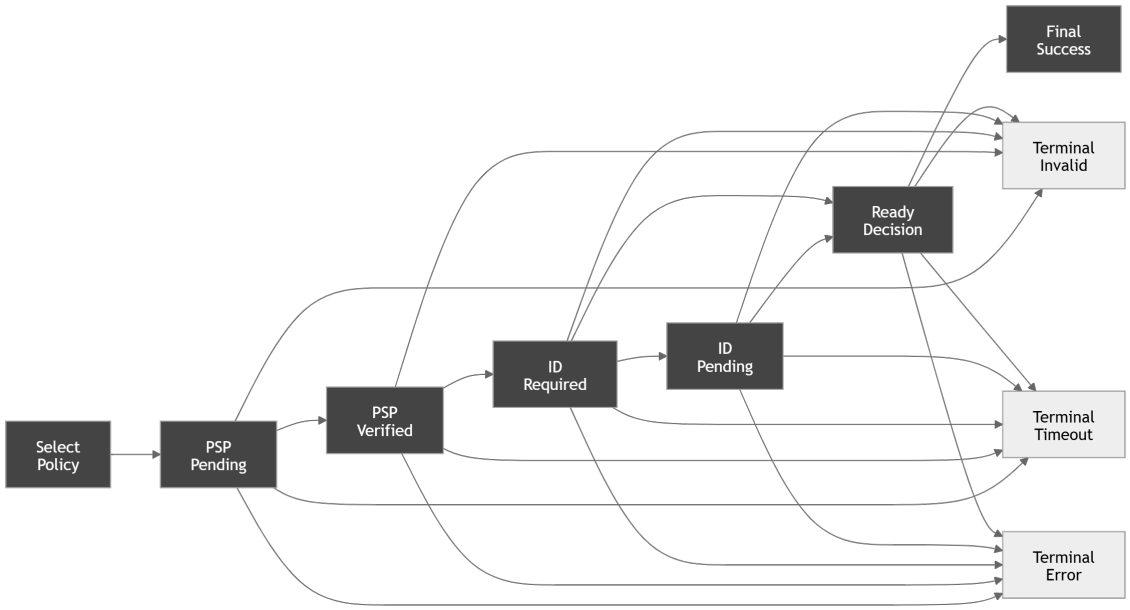

Figure 7.6: The verification state model with the allowed transitions

7.7 Implementation details and demonstration

Having established the theoretical architecture, trust boundaries, and state model, this section can now present how all this is practically implemented in the proof-of-concept.

In the concrete implementation, the Web Verifier component is a Next.js application, which was mainly chosen for its native support for the BFF pattern and server-side isolation, and for being battle tested in production-ready applications. To ensure strict service isolation and consistent deployment behavior, the entire architecture is also containerized. Separate Docker containers are utilized for the Web Verifier, the Generic Verifier, and its PostgreSQL database. The frontend of the application is a completely responsive Progressive Web App (PWA), which enables the usage of the proof-of-concept across a wide range of devices.

A core design principle of the verifier was ensuring minimal ambiguity, with

clear messages and options for its human operators. The following screen captures demonstrate both the fulfillment of this principle and that the proof-of-concept solution is functional in general.

The verification process begins with the selection of the required PSP level on a clean initialization screen.

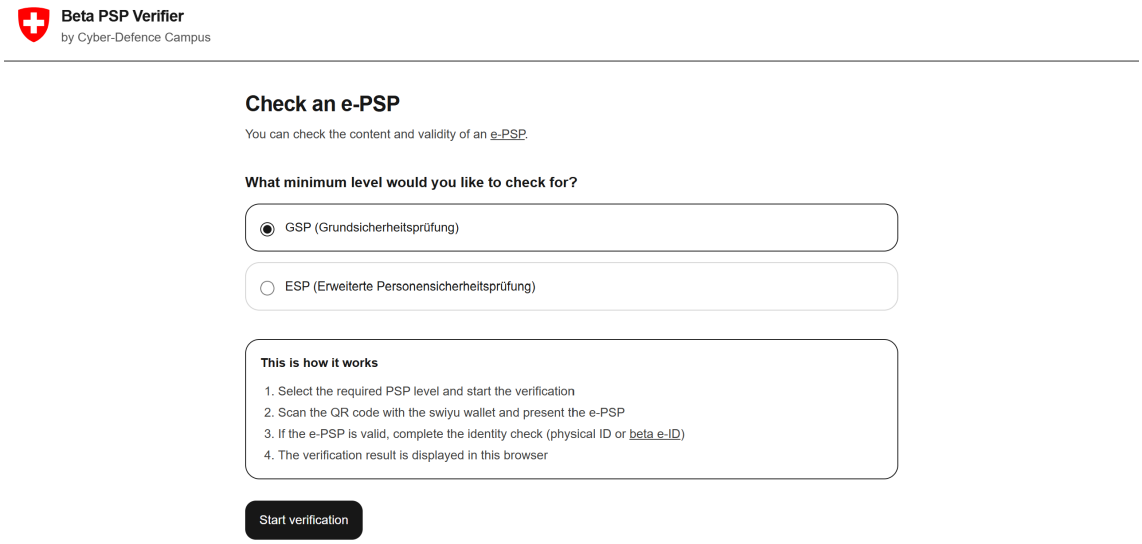

Figure 7.7: The startup screen of the reference verifier implementation

Once the expected level is selected, the single-use QR code is immediately generated and is ready to be scanned by the holder.

Figure 7.8: The presentation request in the reference verifier implementation

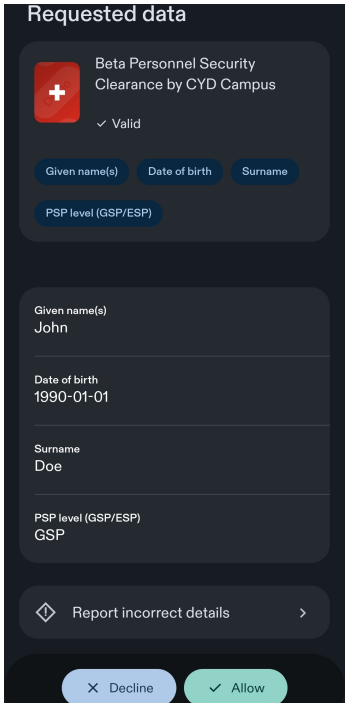

Figure 7.9: The presentation request in the SWIYU wallet app

If the holder agrees to share the data and the e-PSP is both valid and meets the

required minimum level, the intermediate state is reached. Though the PSP's state is already acknowledged, the UI is designed to clearly show that this is not yet a final decision and offers the mandatory choice between a physical or an e-ID based check.

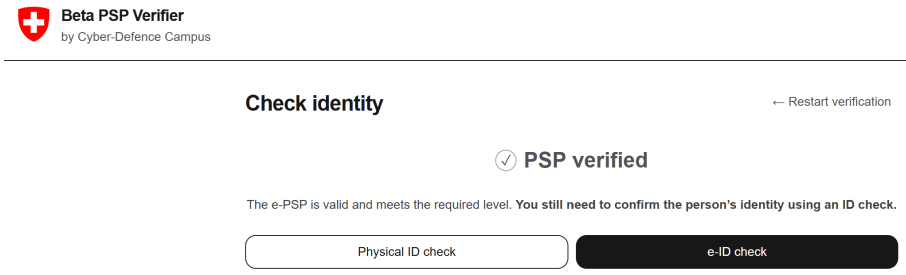

Figure 7.10: The intermediate decision screen in the reference verifier implementation

If the user opts for a physical ID check, the holder's data is presented on the screen and it is the operator's job to compare these to a physical ID card.

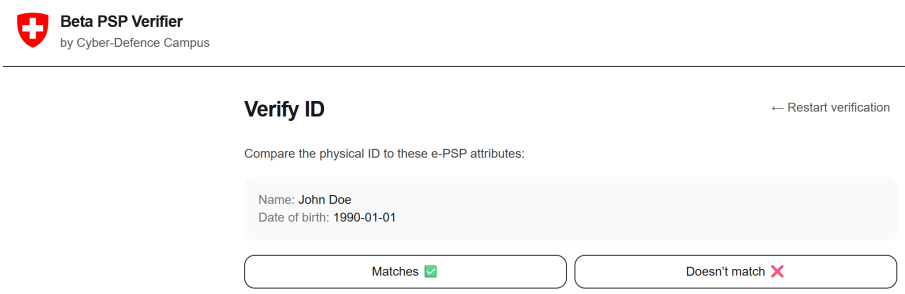

Figure 7.11: The manual ID verification screen in the reference verifier implementation

If the e-ID check option is selected, yet another QR code is generated and the presentation request needs to be accepted in the SWIYU app again. Since the interfaces of these steps are very similar to the respective ones during the e-PSP verification, these screenshots are omitted from the thesis. Either way, no matter whether the operator explicitly indicated that the credentials match or it was retrieved and compared by the verifier, at this point, a final decision screen is reached. The positive outcome can be observed in the next picture.

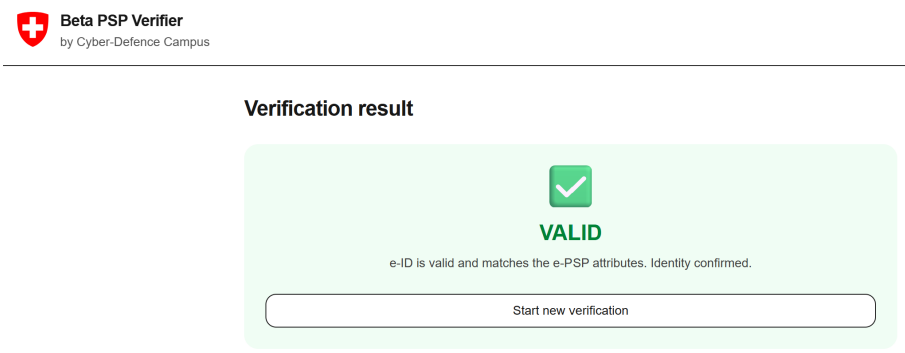

Figure 7.12: A positive final decision screen in the reference verifier implementation

Complying with the aforementioned principles, unsuccessful outcomes look quite similar to the success screen. Whether it is an identity mismatch, insufficient PSP level, or session timeout, the output is a definitive, color-coded terminal screen, with the session immediately locked, and the only recovery option offered being a new verification. This is of course also the case if an invalid state is already reached after the e-PSP presentation.

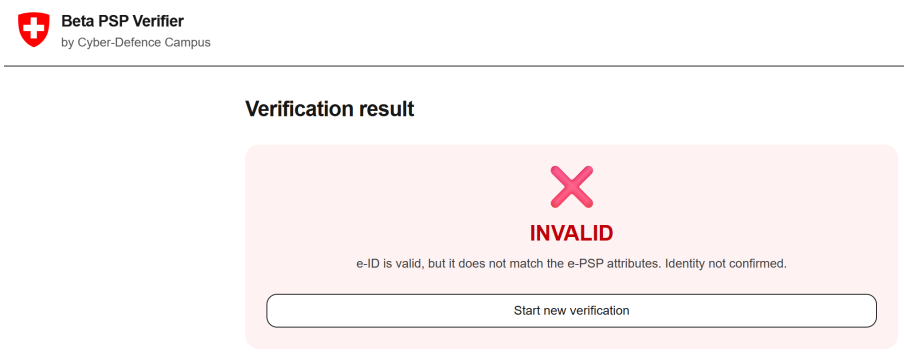

Figure 7.13: A negative final decision screen in the reference verifier implementation with the reason being a holder mismatch

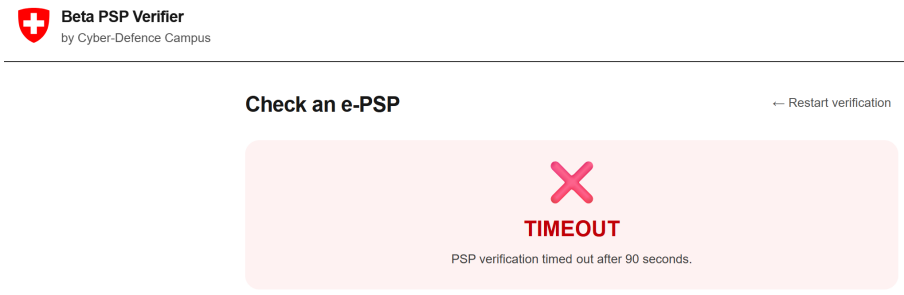

Figure 7.14: A negative final decision screen in the reference verifier implementation with the reason being a session timeout

With issuance demonstrated in Chapter 6 and the reference verifier fully implemented and demonstrated here, the e-PSP proof-of-concept is complete and operational. The focus of the thesis can now naturally progress to the evaluation of the developed artefact, assessing its viability and the underlying infrastructure's readiness, and presenting the feedback received from stakeholders. Then the focus can be widened back even more again, reflecting on the research's general implications and contribution.

8 Evaluation of the Proof-of-Concept

8.1 Evaluation approach, technical viability, and bug reports

The completed proof-of-concept that was the subject of evaluation consists of the web-based issuer and revocation demonstrations, the official SWIYU mobile wallet, and the reference verifier implementation, all operating on top of the SWIYU Base and Trust Registries. Together, these components form the complete, end-to-end e-PSP system.

The evaluation focused on this artifact's practical behavior and correctness under realistic usage conditions, the enforcement of core security-relevant features, and the general feasibility of the system, not on formal benchmarking or user studies. The reason for that is that the results of the latter would be highly dependent on deployment-specific infrastructure and finalized issuer-side requirements and would therefore likely not produce useful output for an iteratively developed prototype built on an infrastructure that is actively being developed itself.

Throughout the development process, all user-facing features of the proof-of-concept were systematically tested. The methods for that included exhaustive manual testing against pre-defined expected outcomes, automated regression testing during development, an adversarial validation of the most crucial expectations, and limited automated testing of SWIYU itself. These approaches together led to the

discovery of multiple implementation-level bugs and system inconsistencies both within the PoC and the underlying trust infrastructure. While the former were simply fixed in the course of the continuous development of the prototype, the latter were reported to and since have mostly been fixed by SWIYU's development team. An example of such a reported issue can be found under this link[1].

Given the project's pioneering nature, the detection of a certain number of issues was expected. This therefore does not change that the evaluated artifact was concluded to be technologically highly viable. The prototype successfully executed the full credential lifecycle and the underlying formats and protocols worked as expected, as can be confirmed by anyone testing the publicly available proof-of-concept. Known issues still waiting to be fixed or notably still missing security features such as key attestations or JWE-encrypted payloads in transit are all expected to be addressed before the go-live and do not undermine the structural viability of SWIYU-based e-PSPs [2].

While the general feasibility of the prototype and the underlying infrastructure is demonstrated, it is important to note that a systematic, deep security evaluation of SWIYU itself was intentionally excluded from the scope of this work. Such an analysis was already conducted in a dedicated, separate thesis published in the summer of 2025 [33]. Therefore, as expected, while some of the formerly mentioned found and reported issues do have security implications, no fundamental flaws or architectural shortcomings were discovered from this perspective either.

As a final note, it is important to remember that the technical viability of the completed system as a whole is just one aspect of the evaluation this work conducts. The evaluation of SWIYU itself as a platform candidate has already been presented in detail in Chapter 4, while the more general operational and societal readiness for the e-PSP system is detailed in Chapter 9. Before that perspective can be discussed,

[1] `https://github.com/swiyu-admin-ch/swiyu-verifier/issues/238`

the way of stakeholder feedback collection is presented in the next section.

8.2 Stakeholder feedback collection

Besides technical validation, the operational viability of the artefact was also assessed. Over the course of the last few months, the proof-of-concept was presented to more than twenty stakeholders from various Swiss governmental organizations, including representatives of the administration, armed forces, legal experts, and developers of SWIYU. Through these demonstrations and the discussions that followed, we received valuable insights into constraints, benefits, and potential blockers.

While the general feedback to the conceptual idea was mostly encouraging and multiple parties confirmed the original research problem and showed interest in the solution, it also became clear that various open questions still need to be answered before personnel security clearances can be hosted on top of the SWIYU ecosystem. While the technology itself works, moving security-critical credentials to a completely new SSI-like infrastructure is a complex issue. Identifying these open questions and requirements was greatly assisted by the fully functioning prototype, the continuous demonstration of which led to the discovery of several non-trivial perspectives. The next section analyzes all this received feedback and open questions in more detail.

9 Discussion: The Readiness of Switzerland for e-PSPs

9.1 From prototype feasibility to national readiness

Throughout the last couple chapters, this thesis has addressed the engineering challenge of modernizing personnel security clearances and found that the created e-PSP system is technologically viable. It is clear that the SWIYU ecosystem is more than capable of supporting the strict requirements of high-trust credentials.

However, as established in the previous chapter, technical feasibility is only the first step. The ultimate question of this thesis still remains to be answered: how ready is a country like Switzerland really to start using security-critical credentials built on SSI-like systems in 2026? To do that, in alignment with our design-science methodology, the focus of the thesis widens back out from the engineering artefact and analyzes both legal and operational, and then societal questions, while also briefly discussing the international landscape of openly accessible similar solutions.

9.2 Legal and operational open questions

As established earlier, PSP outcomes from a legal perspective are only recommendations, and as such, they can also be negative or positive with reservations [9]. While for the proof-of-concept, it was sufficient to model the positive outcome only, a live

national adoption could not ignore the other two categories, and a formal decision would be required on how the alternative outcomes should be handled.

The biggest blocker from this perspective is that under current regulations the subject of a background check is not always permitted to learn every detail about the outcome, including the concrete reasons that prevented them from being fully cleared. Therefore, attempting to issue a conditional or negative e-PSP directly into a user-controlled wallet, besides the formerly mentioned question of how much sense that even makes, could also directly clash with legal restrictions. If e-PSPs are intended to exclusively represent the positive outcome in a final solution too, this needs to be formally declared, otherwise adapting current regulations is likely required. A candidate for this adaptation is articles 40 and 46 of the ISG [5], which were already identified during meetings with legal experts as being incompatible with a potential live e-PSP system.

As also presented in former chapters, current verification processes are fragmented and often imperfect. However, the main reason for this is not that the processes are not defined properly, but rather that either some stakeholders simply do not want to go through the current hustle of verifications, or that there is also a level of confusion and lack of knowledge even within governmental organizations about what PSPs really are and how they are meant to be used. The use of physical notification letters as PSP credentials itself for example likely stems from the latter. Therefore, this thesis suggests that before any new solution is adopted, government personnel should be better educated and the correct use of current systems should be clarified, which on its own would likely eliminate a big portion of ad-hoc workarounds. Additional official audits may also be useful. Once the proper use of the current system is clearly understood by all stakeholders, decisions on how to modernize it can also start being made. It must be determined which parts of the legacy system should be kept, which should be eliminated, and whether SWIYU-

based e-PSPs will operate as the sole verification technology or more realistically, co-exist alongside a refined traditional system. Since co-existing is the most likely outcome, clearing up the current system first is even more critical.

A highly practical operational blocker also surfaces in one very environment that needs e-PSPs the most. While stakeholder feedback confirms the demand for robust, real-time verifications at high-security military bases, these facilities also often prohibit the use of personal smartphones within their premises. Therefore, the current reliance on mobile wallets for credential presentations creates an operational paradox where some of the most security-critical institutions are the ones unable to utilize a tool trying to enhance security. While future SWIYU updates will support offline verification capabilities, even more specialized solutions may be required to make e-PSPs usable in these environments. Since the military also has individual digitalization efforts going on on their own, represented by, for example, a wallet-like application named DIMILAR [35], it needs to be explored how all these initiatives can work together to find the best solution.

Finally, at the time of writing, the question of institutional ownership remains unresolved. While the development and maintenance of a live e-PSP system requires the allocation of budget and manpower, who would need to allocate these is not clear. In a federated organization, the priorities of different units do not necessarily align. Since the shortcomings of current verification practices are mostly experienced by the verifiers of PSPs, the issuer side is understandably less motivated to single-handedly take on the creation of a new system. Without a clear, top-down mandate defining which government unit needs to take ownership and provide the necessary resources, the interorganizational friction will likely slow down modernization efforts.

9.3 Societal readiness

An analysis of Switzerland's readiness for e-PSPs would not be complete without acknowledging that public support for digital identity solutions in the country is still far from being overwhelming. Following an earlier failed referendum regarding the development of a privately issued Swiss e-ID, the current state-operated solution was finally accepted in a second referendum in September 2025, with an extremely narrow margin of 50.39% of the votes being in favor and 49.61% against it [36].

While it is presumable that in case of a smooth, scandal-free go-live of the trust infrastructure citizens will understand the system and its advantages better, the tight vote clearly indicates that for the time being, a significant portion of the population remains skeptical. Even with targeted public education and the flawless operation of the system, overcoming public caution could easily take several years.

This societal skepticism can clearly also complicate the rollout of e-PSPs. While in standard e-government services adoption can often be voluntary, in the high-stakes realm of personnel security clearances, a fragmented utilization could be more problematic. If the government allows legacy, imperfect verification processes to indefinitely run alongside the new digital system to accommodate skeptical employees, and especially if these processes are not cleared up and educated on as recommended by this thesis, the operational vulnerabilities described in Chapter 3 will simply persist. An immediate and true modernization would therefore likely require making e-PSPs mandatory at least for certain personnel, which, however, could easily trigger pushback from the skeptical portion of the population. Resolving this friction between operational necessities and societal reality represents one of the most critical challenges Switzerland needs to address before e-PSPs can become a universal standard.

9.4 Comparable international efforts

To put this work into context, it is useful to briefly position Switzerland's e-PSP initiative against comparable international efforts. During the research phase of this thesis, an Open-Source Intelligence (OSINT) based investigation was conducted to identify whether other nations have completed any similar pilot projects or maybe even already deployed live systems that implement high-trust security clearances as SSI-like verifiable credentials.

Not entirely surprisingly, publicly available information on the topic is scarce. While there is clear international interest in the underlying concept, evidenced, for example, by Canada's User-Centric Verifiable Digital Credential Challenge in 2019 which explicitly proposed appending security clearances to government workers' digital identities [37], concrete implementations seem to be rare, and Switzerland appears to be way ahead by already possessing a functional, publicly demonstrable proof-of-concept integrated into an emerging national trust infrastructure.

While the scarcity of public information does not mean that other nations are completely ignoring the technology and internal governmental analyses or classified pilot projects may very well exist, widespread adoption is definitely lacking, which likely stems from issues and blockers similar to the ones already identified by this work. As SSI-like technologies represent a brand new paradigm and are still far from being battle-tested, governments presumably simply find less risk in relying on legacy, centralized solutions for the time being than in issuing critical clearance information directly into user-controlled wallets.

This makes Switzerland with its efforts to modernize PSPs via its SWIYU infrastructure not only an early adopter, but likely a trailblazer in assessing how state-level security vetting information and the modern principles of decentralized and self-sovereign digital identities can co-exist.

9.5 Overall readiness

Circling back to the ultimate question, the answer to how ready Switzerland is to host security-critical e-PSPs on an SSI-like infrastructure is definitely not black and white. While the technological outlook is promising with no serious blockers identified, operationally, legally, and societally, the landscape is noticeably less mature. Rethinking the technology behind the credentials representing the outcome of a nation's sensitive vetting processes requires much more than just functional software.

Ultimately, while the immediate, universal adoption of e-PSPs may be premature for 2026, the concept is undeniably viable. The future belongs to SSI-like, decentralized architectures, even if these need to prove their reliability with less sensitive, everyday use cases first. Switzerland continues to roll out credentials built on SWIYU with an electronic learner driver permit already being live [38] and the e-ID expected to be available in late 2026 [39], which are all big steps in this direction. Once the underlying infrastructure has proven its worth, the country will find itself well-equipped both technologically and societally to transition its personnel security clearances into a digital, SSI-based era.

10 Conclusion

10.1 Answers to the research questions

The main goal of this thesis was to design, implement, and evaluate a proof-of-concept of an e-PSP system built upon the SWIYU trust infrastructure. Having addressed the background of security clearances, the problems of current verification practices, and the theoretical, engineering, operational, societal, and legal challenges, we are ready to formally conclude the work by directly answering the four formerly defined research questions.

1. **How suitable is the SWIYU trust infrastructure to host an electronic Personnel Security Clearance system?** The trust infrastructure is highly suitable for this use case. Our analysis demonstrated that SWIYU is successful in finding a middle ground between pure decentralized Self-Sovereign Identity and the authoritative governance required for security-critical credentials. By relying on state-operated Base and Trust Registries, SD-JWT verifiable credentials, battle-tested cryptographic algorithms, and OpenID protocols, the system offers confidentiality, tamper-resistance, and privacy-preserving disclosures for credentials. The Token Status List mechanism elegantly solves the question of revocation, and provides cryptographically enforced, real-time status data without enabling centralized tracking. From a developer's perspective, the support for extending SWIYU is highly satisfactory. Documentations are

high-quality and helpful, the generic components do a good job in abstracting away lower-level details, and reported issues are transparently tracked and taken into consideration. Aside from maturity constraints in the current public beta, no fundamental blockers were identified that would disqualify SWIYU as a secure foundation for e-PSP credentials.

2. **What technical design, architecture, and implementation practices enable the secure issuance, lifecycle management, and verification of credentials, with a particular focus on the verifier's application-level responsibilities?** To translate PSPs into verifiable credentials, this thesis recommends isolating and thereby minimizing the data into three conceptual sets named the Identity, Authorization, and Lifecycle layer. Revocation can be managed by mapping a unique and privacy-preserving identifier to the internal id numbers used by SWIYU's Generic Issuer. The verifier component, above all, demands a rigorous, security-hardened architecture and implementation with a strict policy of enforcing everything critical on the server side. The workflow is recommended to be governed by a well-defined, fail-closed state model that actively prevents skipping any steps or reusing any state. An identity binding feature must be a core part of the verification process, ensuring the PSP is presented by its rightful holder.

3. **How viable is the completed prototype from a technological perspective and what does stakeholder feedback indicate about its operational and organizational potential?** The prototype is technologically highly viable and successfully executes the full credential lifecycle within the constraints of the SWIYU ecosystem. The intensive development phase rigorously evaluated the trust infrastructure under real-world conditions, resulting in the discovery of several implementation-level bugs that were systematically reported to and largely resolved by SWIYU developers. Demonstrations of the

prototype to stakeholders resulted in overwhelmingly encouraging feedback. The conceptual approach was well-received, confirming a practical interest and real operational potential for modernized clearance verifications.

4. **Beyond technological and operational feasibility, how ready is Switzerland to host security-critical credentials on an SSI-like infrastructure from a broader legal and societal perspective?** While technologically capable, Switzerland is not immediately ready for the universal adoption of e-PSPs and a live system still faces several blockers. These include the contradiction and legal status of issuing negative or conditional clearance outcomes into user-controlled wallets, existing confusion regarding current practices, an undefined institutional ownership, and other constraints, such as smartphone bans in security-critical zones. Societally, the outcome of the 2025 e-ID referendum indicates a high level of public skepticism toward SWIYU-based digital credentials. e-PSPs will likely need other, lower-risk credentials to increase public trust before security critical ones can also be implemented.

10.2 Main contributions of the thesis

Going beyond mere theoretical exploration, this work provided several conceptual, engineering, and analytical contributions both to Swiss national security and the emerging field of decentralized digital identities in general. These contributions are the following:

1. Operational vulnerability analysis of current PSP verification practices: The thesis collected and analyzed the vulnerabilities of the operational realities of current Swiss PSP verification practices. The risks identified act as an important justification for why modernization is necessary.

2. The three-layer e-PSP credential model: The thesis introduced a structured methodology for mapping clearances into minimal and yet functional verifiable credentials. The conceptual isolation of identity, authorization, and lifecycle layers can act as a blueprint for the digitalization of other future credentials within the ecosystem too.

3. A security-hardened reference verifier architecture: A major engineering contribution of this work is the design of a reference e-PSP verifier, including a Backend-for-Frontend Web Verifier with an explicit server-side state model and fail-closed operational principles.

4. Full-stack e-PSP proof-of-concept and contributions to the SWIYU infrastructure: The theoretical design was realized into a fully functional proof-of-concept available at `https://epsp.ch`. The intensive development and testing of this artifact served as an active battle test for the SWIYU public beta itself and resulted in documented bug reports and system improvements actively implemented by the official maintainers.

5. A socio-technical readiness assessment of Switzerland for e-PSPs: Finally, the thesis delivered a readiness analysis that widened the focus beyond sheer technology. By identifying real-world blockers and recommending mitigations, it provided a concrete list of difficulties that Switzerland must overcome before a live adoption of a SWIYU-based e-PSP system is possible.

10.3 Implications for the field

The conclusion of the thesis will now reflect on the impact of this work from an even broader perspective, touching on implications on high-trust digital credentials, software engineering, and Switzerland's positioning in the global innovation landscape.

An important implication of the research is the validation of a hybrid SSI model for high-trust use cases. By showing that decentralized architectures can also be a match with the critical risk profiles of security clearances when backed by just the right level of state-level trust, the thesis confirms that SSI-inspired models do not have to be restricted to low- or middle-stake use cases.

It also aims to preemptively challenge a dangerous possible mindset of future implementers extending Switzerland's national trust infrastructure: the assumption that their implementation is merely a thin user interface built on top of a whole and complete validation service. While the infrastructure indeed handles status checks and cryptography, the application layer still has a huge security responsibility and must therefore be recognized as a security-critical component in its own right. The mass failure to realize that could easily lead to the significant weakening of the ecosystem.

Finally, on an international scale, this research strengthens Switzerland's position as a pioneer. Even though the aforementioned question of the compatibility of SSI-like paradigms and high-trust credentials is of clear international interest, the transition from legacy, centralized practices to decentralized digital ecosystems for this specific use case is in an extremely early phase. By actively building and testing e-PSPs on its national trust infrastructure, Switzerland is moving beyond theoretical exploration and actively contributes to international efforts.

10.4 Future work

While this thesis is based on the direct pain points and feedback of stakeholders, future research should prioritize an even larger-scale and more structured approach, for example by conducting an official survey. This is recommended as a way to objectively and more precisely measure the exact extent of the operational vulnerabilities within the current system, which would likely help in urging modernization.

The findings could also help refine the recommendations given for mitigating these vulnerabilities within the current legacy system.

To overcome the non-technical blockers identified in this work, future efforts from policymakers and legal experts are also needed. These must result in the creation of formal frameworks that explicitly regulate the use of e-PSPs and should establish clear mandates that assign institutional and budgetary ownership of the e-PSP system.

As the SWIYU trust infrastructure is being developed, the future versions of the developed solution will naturally need to integrate the upcoming capabilities as well. An example in the foreseeable future is the adoption of DCQL to streamline the current multi-credential presentation workflow. Integrating upcoming tools, such as the SWIYU Check App and its offline verification capabilities, will also be essential. Even more importantly, future research addressing how the current blocker of mobile bans in high-security military zones can be resolved, is also necessary.

Finally, to actively drive the modernization of PSPs, the continuation of engagement with stakeholders is necessary. Future work needs to include additional demonstrations at conferences and governmental forums to actively showcase the potential of the technology. By keeping the conversation running, more and more thoughts can be stimulated, which can speed up and pave the way for a more secure, digital future for Switzerland's critical security clearances.

References

[1] Swiss Confederation, *Introduction - swiyu technical documentation*, 2026. Accessed: Apr. 20, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/introduction/.

[2] Swiss Confederation, *Roadmap for the swiyu Public Beta Trust Infrastructure - swiyu technical documentation*, 2026. Accessed: Apr. 20, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/roadmap/.

[3] Š. Čučko and M. Turkanović, "Decentralized and Self-Sovereign Identity: Systematic Mapping Study", *IEEE Access*, vol. 9, pp. 139 009–139 027, 2021. DOI: 10.1109/ACCESS.2021.3117588. Accessed: Apr. 20, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/9558805.

[4] Staatssekretariat für Sicherheitspolitik SEPOS, *Personensicherheitsprüfungen*, 2026. Accessed: Apr. 20, 2026. [Online]. Available: https://www.sepos.admin.ch/de/personensicherheitspruefung.

[5] Swiss Confederation, *Bundesgesetz über die Informationssicherheit beim Bund (Informationssicherheitsgesetz, ISG)*, 2022. Accessed: Apr. 20, 2026. [Online]. Available: https://www.fedlex.admin.ch/eli/cc/2022/232/de.

[6] Interne Revision VBS, *Prüfbericht "Personensicherheitsprüfungen (PSP)"*, 2025. Accessed: Apr. 22, 2026. [Online]. Available: https://www.vbs.admin.ch/dam/de/sd-web/8cJPoW1diY7u/Pruefbericht-A-2025-02-d.pdf.

[7] Swiss Confederation, *Technology Stack - swiyu technical documentation*, 2026. Accessed: Apr. 20, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/technology-stack/.

[8] Swiss Confederation, *Getting started with the swiyu Generic Verifier - swiyu technical documentation*, 2026. Accessed: Apr. 20, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/cookbooks/onboarding-generic-verifier/.

[9] Staatssekretariat für Sicherheitspolitik SEPOS, *Ablauf und Ergebnis einer Personensicherheitsprüfung*, 2026. Accessed: Apr. 20, 2026. [Online]. Available: https://www.sepos.admin.ch/de/personensicherheitspruefung-ablauf-und-ergebnis.

[10] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research", *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004. DOI: 10.2307/25148625. Accessed: Apr. 20, 2026. [Online]. Available: https://aisel.aisnet.org/misq/vol28/iss1/6/.

[11] Swiss Confederation, *Verordnung über die Personensicherheitsprüfungen (VPSP) vom 8. November 2023*, 2023. Accessed: Apr. 20, 2026. [Online]. Available: https://www.fedlex.admin.ch/eli/cc/2023/736/de.

[12] Staatssekretariat für Sicherheitspolitik SEPOS, *Prüfungsarten*, 2026. Accessed: Apr. 20, 2026. [Online]. Available: https://www.sepos.admin.ch/de/personensicherheitspruefung-pruefungsarten.

[13] Staatssekretariat für Sicherheitspolitik SEPOS, *Wer braucht eine Personensicherheitsprüfung?*, 2026. Accessed: Apr. 20, 2026. [Online]. Available: https://www.sepos.admin.ch/de/personensicherheitspruefung-personenkreise.

[14] Swiss Confederation, *Verordnung über die Militärdienstpflicht (VMDP)*, 2017. Accessed: Apr. 20, 2026. [Online]. Available: https://www.fedlex.admin.ch/eli/cc/2017/810/de.

[15] Staatssekretariat für Sicherheitspolitik SEPOS, *Personensicherheitsprüfung – Fragen und Antworten*, 2023. Accessed: Apr. 20, 2026. [Online]. Available: https://www.sepos.admin.ch/de/personensicherheitspruefung-fragen-und-antworten.

[16] Staatssekretariat für Sicherheitspolitik SEPOS, *Merkblatt Personensicherheitsprüfung gemäss ISG*, 2024. Accessed: Apr. 20, 2026. [Online]. Available: https://www.sepos.admin.ch/dam/de/sd-web/DmCKUhETIyls/Merkblatt-Personensicherheitspr%C3%BCfung-ISG-d.pdf.

[17] Watson, *Erpressbar, Drogenprobleme, Vorstrafen: Top-Beamte fliegen durch die Sicherheitsprüfung*, 2024. Accessed: Apr. 22, 2026. [Online]. Available: https://www.watson.ch/schweiz/gesellschaft-politik/502364810-armee-viele-top-beamte-fliegen-durch-die-sicherheitspruefung-des-bundes.

[18] Staatssekretariat für Sicherheitspolitik SEPOS, *Internationale Sicherheit*, 2025. Accessed: Apr. 20, 2026. [Online]. Available: https://www.sepos.admin.ch/de/internationale-sicherheit.

[19] Bundeskanzlei, *Die Personensicherheitsprüfung*, 2024. Accessed: Apr. 20, 2026. [Online]. Available: https://www.bk.admin.ch/dam/bk/de/dokumente/personal-und-ressourcen/Brosch%C3%BCre_2024_DE.pdf.download.pdf/Brosch%C3%BCre_2024_DE.pdf.

[20] World Wide Web Consortium (W3C), *Verifiable Credentials Data Model v2.0*, 2025. Accessed: Apr. 20, 2026. [Online]. Available: https://www.w3.org/TR/vc-data-model-2.0/.

[21] European Union Agency for Cybersecurity (ENISA), "Digital Identity: Leveraging the SSI Concept to Build Trust", Tech. Rep., 2022. Accessed: Apr. 20, 2026. [Online]. Available: https://www.enisa.europa.eu/sites/default/files/publications/ENISA%20Report%20-%20Digital%20Identity%20-%20Leveraging%20the%20SSI%20Concept%20to%20Build%20Trust.pdf.

[22] E. Krul, H.-y. Paik, S. Ruj, and S. S. Kanhere, "SoK: Trusting Self-Sovereign Identity", *Proceedings on Privacy Enhancing Technologies*, vol. 2024, no. 3, pp. 297–313, 2024. DOI: 10.56553/popets-2024-0079. Accessed: Apr. 20, 2026. [Online]. Available: https://doi.org/10.56553/popets-2024-0079.

[23] U.S. Government Accountability Office, *Information Security: OPM Has Implemented Many of GAO's 80 Recommendations, but Over One-Third Remain Open*, 2018. Accessed: Apr. 20, 2026. [Online]. Available: https://www.gao.gov/products/gao-19-143r.

[24] Swiss Confederation, *Open Source Components - swiyu technical documentation*, 2026. Accessed: Apr. 21, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/open-source-components/.

[25] Swiss Confederation, *Trust Protocol based on VCs - swiyu technical documentation*, 2026. Accessed: Apr. 21, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/specifications/trust-protocol/.

[26] Swiss Confederation, *Technology*, 2026. Accessed: Apr. 21, 2026. [Online]. Available: https://www.eid.admin.ch/en/technology.

[27] Swiss Confederation, *Onboarding the swiyu Base & Trust Registry - swiyu technical documentation*, 2026. Accessed: Apr. 21, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/cookbooks/onboarding-base-and-trust-registry/.

[28] Swiss Confederation, *swiyu Trust Infrastructure: Interoperability profile - swiyu technical documentation*, 2026. Accessed: Apr. 21, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/specifications/interoperability-profile/.

[29] Swiss Confederation, *Getting started with the swiyu Generic Issuer - swiyu technical documentation*, 2026. Accessed: Apr. 21, 2026. [Online]. Available: https://swiyu-admin-ch.github.io/cookbooks/onboarding-generic-issuer/.

[30] Swiss Confederation, *How to make the e-ID unlinkable*, 2025. Accessed: Apr. 22, 2026. [Online]. Available: https://www.eid.admin.ch/en/so-wird-die-e-id-unverknuepfbar-e.

[31] Bug Bounty Switzerland, *Bug Bounty Program for the Swiss Digital identity and Trust Infrastructure*, 2026. Accessed: Apr. 22, 2026. [Online]. Available: https://www.bugbounty.ch/en/swiyu-2/.

[32] Swiss Confederation, *swiyu-admin-ch/swiyu-issuer*, 2026. Accessed: Apr. 21, 2026. [Online]. Available: https://github.com/swiyu-admin-ch/swiyu-issuer.

[33] F. Egger, "Security Analysis of the Swiss e-ID & Trust Infrastructure", Master Thesis, École Polytechnique Fédérale de Lausanne, 2025. Accessed: Apr. 22, 2026. [Online]. Available: https://github.com/user-attachments/files/21157739/Security_Analysis_of_the_Swiss_e_ID___Trust_Infrastructure.pdf.

[34] Swiss Confederation, *swiyu-admin-ch/swiyu-verifier*, 2026. Accessed: Apr. 22, 2026. [Online]. Available: https://github.com/swiyu-admin-ch/swiyu-verifier.

[35] Gruppe Verteidigung, *Digitalisierung Milizarmee (DIMILAR)*, 2025. Accessed: Apr. 23, 2026. [Online]. Available: https://www.vtg.admin.ch/de/digitalisierung-milizarmee.

[36] Swiss Confederation, *e-ID law approved at the ballot box*, 2025. Accessed: Apr. 22, 2026. [Online]. Available: https://www.eid.admin.ch/en/e-id-gesetz-an-der-urne-angenommen-e.

[37] Government of Canada, *User-Centric Verifiable Digital Credential Challenge: Kick-Off Meeting*, 2020. Accessed: Apr. 22, 2026. [Online]. Available: https://canada-ca.github.io/ucvdcc/docs/UCVDCC-KICKOFF-MEETING.pdf.

[38] Swiss Confederation, *Electronic learner driver permit (eLDP)*, 2026. Accessed: Apr. 22, 2026. [Online]. Available: https://www.eid.admin.ch/en/pilot-projects.

[39] Swiss Confederation, *Strengthening acceptance of e-ID with additional measures*, 2026. Accessed: Apr. 22, 2026. [Online]. Available: https://www.eid.admin.ch/en/akzeptanz-der-e-id-mit-zusaetzlichen-massnahmen-staerken-e.

Appendix A Declaration on the Use of Generative AI

Through the course of this research generative AI (the latest respective available model of ChatGPT and Gemini at any given time) was used to assist the author in the creation of this thesis document. The primary purpose of this use was to improve the general wording and quality of the text. The tasks mainly included formulating personal, raw, often multilingual notes and ideas into high-quality texts, looking for grammar mistakes and typos in existing texts, and asking for edits, among others, for an appropriate academic style. The use of generative AI tools did not exceed reasonable ethical practices and all the intellectual content within this document is based on the author's own work and research conducted as part of an internship project.