EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL

Ensuring High Assurance of EUDI Wallets
& the NL National eID Scheme



NL-WI

NL EUDI Wallet Certification System

EN 17640 FITCEM PROTECTION PROFILE for EUDI Wallet Instance in a remote HSM architecture

RC1.1

2026.10.27

Document reference	NL-WI
Version	RC1.1
Status	Release Candidate
Date	27/10/2026
Document type	Certification Scheme EN 17640 FITCEM PROTECTION PROFILE for EUDI Wallet Instance in a remote HSM architecture
Parent framework	NL-CS (parts1/2/3/4)
Scheme Owner	Kingdom of the Netherlands Minister of Economic Affairs
Certification target	European Digital Identity Wallets
Primary audience	Scheme Owners, Evaluation Laboratories; CABs, EUDI Wallet providers
Secondary audience	EUDI Wallet ecosystem stakeholders
Assurance level <identity proofing CIR 2015/1502> and <evaluation, EU Regulation 2019/881>	High/High
Related Documents	NL-WI-conditions NL-WSCA (-conditions)

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Version	Change	State	Date
RC1.0	Result of European Collaboration / Discussions	Release Candidate	2026-10-27
RC1.1	Release Candidate for publication following review round NL Certification System Mngt Board	Release Candidate	2026-10-31

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Background

This Protection Profile (PP) is developed based on Regulation (EU) No 910/2014, as amended by Regulation (EU) 2024/1183 (eIDAS 2.0), and Commission Implementing Regulation (EU) 2024/2981.

This PP defines the **Wallet Instance (WI)** as the Target of Evaluation WI-(TOE) for the **Remote HSM configuration** of a European Digital Identity Wallet. In this configuration, the Wallet Instance operates on the user device and interacts with the WSCA running on Wallet Provider backend infrastructure, which in turn uses a certified remote HSM as the WSCD. The Wallet Instance provides security-relevant functionality including user authentication and authorization, protection of Wallet Unit data and device-resident security assets, and secure interaction with the WSCA/WSCD for security-sensitive operations.

This PP specifies the functional and assurance security requirements for the Wallet Instance within the **FITCEM** certification framework. It is intended to provide a common security baseline for evaluating the Wallet Instance as part of an EUDI Wallet using the Remote HSM architecture, including its security dependencies and interactions with the WSCA/WSCD.

Terminology

This Protection Profile uses the terminology defined in the **European Digital Identity Wallet Architecture and Reference Framework (ARF)** and the applicable EU regulations. Unless otherwise stated, terms such as **Wallet Unit, Wallet Instance, Wallet Secure Cryptographic Application (WSCA), Wallet Secure Cryptographic Device (WSCD), Wallet Provider, Wallet User, Wallet Instance Attestation (WIA), Key Attestation (KA),** and **critical assets** shall have the meaning defined in the ARF and the applicable Commission Implementing Regulations.

Where this Protection Profile introduces terms specific to the **Remote HSM architecture**, these terms are explicitly defined in this Protection Profile. In case of ambiguity, the definitions in the applicable EU legislation take precedence over terminology introduced in this Protection Profile. The following terms defining the secure authentication protocol between the Wallet Instance and the WSCA/WSCD are specific to the **Remote HSM architecture**:

- **Remote User Authentication Protocol (RUAP):** The protocol enforces that every PIN verification is performed through a live and fresh interaction between the Wallet Instance and the WSCA. This ensures that the confidentiality of the PIN is preserved, while verification is subject to controlled and rate limited processing at the WSCA. The Wallet Instance must demonstrate the User Authentication Evidence (UAE), which is derived both from a knowledge-based factor (derived from the PIN) and a possession-based factor. The User Authentication Evidence (UAE) is verified against the Authentication Reference Data (ARD) which is shared during enrolment.
- **Authentication Reference Data (ARD):** Authentication Reference Data is shared with the WSCA/WSCD during enrolment and used at the WSCA/WSCD to verify the Authentication Reference Data (UAE).
- **User Authentication Evidence (UAE)** Cryptographic evidence bound to a user authentication event at the wallet instance, provided over the Remote Authentication Channel and verified by the WSCA before it performs a wallet cryptographic operation.
-

Relationship to Other Standards and Schemes

This Protection Profile is intended to be used within the legal, regulatory, certification and technical framework established by the following documents and standards.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

- **eIDAS 2.0:** Regulation (EU) No 910/2014, as amended by Regulation (EU) 2024/1183, establishing the European Digital Identity Framework.
- **EUDI Wallet Certification:** Commission Implementing Regulation (EU) 2024/2981, laying down rules for the certification of European Digital Identity Wallets.
- **EUDI Wallet Core Functionality and Integrity:** Commission Implementing Regulation (EU) 2024/2979.
- **EUDI Wallet Protocols and Interfaces:** Commission Implementing Regulation (EU) 2024/2982.
- **PID and EAA:** Commission Implementing Regulation (EU) 2024/2977, laying down rules concerning person identification data and electronic attestations of attributes.
- **Applicable Standards and Specifications:** Commission Implementing Regulation (EU) **2026/1731**, amending Regulations (EU) 2024/2977, 2024/2979, 2024/2980 and 2024/2982 with updated standards and specifications.
- **Wallet-Relying Parties:** Commission Implementing Regulation (EU) 2025/848 concerning the registration of wallet-relying parties and their access certificates.
- **Certified Wallets:** Commission Implementing Regulation (EU) 2025/849 concerning the submission of information to the Commission and the Cooperation Group for the list of certified EUDI Wallets.
- **Assurance Levels for Electronic Identification:** Commission Implementing Regulation (EU) 2015/1502.
- **EUDI Wallet Architecture and Reference Framework (ARF):** The current applicable version of the ARF, including its requirements, definitions, technical specifications, rulebooks and certification requirements. The ARF provides the technical architecture and reference specifications supporting the EUDI Wallet ecosystem.
- **FITCEM: EN 17640:2022**, *Fixed-time cybersecurity evaluation methodology for ICT products*, October 2022. This standard defines the fixed-time cybersecurity evaluation methodology applicable to the evaluation of the Wallet Instance TOE under the FITCEM certification framework.
- **ECCG Agreed Cryptographic Mechanisms (ACM):** The current applicable version.
- **Cryptographic and PKI standards:** Applicable ETSI, ISO/IEC and IETF standards referenced by the ARF and the applicable implementing regulations.
- **EUDI Oracle:** Yivi, EUDI Oracle – Knowledge Graph Visualization, available at: https://eudi-oracle.yivi.eu/

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Table of Contents

1 Introduction 7
1.1 Protection Profile Reference Identification 7
1.2 TOE Overview 7
1.2.1 The WI- (TOE) in the Wallet ecosystem 7
1.2.2 Wallet Unit components 8
1.2.3 Wallet ecosystem entities 9
1.2.4 Principal security interactions 9
1.2.5 The WI-(TOE) User Device Protection and Self Protection 10
1.3 Wallet Unit Lifecycle and Cryptographic Trust Relationships 12
1.3.1 Wallet Instance Installation 12
1.3.2 Wallet Activation 12
1.3.3 PID/Attestation Provider authentication and Issuance. 16
1.3.4 Credential (PIDs/Attestation) Disclosure 17
1.3.5 Revocation/Deletion 19
1.3.6 Additional Wallet Instance Security Functionality 19
2 Security Problem Definition 21
2.1 Assets 21
2.2 Threat model 22
2.3 Attack scenarios 23
2.4 User and External Entities 24
2.5 Environmental Security Objective 25
2.6 Coverage of the Security Requirements 28
2.6.1 Coverage of the Life Cycle 28
2.6.2 Coverage of the Threats 28
2.6.3 Coverage of all requirements 30
3 Detailed Security Requirements 31
3.1 WI-(TOE) Functional Security Requirements 31
3.2 WI-(TOE) Resilience Security Requirements 41
A Requirement Coverage Rationale 48
A.1 Coverage of all FSR through the life cycle 48
A.2 Coverage of the attack scenarios 49
B Modular and functional Decomposition of the WI-(TOE) 59
B.1 Functional decomposition 59
B.2 Modular decomposition 60
C OWASP MAS mapping to the Security Requirements 66
C.1 Functional security requirement mapping 66
C.2 Resilience security requirement mapping 66

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

1 Introduction

1.1 Protection Profile Reference Identification

Title: EN 17640 FITCEM PROTECTION PROFILE for EUDI Wallet Instance in the remote HSM architecture

PP Identification: FITCEM-PP-WALLETINSTANCE-REMOTE

FITCEM identification: FITCEM: EN 17640:2022, Fixed-time cybersecurity evaluation methodology for ICT products, October 2022.

Version: 0.7

Date: Tuesday, 27 October 2026

Status: Non-Certified

Authors: SGS Brightsight (ITSEF and CB)

1.2 TOE Overview

The Target of Evaluation (TOE) is a European Digital Identity Wallet Instance (here after is abbreviated by WI-(TOE)) is a mobile application designed to give users full control over how they manage and share their personal identity. The WI-(TOE) securely stores official identification data and supporting credentials from authorized PID /Attestation Providers. It lets individuals present these credentials to verified service providers for both online interactions and in-person checks using common standards. By supporting selective disclosure, the app ensures people can share only the specific information required for a service without exposing extraneous details, all while maintaining privacy and offering a straightforward way to sign documents digitally.

The TOE is designed exclusively for a **remote HSM configuration**, in which the security-critical cryptographic assets and operations are managed by a remotely deployed Wallet Secure Cryptographic Device (WSCD) accessed through the Wallet Secure Cryptographic Application (WSCA).

1.2.1 The WI- (TOE) in the Wallet ecosystem

The Wallet Instance, WI-(TOE), is one of the components in the **Wallet Unit**, which acts as part of the **Wallet ecosystem**. In ***Figure 1*** the main components of the Wallet Unit and wallet ecosystem are depicted.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

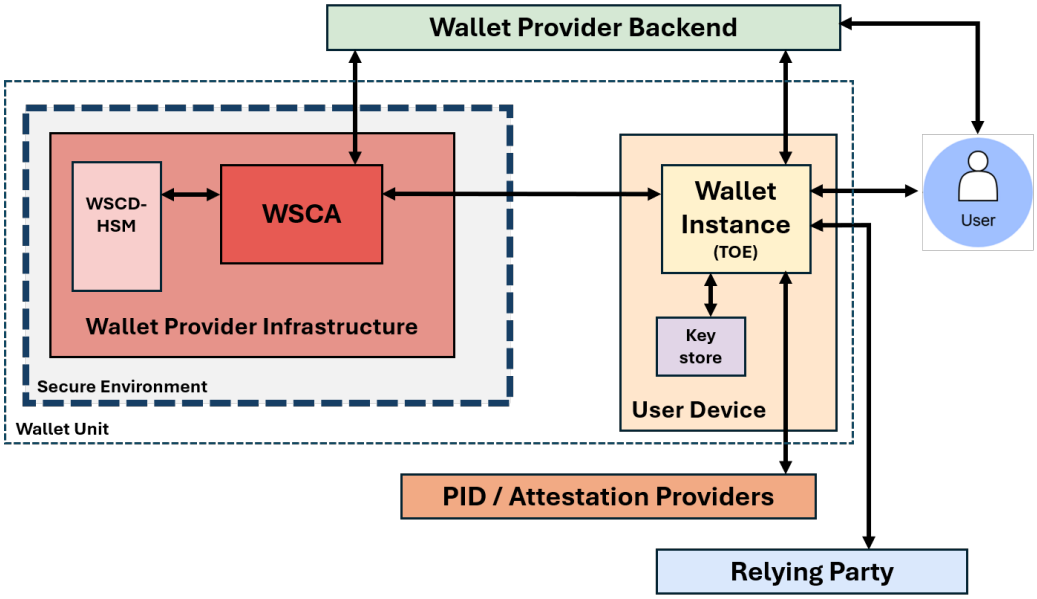


Figure 1 — Wallet Unit as part of the Wallet Ecosystem

1.2.2 Wallet Unit components

The following components are the core components in the **Wallet Unit**, as depicted in ***Figure 1***:

- **User device (UD)**: The hardware, operating system and software environment on which the Wallet Instance executes. The User Device provides the execution environment and local security mechanisms used to protect the Wallet Instance, User authentication and local data. Its security posture, including the integrity of the operating system and relevant security mechanisms, contributes to the overall security of the Wallet Unit.
- **Wallet Instance (WI-(TOE))**: The mobile application installed on the User Device. The WI-(TOE) authenticates ecosystem entities, obtains User approval for security-sensitive actions, manages credentials and cryptographic assets, and initiates cryptographic operations. It shall protect sensitive Wallet Unit information and ensure that security-sensitive operations are performed only under the applicable authorization and User-control conditions. It interfaces with one or more key stores for the management of non-critical cryptographic assets.
- **Wallet Secure Cryptographic Application (WSCA)**: A WSCA implementation deployed on a hardened backend server. It provides security-critical cryptographic functionality and interfaces with the WSCD. The WSCA enforces the applicable security controls for cryptographic operations and ensures that operations involving critical assets are performed only after the required authorization and User authentication conditions have been satisfied.
- **Wallet Secure Cryptographic Device (WSCD)**: A hardware security device used by the WSCA to protect critical cryptographic assets and perform security-sensitive cryptographic operations. The WSCD shall protect the confidentiality and integrity of protected private keys and shall prevent unauthorized export or use of protected private keys.
- **Keystore User Device (Keystore)**: An applicable hardware-backed SE used to generate, store and use cryptographic assets that are not required to be protected by the WSCD. The possession factor used by the authentication protocol is managed in the

keystore. The keystore shall provide the security level appropriate to the assets it protects and prevent unauthorized access to the associated private keys.

1.2.3 Wallet ecosystem entities

The following entities (depicted in Figure ***Figure 1***) are part **Wallet Ecosystem** and interact with the Wallet Unit:

- **Wallet Provider Backend (WPB):** Provides Wallet Provider services, including Wallet Unit onboarding, activation, management, maintenance and security-status related services. The Wallet Provider is responsible for establishing the required trust relationship with the Wallet Instance, issuing the WIA and KA, and taking appropriate action when the security or trustworthiness of a Wallet Unit is compromised.
- **PID and Attestation Providers:** Providers that issue PIDs and Attestations to the Wallet Unit. Before issuance, the Provider authenticates the Wallet Unit and verifies the applicable WIA and KA, including their validity and revocation status. For device-bound credentials, the Provider verifies proof-of-possession of the private key corresponding to the public key contained in the KA and binds the issued credential to that key.
- **Relying Parties (RPs):** Entities that request and verify PID or Attestation attributes from the Wallet Unit to provide a service to the User. The RP authenticates to the Wallet Unit using the applicable Access Certificate and provides the information required for the User to understand and approve the requested disclosure. The RP is responsible for verifying the authenticity, integrity, freshness and applicable cryptographic binding of the received presentation.
- **Access Certificate Infrastructure**: Provides Access Certificates used to authenticate relevant ecosystem entities. The associated trust infrastructure provides the trust anchors required by the Wallet Unit and other ecosystem participants to verify the authenticity of Access Certificates.

1.2.4 Principal security interactions

The principal high level data flows and trust relationships are as follows between the entities:

- **Wallet Instance (WI-(TOE)) – WSCA**: The Wallet Instance communicates with the WSCA through a protected channel. User authentication and the Remote User Authentication Protocol (RUAP), including User Authentication Evidence (UAE) and Authentication Reference Data (ARD), are used to authorize security-sensitive operations. The WSCA invokes the WSCD for operations involving protected cryptographic assets. Per-key authentication shall be implemented at the WSCD and the per-key authorization data shall secure stored by the WI-(TOE).
- **Wallet Instance (WI-(TOE)) – PID/Attestation Providers:** The Wallet Instance communicates with the Provider to request and receive PIDs or Attestations. Where required, the Wallet Unit provides the applicable WIA and KA. The Provider verifies these objects and may request proof-of-possession of the corresponding private keys before issuing a device-bound credential.
- **Wallet Instance (WI-(TOE)) – Relying Party:** The Wallet Instance participates in credential presentation using the applicable presentation protocol. It authenticates the Relying Party, evaluates the presentation request, obtains explicit User approval and performs selective disclosure. For device-bound credentials, the Wallet Unit uses the applicable WSCA/WSCD to perform the required cryptographic operation and proof-of-possession.
- **Wallet Instance (WI-(TOE)) - Keystore User Device:** The Wallet Instance uses applicable keystores to generate, store and use cryptographic assets in accordance with their required security level. The Wallet Instance shall ensure that operations involving

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

protected assets are subject to the applicable User authentication and authorization controls.

- **Wallet Instance (WI-(TOE)) - Wallet Provider backend (WPB):** The Wallet Instance communicates with the Wallet Provider for installation, onboarding, activation, Wallet Unit management and security-status related services. During onboarding and activation, the Wallet Provider receives the attested keys and the WIA key material, as applicable, and securely shares WIA and KA with the WI-(TOE).

The security relationships described above establish the principal boundaries of the Wallet Unit and its interactions with the EUDI Wallet ecosystem. Section 1.3 describes the Wallet Unit lifecycle and the cryptographic trust relationship between the Wallet Instance, WIA, KA, WSCA/WSCD, PIDs, Attestations and Relying Parties. It provides the basis for the subsequent Security Problem Definition and security requirements. The complete set of Functional Security Requirements (FSRs) applicable to the WI-(TOE) is specified in section 3.1.

1.2.5 The WI-(TOE) User Device Protection and Self Protection

The WI-(TOE) is executed on top of a mobile user device, and therefore strongly depend for its security operation on the security measures from the **mobile operating system** and the **mobile hardware**.

The WI-(TOE) however shall properly implement this **self-protection** mechanisms and make appropriate use of the security features provided by the mobile operating system and underlying mobile hardware. These requirements are explicitly defined as part of the **Resilience Security Requirements** (RSRs) in section 3.2. In this section a summary of the expected security measures is provided.

The WI-(TOE) shall securely utilize the following **mobile operating system level** protections mechanisms:

- **Application sandboxing**: isolates the Wallet Instance and its data from other applications.
- **Access control and permissions**: controls access to sensitive device resources such as NFC, camera, biometrics, and storage.
- **Secure storage APIs**: provides protected storage for sensitive wallet data and cryptographic material.
- **Keystore/key-management services**: allows the wallet to generate and use cryptographic keys without exposing private key material directly to the application.
- **User authentication**: enables wallet operations or key usage to be gated by a device PIN, password, or biometric authentication.
- **Application integrity and signing**: ensures that installed applications originate from an unmodified signed package.
- **Secure communication interfaces**: provides OS-controlled networking and IPC mechanisms that limit unauthorized access between applications and services.

The WI-TOE shall securely utilize the security services provided by the platform's **Trusted Execution Environment (TEE)** and/or **Secure Element (SE)**. These trusted components shall provide the following security mechanisms:

- **Non-exportable hardware-backed key generation**, ensuring that cryptographic keys are generated and remain protected within the TEE or SE and cannot be exported in plaintext form.
- **Protected cryptographic operations**, whereby sensitive cryptographic operations are performed within the trusted environment, without exposing private key material to the Rich Execution Environment (REE) or to the Wallet Instance.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

- **Hardware-enforced key usage policies**, restricting the use of protected keys to authorized algorithms, operations, and purposes.
- **Resistance to key extraction**, achieved through hardware and logical isolation mechanisms that provide significantly stronger protection against key disclosure than application-level storage.
- **Key and device attestation**, enabling the platform to provide cryptographic evidence regarding the characteristics of the protected keys, the integrity of the device, and the security properties of the TEE or SE in which the keys are protected.

The WI-(TOE) shall implement self-protection mechanisms to maintain its confidentiality, integrity, authenticity and privacy when operating in a potentially hostile mobile environment. These mechanisms shall include:

- **Security monitoring and reporting**, including the detection and reporting to the Wallet Provider of tampering attempts and insecure platform states.
- **Protection against debugging**, including the absence of production debug functionality and debug artefacts, and the detection of attached debuggers.
- **Platform-state verification**, including the detection of rooted, jailbroken, emulated, virtualised, developer/debug-mode or otherwise insecure operating environments, and the disabling of security-critical functionality when such a state is detected.
- **Secure use of platform security mechanisms**, including application isolation, mandatory access controls and hardware-backed security services.
- **Protection of sensitive assets**, including secure key management, approved cryptographic services, secure random number generation, encrypted storage, secure transmission and secure memory clearing.
- **Secure application design and implementation**, following secure coding practices and protecting against common mobile application security threats.
- **User interface and sensitive-operation protection**, including protection of PIN entry and sensitive screens against screen capture, overlay and tapjacking attacks, accessibility-service abuse, malware-assisted attacks and unauthorised exposure of sensitive information.
- **Privacy protection**, ensuring that personal data is processed in a transparent and privacy-preserving manner and that user tracking is prevented.

In addition, the WI-(TOE) may implement the following best-practice hardening mechanisms. Where implemented, they shall meet the corresponding Resilience Security Requirements:

- **Code and resource obfuscation**, including anti-deobfuscation measures, to hinder reverse engineering of security-sensitive functionality.
- **Protection against runtime instrumentation**, including the detection of or resistance to code injection, function hooking, API interception and dynamic analysis.
- **TOE integrity verification**, including periodic verification of the integrity and authenticity of the TOE's code and security-relevant resources.
- **Device attestation**, providing authenticated and fresh evidence of the security state of the User Device.
- **Application-level payload encryption**, in addition to transport-layer protection.
- **Protection against loss of user-interface focus** during security-sensitive interactions.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

1.3 Wallet Unit Lifecycle and Cryptographic Trust Relationships

The WI-(TOE) forms part of a Wallet Unit within the EUDI Wallet ecosystem. This section provides context for the security and trust relationships established during the Wallet Unit lifecycle.

It describes how the Wallet Instance interacts with the Wallet Provider, WSCA/WSCD, keystores, PID and Attestation Providers, and Relying Parties, and how cryptographic mechanisms are used to establish authenticity, integrity, confidentiality and cryptographic binding where required.

This section distinguishes between trust established through cryptographic mechanisms and trust established through certification, registration, trusted lists and other ecosystem governance mechanisms.

1.3.1 Wallet Instance Installation

The Wallet Instance should preferably be distributed through an official application distribution channel of the User Device operating system. If the Wallet Instance is distributed through another mechanism, the Wallet Provider shall provide a mechanism enabling the User to verify the authenticity and integrity of the Wallet Instance before installation.

1.3.1.1 Mutual Authentication Channel

After installation, the Wallet Instance establishes the required secure communication channels with:

- the Wallet Provider; and
- the WSCA/WSCD

Communication between the Wallet Instance and Wallet Provider shall provide mutual authentication and shall provide confidentiality and integrity protection where required.

1.3.1.2 Initial Data Requested by Wallet Provider

The WI-(TOE) will now send its initial data to the Wallet Provider which contains technical data from the Wallet Instance, including cryptographic and hardware capabilities (such as WSCD and keystore details), system diagnostics (including OS, SDK, and library versions, plus information about available communication interfaces such as BLE and NFC), and security posture signals such as rooting, emulator, and malware detection.

The Wallet Provider shall limit collected device and security-posture information to the data necessary for Wallet Solution deployment, maintenance and security-posture monitoring. Collection shall be minimised and performed with appropriate privacy safeguards.

1.3.2 Wallet Activation

For Wallet Activation, the following steps are performed:

- Creation of a Wallet Provider user account and association of the Wallet Instance with that account (further detailed in section 1.3.2.1).
- Registration of the User authentication methods that will be used to access and operate the wallet (further detailed in section 1.3.2.2).
- Initial Generation and issuance of the WIA and KA (further detailed in section 1.3.2.3)

During activation, the Wallet Provider establishes the required trust relationship with the Wallet Instance and provides the WIA and KA required for subsequent Wallet Unit operations. The WIA provides information about the Wallet Instance and Wallet Solution, while the KA provides information about the WSCA/WSCD and the public keys used by the Wallet Unit.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

The User authentication mechanisms established during activation are subsequently used to control access to the Wallet Unit and, where applicable, to protect security-sensitive cryptographic operations involving critical Wallet Unit assets.

The following sections describe the individual activation steps and the associated cryptographic assets.

1.3.2.1 Wallet Provider User account

The User must register with the Wallet Provider to create a Wallet Provider User account. The wallet provider must associate the User account with the installed Wallet Instance.

The Wallet Provider associates the Wallet Unit with a User account to support Wallet Unit management, including User-requested revocation. The account does not necessarily need to contain the User's real-world identity unless required for a specific Wallet function or legal obligation.

1.3.2.2 User Authentications

The User enrols two user authentication mechanisms to the Wallet Instance. These two User authentication mechanisms are:

OS-level User authentication:

- The Wallet Instance shall require OS-level User authentication whenever the Wallet Instance is opened and before performing Wallet Unit operations. The OS-level authentication mechanism shall use possession of the User Device together with a knowledge-based or inherence-based factor.
- This local OS-based authentication shall be enforced at startup of the WI-(TOE) and **before security-sensitive actions**, such as
  - generating and presenting pseudonyms,
  - accessing the transaction log via the dashboard,
  - requesting the erasure of personal data by a Relying Party;
  - other operations as specified by the applicable ARF requirements.

WSCA/WSCD-level Remote User Authentication Protocol (RUAP):

- This specific **Remote User Authentication Protocol (RUAP),** including the use of User **Authentication Evidence (UAE)** and **Authentication Reference Data (ARD),** is a PP-specific mechanism defined to implement the required WSCA/WSCD-level User authentication.
- The RUAP shall support **User authentication** meeting the applicable requirements for **Level of Assurance High** as specified in Commission Implementing Regulation (EU) 2015/1502. In particular, the authentication mechanism shall use at least two authentication factors from different categories and shall implement security controls such that guessing, eavesdropping, replay and manipulation of the authentication communication cannot, with high attack potential, subvert the authentication mechanism.
- The Wallet Instance WI-(TOE) must demonstrate the User Authentication Evidence (UAE), which is derived both from a **knowledge-based authentication factor** (derived from the PIN) and a **possession-based authentication factor**, which are jointly validated to achieve two-factor authentication and to prevent manipulation or forged attempts.
- Remote User Authentication Protocol (RUAP) ensure PIN Verification and ensure that the User has **sole control** over the critical assets located in the WSCD, such that these assets cannot be used without the User's involvement through the required authentication factors;

- The protocol protects the confidentiality and integrity of the authentication data during PIN verification and enrolment, and provides protection against **duplication, tampering, eavesdropping and replay attacks**, including through freshness and challenge binding;
- Each PIN verification is performed through a live and fresh interaction between the WI-(TOE) and the WSCA/WSCD, preventing reuse of previously exchanged authentication information;
- During enrolment, the Wallet Instance WI-(TOE) ensures that both the knowledge and possession factors are securely initialized and cryptographically bound to the **Authentication Reference Data (ARD)**. Authentication Reference Data (ARD) is derived and is shared with the WSCA /WSCD during initial enrolment. This binding preserves confidentiality and integrity of the knowledge-based authentication factor (derived from the PIN) and a possession-based authentication factor and prevents compromise also if the confidentiality of the Authentication Reference Data (ARD) is not preserved.
- A **Personal Unblocking Key (PUK)** can be integrated in the protocol to reset the knowledge-based factor in case the PIN was forgotten.
- Per-key authentication data shall be sent to the Wallet Instance WI-(TOE) and be provided at every security-sensitive cryptographic operation of the applicable key, to ensure sole control of the Wallet User before the execution by the WSCD.
- This authentication mechanism is subsequently required whenever the Wallet Unit performs cryptographic operations involving protected assets within the WSCA/WSCD, ensuring that all such operations remain securely bound to both the user and the user device. At a minimum, in the following situations. WSCA/WSCD-level User Authentication is required:
  - Initial Generation of KAs and WIA (further detailed in section 1.3.2.3).
  - The Wallet Unit asks the User for approval of selective disclosure to a Relying Party (further detailed in section 1.3.4.2)
  - The User deletes a PID or attestation in their Wallet Unit (further detailed in section 1.3.4.3).
  - The User instructs the Wallet to request the issuance of a new PID or Attestation.

1.3.2.3 Initial Generation of the WIA and KA

During Wallet Activation, at least one set of Wallet Unit Attestations (WUA) is issued. These attestations comprise the Key Attestation(s) (KA) and the Wallet Instance Attestation (WIA).

The KA provides information about the WSCA/WSCD and contains the cryptographic public keys of the attested key pairs. The corresponding private keys are protected by the WSCA/WSCD. The WIA provides information about the Wallet Instance and Wallet Solution, including applicable certification information. The WIA device-binding key pair may optionally be generated within the WSCA/WSCD or managed within the context of the Wallet Instance.

The WIA and KA are issued and signed by the Wallet Provider after the Wallet Provider has performed the applicable verification of the Wallet Instance and cryptographic environment.

1.3.2.3.1 Key attestations (KA)

The User must perform WSCA/WSCD-level user authentication through the TOE to request the WSCA/WSCD to generate the attested keys for the Key Attestation (KA). The Key Attestation (KA) is a signed information object, generated and signed using the Wallet Provider signing key (WP_SIG_PRIV1), and contains:

- One or more attested public keys (PID/ATT_PUB_KEYs)
- Applicable certification and security information concerning the WSCA/WSCD.

The attested public keys (PID/ATT_PUB_KEYs) shall be securely transferred from the WSCA/WSCD to the Wallet Provider during KA issuance.

Key Attestation (KA) has the following **purposes**:

- The Wallet Unit shares the KA with the PID and Attestation Providers during issuance of a device-bound PID or Attestation.
- PID Provider or an Attestation Provider verify the signature using the WP_SIG_PUB1, verified through a certificate chain, where the Provider's certificate is validated through the Access Certificate Authority (ACA) up to a trusted anchor provisioned by the relevant EUDI Wallet trust framework.
- PID Provider or an Attestation Provider will use the Applicable certification and security information of the WSCA/WSCD, to verify if the WSCA/WSCD is fit to protect the private keys for a PID or Attestation or whether it is revoked.
- PID Provider or an Attestation Provider can request a proof-of-possession, by sending a nonce via the wallet instance to the WSCA/WSCD to be signed with private keys (PID/ATT_PRIV_KEYs) corresponding to PID/ATT_PUB_KEYs. This proves that the (PID/ATT_PRIV_KEYs) corresponding to PID/ATT_PUB_KEYs are generated by and stored in the WSCA/WSCD.
- PID Provider or an Attestation Provider will include the PID/ATT_PUB_KEYs the issued PID or Attestation.

1.3.2.3.2 Wallet Instance Attestation (WIA)

The Wallet Instance Attestation (WIA) is a signed information object, signed by the Wallet Provider Key (WP_SIG_PRIV2) and contains:

- Wallet Information: Wallet Solution ID/Name, version and URIs with Certification information and additional Wallet Solution information.
- Expiration and revocation status. The technical validity period of a WIA shall be less than 24 hours.
- WIA public key: WIA_KEY_PUB.

The WIA device-binding key pair (WIA_KEY_PRIV and WIA_KEY_PUB) may be managed either by the WSCA/WSCD or by the Instance. Where managed by the Wallet instance, the private key shall be adequately protected against unauthorised access, disclosure, modification, and use. Where managed by the WSCA/WSCD, the key pair is subject to the security controls and protection mechanisms of the WSCA/WSCD.

The WIA device-binding public key WIA_KEY_PUB shall be securely transferred from the Wallet Instance or from the WSCA/WSCD to the Wallet Provider during WIA issuance.

Key Attestation has the following **purposes**:

- The Wallet Unit shared Wallet Instance Attestation (WIA) with the PID and Attestation Providers.
- PID Provider or an Attestation Provider verify the signature using the WP_SIG_PUB2, verified through a certificate chain, where the Provider's certificate is validated through the Access Certificate Authority (ACA) up to a trusted anchor provisioned by the relevant EUDI Wallet trust framework.
- Sent to PID and Attestation Providers to inform them about the Wallet Instance details, such as Wallet Solution name, version, and certification of the Wallet Solution. This allows the PID or Attestation Provider to verify information about the Wallet Solution and Wallet Instance and to determine whether the associated Wallet Instance is currently revoked. The Wallet Provider performs the applicable integrity verification

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

before issuing the WIA. The verify if the TOE belongs to a certified Wallet Solution registered in the Wallet Provider Trusted List.
- PID Provider or an Attestation Provider will use the Wallet Identifiers to verify if the TOE belongs to a certified Wallet Solution registered in the Wallet Provider Trusted List.
- PID Provider or an Attestation Provider uses the revocation reference allowing a PID Provider or an Attestation Provider to verify if the Wallet Solution has been revoked.
- PID Provider or an Attestation Provider can request a proof-of-possession, by sending a nonce via the wallet instance to the WSCA/WSCD to be signed with private keys (WIA_KEY_PRIV) corresponding to WIA_KEY_PUB. This proves that the (WIA_KEY_PRIV) corresponding to WIA_KEY_PUB is managed by the Wallet Unit.

1.3.3 PID/Attestation Provider authentication and Issuance.

Before the PID/Attestation Provider issues the PID or the Attestation to the Wallet Unit, they will first authenticate to each other.

1.3.3.1 PID/Attestation Provider authentication to the Wallet Unit

The first step in the issuance process is the authentication of the Provider by the Wallet Instance.

During Provider authentication the TOE executes the following steps:

- TOE initiates connection with the PID/Attestation Provider.
- The Provider returns to the TOE its Access Certificate (ISS_ACC_CERT) and, where applicable, a Registration Certificate (ISS_REG_CERT).
- The TOE verifies the certificates using the applicable trusted anchor.
- The TOE verifies the Provider's registered identity, role and, where applicable, the credential types or services for which the Provider is registered. This information is obtained from the Registration Certificate or the applicable Registrar service.

1.3.3.2 Wallet Unit authentication to the PID/Attestation Provider

The next step in the Issuance process is for the TOE to send the PID/attestation request (OpenID4VCI -Credential Request) to the Provider, and for the Provider to authenticate the WUAs (KA & WIA).

During WIA authentication the following steps are executed:

- TOE sends the signed WIA to the Provider.
- The Provider verifies the WIA signature using the applicable Wallet Provider public key (WP_SIG_PUB2). Trust in the Provider's public key shall be established through the applicable Trust List or LoTE.
- The Provider verifies the validity and revocation status of the WIA.
- The Provider requests proof-of-possession from the Wallet Unit of WIA_KEY_PRIV corresponding to WIA_KEY_PUB contained in the WIA. The Provider provides a nonce to the TOE. The TOE signs the nonce using WIA_KEY_PRIV, either directly or through the WSCA/WSCD where the key is stored there. The signed nonce is returned to the Provider and verified using WIA_KEY_PUB.

During KAs authentication the following steps are executed:

- TOE sends a signed KA(s) to the Provider.
- The Provider verifies the signatures using the Wallet Provider public (WP_SIG_PUB1) key. Trust in the Provider's public key shall be established through the applicable Trust List or LoTE.
- The Provider verifies the validity and revocation status of the KA.
- The Provider requires proof-of-possession from the Wallet Unit, of the private counterpart (PID/ATT_PRIV_KEY) of the public key in the KA (PID/ATT_PUB_KEYs), as

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

defined in OpenID4VCI. For this, the Provider provides a nonce to the TOE. The TOE will request the WSCA/WSCD to sign this nonce with the PID/ATT_PRIV_KEY. The signed nonce is returned to the Provider and the Provider will verify the signature with PID/ATT_PUB_KEY from the KAs.

- To avoid a second request from the TOE to the WSCA/WSCD, If the TOE generates a new KA during PID/Attestation issuance the nonce can be included in the KA, and the KA is additionally signed by the PID/ATT_PRIV_KEYs matching the public keys in the KA (PID/ATT_PUB_KEYs).

The TOE sends a KA to a PID Provider or Attestation Provider only for device-bound, not for non-device-bound PIDs/Attestations.

1.3.3.3 PID/Attestation Generation

Now that the Provider has ensured that it is interacting with the Wallet Unit holding the private key corresponding to the public key embedded in the KA, has verified the applicable WSCA/WSCD security properties, and has verified the Wallet Instance through the WIA, it can generate the PID or Attestation.

For a device-bound PID or Attestation, the Provider binds the issued credential to one of the public keys contained in the applicable KA. The issued PID/Attestation is signed using the Provider's private signing key (ISS_SIG_PRIV) and is sent to the TOE using the OpenID4VCI Credential Response.

The exact cryptographic structure of the issued PID or Attestation depends on the credential format and the applicable technical specification.

The Wallet Unit supports, where applicable:

- **SD-JWT VC** for remote presentation to Relying Parties.
- **ISO/IEC 18013-5 mdoc** for proximity presentation.

Both formats support selective disclosure based on a common high-level cryptographic principle: the credential contains cryptographic commitments, typically in the form of salted hashes, that allow individual attributes to be disclosed and subsequently verified without disclosing the other attributes. The disclosed attribute and the corresponding cryptographic information allow the Relying Party to verify that the attribute was included in the credential and protected by the Provider's signature.

The detailed cryptographic structures and verification mechanisms are format-specific and are defined by the applicable SD-JWT VC and ISO/IEC 18013-5 specifications.

For device-bound credentials, the public key associated with the protected private key is included or referenced in the credential in accordance with the applicable credential format. During presentation, the Wallet Unit uses the corresponding private key to provide the required proof of possession or device binding.

1.3.4 Credential (PIDs/Attestation) Disclosure

At this point in the lifecycle, the TOE holds PIDs or Attestations and is ready to disclose selected attributes/claims to the Relying Party.

PIDs/Attestation selective disclosure includes the following steps:

- The TOE authenticates the Relying Party.
- The TOE requests User approval of the attributes/claims to be disclosed and performs selective disclosure.

1.3.4.1 TOE authenticates Relying Party

The TOE authenticates the Relying Party using an Access Certificate (RP_ACC_CERT) and obtains or verifies the applicable Relying Party registration information using a Registration Certificate (RP_REG_CERT) or the applicable Registrar's online service.

Trust in the Access Certificate and Registration Certificate is established using the applicable trusted anchors.

The Access Certificate is used to authenticate the Relying Party or intermediary participating in the presentation transaction. The Registration Certificate or Registrar service provides information about the Relying Party's registered role, Service, intended use and applicable information relevant to the presentation request.

The TOE must inform the User of the outcome of the Relying Party authentication and certificate validation.

1.3.4.2 User approval and selective disclosure

Prior to User approval, the TOE shall authenticate the User using OS-level or WSCD/WSCA-based authentication.

The TOE shall present the following information to the User and obtain explicit approval before disclosure:

- The identity of the Relying Party, and intermediary parties if present.
- Attributes/claims the Relying Party requested
- Relying Party's intended use (from Registration cert or service), and privacy policy for these attributes,
- Outcome of the evaluation of the requested attributes

The TOE shall support selective disclosure using the mechanism defined by the applicable PID/Attestation format. At a high level, the individual attributes are represented by salted cryptographic hashes that are protected by the Provider's signature. When an attribute is disclosed, the Relying Party can use the disclosed attribute and the corresponding hash information to verify that it was included in the credential, without requiring disclosure of the other attributes.

The TOE shall generate a presentation bound to the transaction, including the freshness value provided by the Relying Party, and use the WSCD/WSCA to sign the presentation with the corresponding PID/Attestation private key (PID/ATT_PRIV_KEY). This provides proof-of-possession of the private key.

The Relying Party shall verify the presentation, including its authenticity, integrity, freshness and the authenticity of the Provider, using the applicable credential and presentation specifications. Trust in the Provider's public key shall be established through the applicable Trust List or LoTE.

After the transaction, the TOE shall enable the User to report unlawful or suspicious requests for personal data and to request deletion of personal data from the Relying Party.

1.3.4.3 Pseudonyms

The TOE shall support the generation and use of pseudonyms for interactions with Relying Parties where required or selected by the User.

Pseudonyms shall be unique and unlinkable across different Relying Parties unless the User explicitly chooses otherwise. The TOE shall securely store and manage the information required to use the pseudonyms and shall prevent disclosure of the User's real-world identity through the pseudonym.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

The User shall be able to manage and delete pseudonyms through the Wallet Unit.

1.3.5 Revocation/Deletion

The Wallet Provider must revoke the Wallet Unit upon explicit User request, upon an explicit request from a PID Provider in accordance with the applicable revocation procedure, and when a security breach or compromise affects the trustworthiness or reliability of the Wallet Unit.

The Wallet Provider must regularly verify that the security of the Wallet Unit is not breached or compromised. This includes the security of the Wallet Instance, the mobile device and OS, and the WSCA/WSCD used to manage critical cryptographic assets. When a security breach or compromise is detected, the Wallet Provider must analyse its cause and impact and, where the trustworthiness or reliability of the Wallet Unit is affected, revoke the Wallet Instance. Where the compromise affects a WSCA/WSCD, the Wallet Provider must also revoke the corresponding WSCA/WSCD.

A Wallet Instance is revoked by setting the `revoked` status at all index positions specified in the WIAs issued to that Wallet Instance. A WSCD is revoked by setting the `revoked` status at all index positions specified in the corresponding KAs.

For a revocable PID, the PID Provider must regularly verify the revocation status of the associated Wallet Instance and WSCD using the WIA and KA received during issuance. If either has been revoked, the PID Provider must revoke the corresponding PID. An Attestation Provider may apply a similar procedure to revocable attestations in accordance with its published revocation policy.

The Wallet Unit shall support secure deletion of its contents through a factory reset. This shall delete all attestations, logs, personal data, settings and configurations and, where connected, request the associated WSCA/WSCD to delete the cryptographic assets related to the Wallet Unit.

1.3.6 Additional Wallet Instance Security Functionality

In addition to the core lifecycle and trust relationships, the Wallet Unit provides the following security-relevant functionality for specific use cases and operational scenarios.

1.3.6.1 Qualified Electronic Signature and Seal

The Wallet shall support the creation of qualified electronic signatures and seals using a local or remote QSCD. The Wallet authenticates the User and obtains explicit approval for the signing operation before securely requesting the signature. For a remote QSCD, the Wallet communicates with the authorized QTSP through a protected channel. A PID or attestation may be used to establish or verify the User's identity where required but is not inherently required for authorization of the signing operation. The resulting signature is returned to the Wallet and the transaction is recorded.

1.3.6.2 Wallet-to-Wallet (W2W)

The Wallet Unit supports secure Wallet-to-Wallet interactions between Wallet Units. The W2W interaction establishes an authenticated and protected communication channel and requires User awareness and approval before attributes or electronic attestations are disclosed. The interaction protects the exchanged information and transaction against eavesdropping, modification, replay and transaction manipulation.

1.3.6.3 Migration

The Wallet Unit supports secure migration of Wallet Unit contents to a new Wallet Instance. The migration process uses a protected Migration Object containing the information required to restore or re-issue applicable credentials while excluding sensitive information such as WIA, KA and private keys. The Migration Object is protected during creation, storage, export,

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

import and processing against unauthorized disclosure, modification, replacement and replay.

1.3.6.4 DPA Reporting

The Wallet Unit provides the User with a function to report suspicious or inappropriate Relying Party presentation requests to the competent Data Protection Authority (DPA). The function allows the User to identify the relevant Relying Party and transaction and to provide information supporting the complaint.

1.3.6.5 Trust Mark

The Wallet Instance displays the official EUDI Wallet Trust Mark and associated certification information to the User. The Trust Mark information is obtained from the Wallet Provider and protected against unauthorized modification.

1.3.6.6 Logging as an Explicit Lifecycle/Security Function

The Wallet Unit maintains transaction logs covering security-relevant Wallet activities, including issuance, presentation, W2W interactions and electronic signing or sealing transactions. The logs are protected against unauthorized access, modification and deletion and shall not unnecessarily contain the actual values of presented attributes. The wallet instance also supports User access to the logs and recording of relevant DPA reports. This provides traceability of security-relevant Wallet Unit operations throughout the lifecycle.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

2 Security Problem Definition

2.1 Assets

The assets defined in this in ***Table*** 1 represent the information, cryptographic material, security evidence, and trust relationships that require protection by the WI-(TOE). The WI-(TOE) shall protect these assets against unauthorized disclosure, modification, destruction, misuse, replay, substitution, and other security violations that may be attempted by the threat actors.

The required protection of the assets will be ensured by both measures implemented by the WI-(TOE) and measures implemented by the entities in the ecosystem of the WI-(TOE).

Table 1 ***— Assets***

ID	Asset	Description	Protection
A.MUT_KEY	Mutual Authentication Channel	Cryptographic keys and associated authentication data used to establish the mutually authenticated and protected communication channel between the Wallet Instance and the WSCA/WSCD.	Confidentiality, Integrity, Authenticity
A.PIN	User PIN / Knowledge Factor	User-provided authentication secret used as the knowledge factor for WSCA/WSCD-level User Authentication.	Confidentiality
A.PUK	Personal Unblocking Key	Personal knowledge-based factor to reset the knowledge-based factor for authentication (A.PIN) in case the PIN was forgotten.	Confidentiality
A.POSS	User Possession Factor	Device- or Wallet-bound secret, cryptographic key, or other protected possession factor used together with the knowledge factor (A.PIN or A.PUK) to derive or generate the User Authentication Evidence (UAE).	Confidentiality, Integrity, Authenticity.
A.UAE	User Authentication Evidence	Authentication evidence derived or generated using the User Possession Factor and the applicable User authentication factor(s), demonstrating successful User authentication to the WSCA/WSCD.	Integrity, Authenticity, Freshness.
A.ARD	Authentication Reference Data	Authentication reference material established during User Authentication enrolment and used by the WSCA/WSCD to validate UAE. Disclosure shall not by itself enable User impersonation.	Integrity, Authenticity
A.KEYAUTH	Per-key authorisation data	Per-key authorisation data is generated during key generation, is held by the wallet instance and provided with each operation.	Confidentiality, Integrity, Authenticity
A.CMD	WSCA/WSCD Commands	Security-relevant commands and requests exchanged with the WSCA/WSCD, including key generation, signing and proof-of-possession requests.	Integrity, Authenticity, Freshness
A.WIA	Wallet Instance Attestation	Attestation issued by the Wallet Provider providing evidence concerning the Wallet Instance and its security status.	Integrity, Authenticity
A.KA	Key Attestation	Attestation providing evidence concerning a cryptographic key and its association with the WSCA/WSCD.	Integrity, Authenticity

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

ID	Asset	Description	Protection
A.PIDATT	PIDs and Attestations	PIDs and Attestations stored by the Wallet Unit and used for presentation to Relying Parties or other authorised entities.	Confidentiality Integrity, Authenticity
A.PRES	Presentation Data	Presentation data generated for credential presentation transactions, including OpenID4VP and mdoc presentation objects.	Confidentiality Integrity, Authenticity,
A.NONCE	Nonces / Freshness Data	Challenge and freshness values used to prevent replay and bind protocol messages to a transaction or session.	Integrity, Freshness
A.LOG	Transaction Logs	Records of Wallet transactions, including issuance, presentation, signing/sealing, W2W and other security-relevant events.	Confidentiality, Integrity, Authenticity
A.MON	Monitoring and Security Data	Security-relevant information used for Wallet Provider monitoring, security assessment and security-status decisions.	Confidentiality, Integrity, Authenticity
A.INIT	Initial Wallet Data	Technical and security-related information exchanged during Wallet Instance installation and activation.	Confidentiality, Integrity, Authenticity
A.ANCHOR	Trust Anchors and Trust Lists	Trusted lists, trust anchors and associated information used to validate certificates and signed objects.	Integrity, Authenticity
A.CERT	Access and Registration Certificates	Certificates used to authenticate and authorize Wallet Providers, PID/Attestation Providers, Relying Parties and other authorised entities.	Integrity, Authenticity
A.WIID	Wallet Instance Identifier	Identifier uniquely associated with the Wallet Instance and its Wallet Provider relationship.	Integrity, Authenticity
A.TRUSTM	Trust Mark Information	Information used to display the official EUDI Wallet Trust Mark and associated certification information.	Integrity, Authenticity
A.PSEUD	RP-specific Pseudonyms	Cryptographic information and resulting pseudonyms used to provide Relying Party-specific identification and prevent unnecessary cross-RP linkability.	Confidentiality, Integrity, Privacy
A.MIGR	Migration Object	Information used to migrate Wallet contents and initiate restoration or re-issuance of credentials.	Confidentiality, Integrity, Authenticity
A.QESDATA	QES Signing Data	User-approved data or document and associated transaction information submitted for qualified signature or seal creation.	Confidentiality, Integrity, Authenticity

2.2 Threat model

The Wallet solution shall counter the following threats, described in ***Table*** 2. These threats describe the primary security risks associated with user impersonation and unauthorized disclosure of Personal Data during presentation.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Table** 2 — **Primary Threats

Threat ID	Description
T.USER_IMPERSONATION	An attacker presents a stolen or otherwise obtained PID or Attestation to a Relying Party, including a freshly generated and validly signed nonce, in order to impersonate the legitimate Wallet User without the involvement or authorization of the legitimate User.
T.LEAKED_PRESENTATION_DATA	An attacker causes Personal Data contained in a Wallet Unit to be disclosed to a Relying Party or other unauthorized recipient without the knowledge and authorization of the legitimate Wallet User, including by manipulating or compromising the presentation process.

These two threats **do not have the same security impact**:

- **T.USER_IMPERSONATION is considered the more severe threat**, since a successful attack allows an attacker to act as the legitimate Wallet User and to generate a valid Proof-of-Possession for a stolen or otherwise obtained PID or Attestation.
- **T.LEAKED_PRESENTATION_DATA** addresses unauthorized disclosure of Personal Data during the presentation process. Although this threat can have significant privacy consequences, it does not necessarily provide the attacker with the ability to impersonate the User.

The evaluation shall place **greater focus on the assets and requirements covering T.USER_IMPERSONATION,** as this threat has a higher security impact than T.LEAKED_PRESENTATION_DATA. The requirements mitigating T.LEAKED_PRESENTATION_DATA shall also be evaluated, but with a more limited focus.

Section 2.6 identifies which requirements and assets contribute to the mitigation of each threat and its associated attack scenarios identified in Section 2.3.

2.3 Attack scenarios

***Table** 3 describes the attack scenarios considered to achieve the T.USER_IMPERSONATION.*

Table** 3 — **Attack scenario for T.USER_IMPERSONATION.

Attack scenarios	Description
AS.UI.01 — Fake WSCA/WSCD during channel establishment	Manipulate the enrolment or establishment of the secure channel between the Wallet Instance and the WSCA/WSCD so that the Wallet Instance connects to an attacker-controlled WSCA/WSCD. The attacker uses stolen or compromised Wallet Provider signing keys to generate or sign fraudulent WIA/KA information, causing the resulting attestation information to appear legitimate.
AS.UI.02 — Cloning of Wallet Instance authentication factors	Obtain or compromise both the possession factor and knowledge factor (PIN) used for WSCA/WSCD authentication. The attacker uses the stolen factors to gain control of the Wallet Instance and establish the required authenticated channel to the WSCA/WSCD, allowing the attacker to perform security-sensitive Wallet Unit operations.
AS.UI.03 — Theft of device-bound private key	Obtain the private key corresponding to a public key bound to a PID or Attestation. The attacker uses the private key to generate a fresh and valid proof of possession for a Relying Party, allowing a stolen or copied PID/Attestation to be presented without the involvement of the legitimate Wallet User.
AS.UI.04 — Bypass of User authentication	Exploit or manipulate the Wallet Instance to bypass OS-level User authentication or the WSCA/WSCD-level Remote User Authentication Protocol (RUAP). The attacker then causes protected cryptographic operations to be performed without the involvement or authorization of the legitimate Wallet User.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Attack scenarios	Description
AS.UI.05 — Manipulation of WIA/KA binding	Manipulate the WIA or KA, so that attacker-controlled keys are presented as belonging to the legitimate Wallet Unit. The attacker subsequently uses the corresponding private key to generate a valid proof of possession during credential presentation.
AS.UI.06 — Replay or manipulation of authentication data	Capture, replay, or manipulate authentication information exchanged between the Wallet Instance and WSCA/WSCD in an attempt to establish an authenticated session or obtain authorization for a security-sensitive cryptographic operation without the legitimate User's involvement.
AS.UI.07 — Bypass via Migration	Exploit weaknesses in the Wallet migration or restore process to transfer Wallet data or credentials to an attacker-controlled or insufficiently authenticated Wallet Instance. The attacker bypasses the required User authentication, source-instance authorization, or destination-instance verification and obtains access to a migrated PID/Attestation or the associated device-bound private key. The attacker can then use the credential and key to generate a valid Proof-of-Possession and impersonate the legitimate Wallet User.

Table 4 describes the attack scenarios considered to achieve the T.LEAKED_PRESENTATION_DATA.

Table 4 ***— Attack scenario for T.LEAKED_PRESENTATION_DATA***

Attack scenarios	Description
AS.LPD.01 — Fake Relying Party	Manipulate or replace the Relying Party authentication information so that the Wallet Instance accepts an attacker-controlled or unauthorized entity as the legitimate Relying Party. The attacker then requests Personal Data from the Wallet Instance and causes the User to disclose data to the attacker.
AS.LPD.02 — Manipulation of presentation request	Manipulate the presentation request between the Relying Party and Wallet Instance to alter the requested attributes or claims. The attacker causes the Wallet Instance to request or disclose additional Personal Data that was not originally requested or authorized by the User.
AS.LPD.03 — Bypass of User approval	Compromise or manipulate the Wallet Instance to bypass the required User authentication and explicit approval before presentation. The attacker causes the Wallet Instance to perform selective disclosure and generate a presentation without the User knowingly authorizing the disclosure.
AS.LPD.04 — Manipulation of User approval information	Manipulate the information presented to the User during the approval process, such that the User is shown an incomplete or incorrect representation of the Relying Party, requested attributes, intended use, or privacy information. The User consequently authorizes a presentation containing Personal Data that was not knowingly approved.
AS.LPD.06 — Manipulation of selective disclosure	Manipulate the Wallet Instance or presentation processing so that attributes other than those explicitly selected and approved by the User are included in the presentation. The resulting presentation remains cryptographically valid but contains Personal Data beyond the User's intended disclosure.
AS.LPD.07 — Presentation data interception	Intercept Personal Data during the presentation process between the Wallet Instance and the Relying Party or an intermediary. The attacker obtains the disclosed attributes by compromising or manipulating the communication channel or presentation interfaces.
AS.LPD.08 — Replay or transaction manipulation	Capture or manipulate information from a presentation transaction and attempt to reuse it in another transaction or with another Relying Party. The attacker attempts to cause Personal Data to be disclosed outside the transaction that was explicitly approved by the User.

2.4 User and External Entities

Table 5 identifies the User and external entities that directly interact with the WI-(TOE) in security-relevant Wallet operations. Section 2.5 outlines the required security verification that must be enforces by the subjects.

Table 5 — *User and External Entities*

ID	Subject	Role
S.USER	User	Operates the Wallet Instance and authorizes security-sensitive operations such as authentication, credential disclosure and QES.
S.WP	Wallet Provider	Provides Wallet Instance services, including Wallet Instance activation, WIA issuance and Wallet lifecycle/status management.
S.WSCA/WSCD	WSCA/WSCD	Performs or controls security-sensitive cryptographic operations and protects cryptographic assets used by the Wallet Instance.
S.PROVIDER	PID / Attestation Provider	Issues PIDs and attestations to the Wallet Instance and verifies the Wallet's WIA/KA and associated cryptographic proof.
S.RP	Relying Party	Requests and receives credential presentations from the Wallet Instance.
S.QTSP	QTSP / Remote QSCD	Provides remote qualified signature/seal creation services when the QES functionality is externally managed.

2.5 Environmental Security Objective

Table 6 identifies the security objectives applicable to the subjects and external entities in the operational environment of the Wallet Instance. These objectives define the security activities and responsibilities that support the secure operation of the Wallet Unit, the protection of cryptographic assets, and the establishment and maintenance of the required trust relationships.

Table 6 — *Environmental Security Objective*

ID	Environmental Security Objective
OE.USER	**User Control and Protection of Authentication Factors** The User shall maintain control over the Wallet Instance and protect the credentials and authentication factors required to operate it. ▪ Keep Wallet Instance credentials and authentication factors confidential and not share them with others. ▪ Protect the PIN against disclosure, including during PIN entry. ▪ Use the Wallet Instance only under their own control.
OE.WP	**Wallet Provider Trust and Lifecycle Management** The Wallet Provider shall establish, maintain and protect the trust relationship with the Wallet Instance throughout its lifecycle. ▪ Securely activate the Wallet Instance and associate it with the appropriate Wallet Provider account. ▪ Establish the required communication and cryptographic trust relationships. ▪ Issue the WIA and KA required for Wallet Unit operations. ▪ Sign the WIA and KA using the Wallet Provider signing key. ▪ Ensure the Wallet Provider signing key is verifiable through the applicable trusted-anchor certificate chain. ▪ Maintain Wallet Instance validity and revocation status and revoke a compromised Wallet Instance. ▪ Protect the integrity and authenticity of information used by PID and Attestation Providers to establish trust in the Wallet Instance
OE.WSCA/WSCD	**Protection of Cryptographic Assets and Secure Cryptographic Services** The WSCA/WSCD shall protect the Wallet Unit's security-critical cryptographic assets and provide a trusted and secure cryptographic service to the Wallet Instance.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

ID	Environmental Security Objective
	■ Protect security-critical Wallet Unit cryptographic assets, including PID/Attestation private keys and, where applicable, the WIA device-binding private key. ■ Establish an authenticated and protected communication channel with the Wallet Instance, providing confidentiality and integrity protection against eavesdropping, modification and manipulation. ■ Perform security-sensitive cryptographic operations only through the established secure channel and following the required User authentication and authorization. ■ Protect authentication and cryptographic operations against unauthorized use, disclosure, modification, replay and manipulation. ■ Provide proof-of-possession operations without exposing the protected private keys. ■ Limit the number of User authentication attempts: keep a failure counter for each User that the Wallet Instance cannot reset. Block the use of protected cryptographic assets after a defined number of consecutive failed verifications and apply increasing delays between attempts before that. ■ Unblock only after a successful reset procedure (e.g. recovery using PUK). Apply this limit to every verification of User Authentication Evidence, so that the knowledge factor cannot be guessed online through the WSCA/WSCD. ■ Per-key authorisation data is generated by the WSCD during key generation, is sent to the wallet instance and afterwards provided with each operation. The per-key authorization data will be securely deleted by the WSCA when its no longer needed by the WSCA.
OE.PROVIDER	**PID/Attestation Issuance and Wallet Unit Trust** The PID/Attestation Provider shall establish trust in the Wallet Unit and securely bind issued credentials to the verified Wallet Unit cryptographic keys. ■ Authenticate to the Wallet Instance using the applicable access certificate and registration certificate. ■ Verify the WIA and KA for authenticity, validity and revocation status using the applicable trust framework. ■ For device-bound credentials, verify proof of possession of the private key corresponding to the public key contained in the KA. ■ Bind the issued PID or Attestation to the verified public key. ■ Sign issued PIDs and Attestations using the applicable Provider signing key. ■ Ensure the Provider signing key and issued credentials can be validated through the applicable trusted-anchor certificate chain.
OE.RP	**Presentation Verification and Transaction Trust** The Relying Party shall establish trust in the presentation and verify that the presented Personal Data and associated credentials are valid for the transaction. ■ Authenticate to the Wallet Instance using the applicable access certificate and registration certificate. ■ Verify the authenticity, integrity and freshness of each presentation. ■ Verify the applicable PID/Attestation Provider signature using the applicable trusted-anchor certificate chain. ■ For device-bound credentials, verify proof of possession of the private key associated with the presented credential. ■ Accept and process only Personal Data validly presented and authorized for the applicable transaction.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

ID	Environmental Security Objective
OE.QTSP	**Protected Qualified Signature and Seal Services** The QTSP/Remote QSCD shall provide protected qualified signature or seal services and ensure that signing operations are performed only as authorized by the User. ■ Protect qualified signing keys against unauthorized access and use. ■ Perform qualified signature or seal operations only through the authorized signing service and following the required User authentication and explicit authorization. ■ Protect communication with the Wallet Instance against unauthorized disclosure and manipulation. ■ Securely return the resulting qualified signature or seal to the Wallet Instance.
OE.USER_DEVICE	**Secure User Device Platform** The User Device on which the Wallet Instance is installed shall provide a hardware-supported security platform with an operating system that still receives security updates. The Wallet Provider shall support the Wallet Instance only on User Devices that meet these conditions: ■ Provide a Trusted Execution Environment (TEE) or Secure Element (SE) offering an isolated execution and storage environment for security-relevant Wallet Instance assets. ■ Provide a hardware-backed keystore, implemented in the TEE or SE, in which keys are generated, stored and used without the private key material being exported, cloned or exposed to the operating system or other applications. ■ Enforce key-usage restrictions, including binding key use to User authentication where requested by the Wallet Instance. ■ Provide key and device attestation giving authentic evidence of the properties of protected keys and the security state of the device. ■ Provide a trusted boot mechanism so that only authenticated and authorised firmware and operating-system components run with privileged access. ■ Enforce application sandboxing, process isolation and permission controls between the Wallet Instance and other applications. ■ Provide OS-level User authentication based on possession of the device combined with a knowledge-based or inherence-based factor. ■ Receive security updates from the device or OS manufacturer for the period during which the Wallet Provider supports the device
OE.APP_DISTRIBUTION	**Authentic Distribution and Update of the Wallet Instance** The Wallet Provider shall distribute the Wallet Instance and all its updates only through channels that preserve their authenticity and integrity: ■ Distribute the Wallet Instance and its updates through the official application distribution channel of the User Device operating system, or through another mechanism with equivalent protection that lets the User verify the authenticity and integrity of the package before installation. ■ Sign every release with a Wallet Provider signing key protected against unauthorised use and disclosure. ■ Ensure that the package signature is verified before the Wallet Instance is installed or updated, and that updates are accepted only when signed by the same signing identity as the installed version. ■ Release only versions covered by the applicable certification or certification maintenance process. ■ Provide security updates in a timely manner when vulnerabilities affecting the Wallet Instance are identified. ■ Define a minimum supported version and prevent older versions from being used for security-sensitive operations, for example by refusing to issue a WIA to them.

2.6 Coverage of the Security Requirements

Chapter 3 provides the **complete set of detailed security requirements**, including the Functional Security Requirements (FSRs) and Resilience Security Requirements (RSRs). The FSRs are derived from the ARF and applicable regulations, while the RSRs provide additional resilience and robustness measures for the Wallet Instance.

2.6.1 Coverage of the Life Cycle

Appendix A.1 provides the coverage mapping between the security requirements and the relevant life-cycle sections defined in section 1.3. The mapping demonstrates that the security-relevant activities throughout the Wallet Instance life cycle are addressed by the identified FSRs.

2.6.2 Coverage of the Threats

This section identifies the assets, Functional Security Requirements (FSRs), Resilience Security Requirements (RSRs), and Environmental Security Objectives that contribute to the mitigation of the two threats defined for the Wallet Instance. The coverage is based on the attack scenarios identified in Appendix A.2.

2.6.2.1 Requirements and Assets covering T.USER_IMPERSONATION

The primary objective for T.USER_IMPERSONATION is to ensure that an attacker cannot successfully act as the legitimate Wallet User by presenting a PID or Attestation together with a valid Proof-of-Possession. ***Table*** 7 gives a coverage assessment for T.USER_IMPERSONATION.

Table *7 — Coverage for T.USER_IMPERSONATION*

Assets and Requirements	Coverage
Assets	The assets that require particular protection for this threat are **A.WIID, A.PIN, A.PUK, A.KEYAUTH**, **A.POSS**, **A.UAE** and **A.ARD**, which together support the User authentication mechanism; **A.MUT_KEY** and **A.CMD**, which protect the authenticated communication and security-sensitive WSCA/WSCD operations; **A.WIA** and **A.KA**, which establish trust in the Wallet Instance and its cryptographic keys; **A.PIDATT**, which contains the credentials used for identification; and **A.NONCE**, which provides freshness for proof and presentation transactions. **A.MIGR** is used to perform secure migration.
Functional Security Requirements	The threat is mitigated through the combined protection of **FSR_WI_INSTALL, FSR_SEC_CHANNEL, FSR_LOCAL_AUTH, FSR_REMOTE_AUTH and FSR_WIA, FSR_MIGRATION**. **FSR_WI_INSTALL** and **FSR_WIA** and **FSR_KEY_ATTEST** ensure that the Wallet Instance and its cryptographic bindings are established and verified every 24h. **FSR_SEC_CHANNEL** protects channel establishment and authentication exchanges against interception, replay and manipulation, while **FSR_LOCAL_AUTH** and **FSR_REMOTE_AUTH** prevent authentication bypass and require the legitimate User to authorize security-sensitive operations. Together, these measures prevent an attacker from establishing a connection with a fraudulent WSCA/WSCD, using cloned authentication factors, or using compromised device-bound keys or manipulated authentication/binding data to impersonate the legitimate User. **FSR_MIGRATION** ensures that migration from one wallet instance to an other wallet instance is properly secured.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Assets and Requirements	Coverage
Resilience Security Requirements	The collective RSRs in section 3.2 mitigate this threat by maintaining the security of the Wallet Instance, its authentication factors, cryptographic keys, execution environment and security-sensitive interactions even in the presence of active attacks or a compromised execution environment. They prevent unauthorized extraction or cloning of device-bound keys and authentication data, detect compromised or manipulated device and software states, protect sensitive authentication input and communications, and prevent security-critical operations when the required security state or integrity cannot be established. This collectively limits the ability to substitute or manipulate the Wallet Instance, obtain the factors or keys required for authentication, bypass user interaction, or use compromised components to perform operations as the legitimate User.
Environmental Objectives	The **environmental objectives** are particularly important for this threat because the Wallet Instance does not itself provide all of the required security functionality. **OE.USER** protects the User's authentication factors and requires the User to retain control of them. **OE.WP** establishes and maintains the trust relationship with the Wallet Provider and protects the WIA/KA issuance chain. **OE.WSCA/WSCD** protects the security-critical private keys and provides the cryptographic operations required to generate Proof-of-Possession without exposing those private keys. **OE.PROVIDER** verifies the WIA, KA and Proof-of-Possession and binds issued credentials to the verified public key. Finally, **OE.RP** verifies the authenticity of the credential and the Proof-of-Possession during presentation. These environmental responsibilities are explicitly defined in the Protection Profile. **OE.USER_DEVICE** mandates minimum security related requirements for the underlying hardware. Furthermore, **OE.APP_DISTRIBUTION** mandates a secure application distribution channel.

2.6.2.2 Requirements and Assets covering T.LEAKED_PRESENTATION_DATA

T.LEAKED_PRESENTATION_DATA has a different and narrower security objective. Here, the attacker does not necessarily need to impersonate the User. The threat is that Personal Data contained in the Wallet Unit is disclosed to an RP or another recipient without the User's knowledge and authorization. ***Table*** 8 gives a coverage assessment T.LEAKED_PRESENTATION_DATA.

Table 8 ***— Coverage for T.LEAKED_PRESENTATION_DATA***

Assets and Req	Coverage
Assets	The principal assets for this threat are **A.PIDATT**, because it contains the Personal Data-bearing PIDs and Attestations; **A.PRES**, because it contains the presentation data generated for a transaction; **A.CERT**, because RP access and registration certificates establish the identity and authorization of the RP; **A.ANCHOR**, because trust anchors are required to validate the associated certificates and signed objects; and **A.NONCE**, because freshness data binds the presentation to the intended transaction and prevents replay.
Functional Security Requirements	The core FSRs for this threat are consequently **FSR_TRUST_ANCHOR, FSR_ACC_REG_CERT, FSR_SELECT_DISCLO, FSR_PRESENTATION, FSR_PROTOCOL FSR_DATA_DELETION** and **FSR_MDOC**. **FSR_TRUST_ANCHOR** and **FSR_ACC_REG_CERT** ensures that the RP is authenticated and that its access and registration information is valid before disclosure. It also ensures that the RP or service information presented to the User corresponds to the authenticated entity. **FSR_SELECT_DISCLO** is the central requirement for this threat because it ensures that only attributes explicitly approved by the User are included in the presentation. **FSR_PROTOCOL** ensures that the presentation request, User approval and resulting presentation remain associated with the same transaction and are protected against manipulation and replay. **FSR_PRESENTATION** ensures that only valid credentials are presented, while **FSR_MDOC** provides the corresponding protection for mdoc presentations. **FSR_DATA_DELETION** ensures that presentation data is properly deleted.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Assets and Req	Coverage
Resilience Security Requirements	This threat is mitigated by the collective resilience measures that protect the confidentiality and integrity of Personal Data throughout the presentation process and maintain the Wallet Instance in a trusted security state. They prevent unauthorized applications, compromised platform components, or manipulated interfaces from intercepting or altering presentation information, protect sensitive User interactions and approval information from capture or UI manipulation, and ensure that security-critical processing is stopped when the Wallet Instance or its execution environment is compromised. The protection of cryptographic keys and sensitive data further prevents attackers from extracting or reusing presentation-related credentials. Together, these measures limit the ability to impersonate or manipulate the Relying Party, alter presentation requests or User approvals, bypass User interaction, disclose attributes beyond those selected, intercept Personal Data, or reuse presentation information outside the transaction authorized by the User.
Environmental Objectives	The environmental objectives needed for this threat are principally **OE.USER** and **OE.RP**. **OE.USER** establishes the User's responsibility to operate the Wallet under their own control and authorize security-sensitive actions. **OE.RP** establishes the RP-side responsibility to authenticate, verify the presentation, verify its authenticity, integrity and freshness, and process only Personal Data validly presented and authorized for the transaction.

2.6.3 Coverage of all requirements

Table 9 summarizes the relationship between the two primary threats, the Functional Security Requirements (FSRs), and the assets requiring protection. Requirements and assets may contribute to the mitigation of both threats. The Supporting category identifies the remaining requirements and assets that support other Wallet security functions and lifecycle activities. Applicable Resilience Security Requirements (RSRs), as defined in Section 3.2, also contribute to threat mitigation.

Table 9 ***— Coverage of all requirements***

Classification	Requirement	Assets
T.USER_IMPERSONATION	FSR_WI_INSTALL, FSR_SEC_CHANNEL, FSR_LOCAL_AUTH, FSR_REMOTE_AUTH and FSR_WIA, FSR_MIGRATION. All mandatory resilience requirements.	A.PIN; A.PUK; A.KEYAUTH; A.POSS; A.UAE; A.ARD; A.MUT_KEY; A.CMD; A.WIA; A.KA; A.PIDATT; A.NONCE; A.WIID; A.MIGR
T.LEAKED_PRESENTATION_DATA	FSR_TRUST_ANCHOR, FSR_ACC_REG_CERT, FSR_SELECT_DISCLO, FSR_PRESENTATION, FSR_PROTOCOL, FSR_DATA_DELETION and FSR_MDOC. All mandatory resilience requirements.	A.PIDATT; A.PRES; A.CERT; A.ANCHOR; A.NONCE; A.CERT; A.MUT_KEY
Supporting	FSR_ISSUANCE, FSR_PSEUDONYM; FSR_REVOCATION; FSP_Q_SIGNATURE; FSR_W2W; FSR_DPA_REPORT; FSR_TRST_MRK; FSR_LOG_DATA; All mandatory resilience requirements.	A.LOG; A.MON; A.INIT; A.ANCHOR; ; A.TRUSTM; A.PSEUD; A.QESDATA

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

3 Detailed Security Requirements

3.1 WI-(TOE) Functional Security Requirements

Table 10 defines the **Functional Security Requirements (FSRs)** for the WI-(TOE). The FSRs specify the security functionality that shall be provided by the Wallet Instance and its security-critical components to protect the identified assets and address the threats defined in this Protection Profile.

The requirements cover the security functions necessary for the secure operation of the Wallet, including User authentication and authorization, protection of cryptographic assets, secure communication, credential issuance and presentation, trust and certificate validation, qualified electronic signatures, lifecycle management, logging, migration, and data deletion.

Table 10 — ***Functional Security Requirements***

FSR_ID	Functional Security Requirements
FSR_WI_INSTALL	**Wallet Instance Installation and Activation** The WI-(TOE) shall be installed through an **official application distribution channel** (e.g. Google Play Store or Apple App Store) or through another distribution mechanism providing equivalent protection for the authenticity and integrity of the TOE. The WI-(TOE) shall: ■ During installation and activation, the TOE shall establish the required association with the Wallet Provider and shall securely provide the required initial information to the Wallet Provider. This information shall include, where applicable: ○ a unique identifier for the Wallet Instance; ○ the device model and operating-system version; and ○ the characteristics of the available keystores and WSCDs required to determine their suitability. ■ The TOE shall protect the confidentiality, integrity and authenticity of the initial information exchanged with the Wallet Provider. ■ The TOE shall verify that the installed TOE version and the underlying operating-system environment meet the applicable minimum-security requirements before activating the Wallet Instance. ■ If the TOE or operating-system environment does not meet the applicable security requirements, the TOE shall prevent activation or security-sensitive Wallet Unit operations and shall inform the User. ■ During activation, the TOE shall establish the required User authentication mechanisms in accordance with FSR_LOCAL_AUTH and FSP_REMOTE_AUTH. The WI-(TOE) shall provide a User-initiated factory reset function. Upon factory reset, the TOE shall: ■ delete the PIDs and electronic attestations held by the Wallet Instance; ■ delete transaction logs and Wallet Instance configuration and settings; ■ request secure deletion of associated cryptographic assets from the WSCA/WSCD and applicable secure keystores; ■ prevent the deleted Wallet Unit data and cryptographic assets from being subsequently used by the Wallet Instance; and ■ inform the User of the consequences of performing the factory reset before the reset is executed.

FSR_ID	Functional Security Requirements
FSR_SEC_CHANNEL	**Secure Communication Channels** The WI-(TOE) establishes and maintains mutually authenticated and protected communication channels with the WSCA and the Wallet Provider. The communication channels shall provide **authentication, confidentiality and integrity protection**, and shall protect security-relevant communications against **replay and modification**. All communication between the WI-(TOE) and the WSCA or the Wallet Provider shall use these protected channels.
FSR_LOCAL_AUTH	**Local User Authentication** The WI-(TOE) implements local OS-based authentication. The authentication mechanism shall be **multi-factor authentication**, including **possession of the User's device as one factor** and an **additional knowledge-based or inherence-based factor**. This local OS-based authentication shall be enforced at startup of the WI-(TOE) and before security-sensitive actions which require User authentication, as defined in the applicable Regulation and ARF requirements. The TOE shall ensure that authentication is no longer considered valid after the applicable inactivity timeout and shall require the User to authenticate again before performing protected actions.
FSR_REMOTE_AUTH	**Remote User Authentication** The WI-(TOE) implements the WSCA/WSCD-level Remote User Authentication Protocol (RUAP) between the WI-(TOE) and the WSCA/WSCD. The RUAP shall support User authentication meeting the applicable requirements for Level of Assurance High as specified in Commission Implementing Regulation (EU) 2015/1502. In particular, the authentication mechanism shall use at least two authentication factors from different categories and shall implement security controls such that guessing, eavesdropping, replay and manipulation of the authentication communication cannot, with high attack potential, subvert the authentication mechanism. This specific Remote User Authentication Protocol (RUAP), including the use of User Authentication Evidence (UAE) and Authentication Reference Data (ARD), is a PP-specific mechanism defined to implement the required WSCA/WSCD-level User authentication. The **protocol** shall ensure that: ▪ The Wallet Instance WI-(TOE) must demonstrate the User Authentication Evidence (UAE), which is derived both from a **knowledge-based authentication factor** (derived from the PIN) and a **possession-based authentication factor**, which are jointly validated to achieve two-factor authentication and to prevent manipulation or forged attempts; ▪ Remote User Authentication Protocol (RUAP) ensure PIN Verification and ensure that the User has **sole control** over the critical assets located in the WSCD, such that these assets cannot be used without the User's involvement through the required authentication factors; ▪ The protocol protects the confidentiality and integrity of the authentication data during PIN verification and enrolment, and provides protection against **duplication, tampering, eavesdropping and replay attacks**, including through freshness and challenge binding; ▪ Each **PIN** verification is performed through a live and fresh interaction between the WI-(TOE) and the WSCA/WSCD, preventing reuse of previously exchanged authentication information; ▪ During enrolment, the Wallet Instance WI-(TOE) ensures that both the knowledge and possession factors are securely initialized and cryptographically bound to the **Authentication Reference Data (ARD)**. Authentication Reference Data (ARD) is derived and is shared with the WSCA /WSCD during initial enrolment. This binding preserves confidentiality and integrity of the knowledge-based authentication factor (derived from the PIN) and a possession-based authentication factor and prevents compromise also if the confidentiality of the Authentication Reference Data (ARD) is not preserved. ▪ A **PUK** can be integrated in the protocol to reset the knowledge-based factor in case the PIN was forgotten.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

FSR_ID	Functional Security Requirements
	▪ **Per-key authentication data** shall be sent to the Wallet Instance WI-(TOE) at key generation and be provided at every security-sensitive cryptographic operation of the applicable key, to ensure sole control of the Wallet User before the execution by the WSCD. As part of the enrolment process, the Wallet User initializes the WSCA/WSCD-level User Authentication by instructing the WI-(TOE) to generate the required cryptographic keys within the secure keystore. The User shall provide their PIN during this process, from which the knowledge-based factor is derived and securely incorporated into the authentication mechanism. The WSCA/WSCD-level Remote User Authentication shall be performed before security-sensitive operations involving critical assets located in the WSCD, as defined in the applicable Regulation and ARF requirements.
FSR_WIA	**Wallet Instance Attestation** The WI-(TOE) maintains and protects the Wallet Instance Attestation (WIA) and presents the WIA to PID Providers and Attestation Providers as required, enabling them to verify the security properties and status of the Wallet Unit. The TOE shall: ▪ obtain the WIA from the Wallet Provider through an authenticated and integrity-protected communication channel; ▪ verify the **authenticity and integrity** of the WIA before using or presenting it; ▪ ensure that the WIA applies to the corresponding Wallet Instance; ▪ verify that the WIA is **technically valid and has not expired** before use; ▪ verify the revocation status of the Wallet Instance as indicated by the WIA before use, where required by the applicable ARF and technical specifications; ▪ protect the WIA against unauthorized disclosure, modification, replacement and replay; and ▪ use the WIA only for the applicable issuance processes and in accordance with the applicable OID4VCI requirements. The TOE shall provide the Wallet Instance Attestation together with the required **Proof-of-Possession (PoP)** during issuance, as specified by OID4VCI. The technical validity period of a WIA shall be **less than 24 hours**, measured from the time at which the Wallet Provider verified the integrity of the Wallet Instance to the expiry time indicated in the WIA. The technical validity period shall not be confused with the period for which the Wallet Provider maintains the revocation status of the Wallet Instance. If a WIA is stored in the private storage of the TOE, the WIA shall be protected against unauthorized access, modification and deletion. The TOE shall verify its integrity and validity before use. The TOE must prove its integrity to the Wallet provider before the Wallet provider generates a WIA.
FSR_KEY_ATTEST	**Key Attestation** The WI-(TOE) maintains and protects Key Attestations (KA) and presents KAs to PID Providers and Attestation Providers, where required, to provide evidence of the security properties of the keys used by the Wallet Unit. The TOE shall: ▪ verify the **authenticity and integrity** of a KA before using or presenting it; ▪ ensure that a KA is associated with the **WSCD** in which the corresponding private key is stored; ▪ ensure that the public key(s) attested by a KA correspond to private key(s) stored in the referenced WSCD; ▪ use a KA only for the **applicable issuance or re-issuance process** and shall not reuse a KA beyond the permitted issuance process; ▪ protect KAs against unauthorized disclosure, modification, replacement and replay; and

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

FSR_ID	Functional Security Requirements
	▪ verify the validity and revocation status of the relevant KA, WSCD before using the KA where required by the applicable ARF and technical specifications. The TOE shall present a KA only to authorised PID Providers or Attestation Providers during applicable issuance processes and shall not present a KA to a Relying Party. If a KA is stored in the private storage of the TOE, the KA shall be protected against unauthorized access, modification and deletion. The TOE shall verify its integrity before use. The KA shall be used in accordance with the applicable OID4VCI proof mechanism and the cryptographic mechanisms specified by the applicable EU technical specifications.
FSR_PROTOCOL	**Secure Wallet Protocol Processing** The WI-(TOE) implements or supports the applicable OpenID4VCI, OpenID4VP, ISO/IEC 18013-5, ISO/IEC 18013-7 and Digital Credentials API flows. The TOE shall protect the Wallet User and the Wallet Unit during these flows against **phishing, relay, session-hijacking, replay and transaction-manipulation attacks**. The TOE shall: ▪ verify the authenticity and integrity of security-relevant protocol messages before processing them; ▪ ensure that security-sensitive messages are bound to the applicable transaction and communication session; ▪ prevent replay or reuse of security-sensitive protocol messages; ▪ ensure that the transaction presented to the User corresponds to the transaction actually processed and transmitted; and ▪ terminate or reject the affected transaction if the required security checks fail. The TOE shall implement the applicable protocol flows in accordance with the relevant ARF and technical specifications.
FSR_TRUST_ANCHOR	**Trusted Anchor Management** The WI-(TOE) maintains and protects the applicable **trusted anchors (CA)** required for verification of certificates and signed objects used by the Wallet Unit. The TOE shall: ▪ use only trusted anchors obtained from **authorised and applicable trust sources**; ▪ protect trusted anchors against unauthorized modification, deletion, replacement.; ▪ verify the integrity and authenticity of trusted-anchor information before using it; ▪ use the applicable trusted anchor to verify the authenticity and integrity of certificates and signed objects; ▪ ensure that the trusted anchor used for verification is applicable to the type of certificate or signed object being verified; and ▪ use trusted-anchor and certificate status information in accordance with the applicable ARF, Trusted List and technical specifications. If trusted-anchor information cannot be successfully authenticated or its integrity cannot be verified, the TOE shall not use that information for security-sensitive verification.
FSR_ACC_REG_CERT	**Relying Party Access and Registration Certificate Validation** The WI-(TOE) verifies the Wallet-Relying Party Access Certificate (RP_ACC_CERT) and Wallet-Relying Party Registration Certificate (RP_REG_CERT) before processing a request for disclosure of a PID or electronic attestation from a Relying Party. The TOE shall: ▪ authenticate and validate the RP_ACC_CERT, including its certificate chain, validity and applicable trust anchor; ▪ authenticate and validate the RP_REG_CERT; ▪ verify that the request is associated with the authenticated Relying Party; ▪ verify that the requested attributes are consistent with the information contained in the RP_REG_CERT, where applicable; and ▪ inform the User of the relevant information contained in the RP_ACC_CERT and RP_REG_CERT before disclosure, where required.

FSR_ID	Functional Security Requirements
	▪ inform the User when authentication or validation of the Relying Party fails; ▪ ensure that the authenticated Relying Party and the applicable Service are identified before the User approves a presentation; and ▪ ensure that the Relying Party and Service information presented to the User corresponds to the authenticated RP Access Certificate and applicable RP Registration Certificate. If validation or authentication of the RP_ACC_CERT or RP_REG_CERT fails, the TOE shall not disclose the requested PID or electronic attestation.
FSR_ISSUANCE	**Credential Issuance and Re-Issuance** The WI-(TOE) shall securely process the issuance and re-issuance of PIDs and electronic attestations in accordance with the applicable ARF and technical specifications. The TOE shall: ▪ request issuance only from an authenticated and authorised PID Provider or Attestation Provider; ▪ provide the applicable WIA and KA information required for the issuance process; ▪ verify that the received PID or electronic attestation corresponds to the credential requested; ▪ verify the authenticity and integrity of the received PID or electronic attestation using the applicable trusted anchor; ▪ verify the validity of the received PID or electronic attestation before storing it; ▪ require User approval before storing the received PID or electronic attestation; ▪ delete a received PID or electronic attestation if a required issuance verification fails and inform the User of the failure; ▪ securely handle failed or refused issuance and re-issuance operations; and ▪ ensure that obsolete credentials are no longer used for presentation where required by the applicable ARF and technical specifications.
FSR_PRESENTATION	**Credential Presentation** The WI-(TOE) maintains and protects PIDs and electronic attestations of attributes and presents them to authorised Relying Parties and other authorised entities in accordance with the applicable ARF requirements. The TOE shall: ▪ verify the **authenticity and integrity** of a PID or electronic attestation before storing or using it; ▪ ensure that the PID or electronic attestation is associated with the corresponding Wallet Unit and, where applicable, the WSCA/WSCD; ▪ protect stored PIDs and electronic attestations against unauthorized disclosure, modification, replacement and deletion; ▪ verify the **validity and revocation status** of a PID or electronic attestation before presentation, where required by the applicable ARF and technical specifications; ▪ ensure that a PID or electronic attestation is presented only to an authorised entity and only after the applicable authentication and authorization requirements have been satisfied; ▪ allow the User to delete held PIDs and electronic attestations and inform the User of the deletion; and ▪ allow the User to initiate deletion requests towards Relying Parties for PIDs and electronic attestations held by those Relying Parties, where supported by the applicable protocol. The TOE shall not present a PID or electronic attestation if its authenticity, integrity or validity cannot be successfully verified, or if the applicable Relying Party or other recipient has not been successfully authenticated. The TOE shall preserve the cryptographic binding of device-bound PIDs and electronic attestations to the corresponding Wallet Unit and shall ensure that such credentials cannot be used with an unauthorized or different Wallet Unit.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

FSR_ID	Functional Security Requirements
FSR_SELECT_DISCLO	**Selective Disclosure and User Consent** The WI-(TOE) enforces selective disclosure by ensuring that **only User-approved attributes and electronic attestations** are released to authorised Relying Parties, consistent with the principles of user consent, data minimisation and selective disclosure. The TOE shall: ▪ display to the User the attributes and electronic attestations requested by the Relying Party before presentation; ▪ process and enforce applicable **embedded disclosure policies** associated with electronic attestations; ▪ ensure that the User can approve the requested presentation **in full or in part**; ▪ ensure that no requested attributes or electronic attestations are presented before the applicable disclosure policies have been processed and the User has approved the presentation; ▪ ensure that only the attributes and electronic attestations approved by the User are included in the presentation; and ▪ prevent disclosure if the applicable disclosure policy is not satisfied or the required User approval has not been obtained. ▪ The TOE shall evaluate applicable embedded disclosure policies against the authenticated Relying Party and Service information and shall prevent disclosure if the policy requirements are not satisfied. ▪ The TOE shall ensure that the User approval applies to the actual presentation transaction and to the attributes and electronic attestations that are transmitted to the Relying Party. ▪ The TOE shall retain control over security-sensitive User approval and disclosure decisions and shall not delegate such decisions to a browser, operating-system component or other intermediary application. The TOE shall ensure that selective disclosure is performed in accordance with the applicable ARF and technical specifications and shall support privacy-preserving mechanisms, including unlinkability where required.
FSR_MDOC	**mDoc Presentation** The WI-(TOE) supports ISO/IEC 18013-5 mdoc-based presentation flows in accordance with the applicable ARF and technical specifications. The TOE shall: ▪ support the applicable ISO/IEC 18013-5 requirements for mdoc presentation; ▪ verify the authenticity and integrity of received mdoc presentation requests before processing them; ▪ ensure that the mdoc response contains only the attributes and data elements approved by the User and permitted by the applicable disclosure policies; ▪ protect the mdoc presentation against replay, modification and transaction manipulation; ▪ ensure that security-relevant mdoc messages are bound to the applicable presentation session and transaction; and ▪ verify the authenticity and integrity of received mdoc data before using or presenting it. Where an API-mediated presentation mechanism based on ISO/IEC 18013-7 is used, the TOE shall comply with the applicable ISO/IEC 18013-7 requirements and the corresponding EUDI Wallet technical specifications.
FSR_PSEUDONYM	**Relying Party-Specific Pseudonyms** The WI-(TOE) supports the generation, management and presentation of **Wallet-Relying Party-specific pseudonyms** in accordance with the applicable ARF and technical specifications. The TOE shall: ▪ generate a pseudonym that is specific and unique to the requesting Relying Party;

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

FSR_ID	Functional Security Requirements
	■ ensure that the same pseudonym is not used for different Relying Parties where the applicable pseudonym mechanism requires Relying Party-specific pseudonyms; ■ provide the generated pseudonym to the requesting Relying Party, either standalone or together with the requested PID or electronic attestation; ■ ensure that the pseudonym is generated and presented only after the Relying Party has been authenticated and the User has approved the applicable transaction; ■ protect pseudonyms against unauthorized disclosure, modification and replacement; and ■ allow the User to manage pseudonyms in accordance with the applicable technical specifications. ■ store the cryptographic information required for pseudonym generation and authentication in the WSCA/WSCD; ■ ensure that the same pseudonym is not disclosed to different Relying Parties unless explicitly permitted by the applicable pseudonym mechanism and User choice; ■ allow the User to view, select, manage and delete pseudonyms; ■ provide the User with sufficient information to distinguish pseudonyms associated with different Relying Parties; ■ protect pseudonym authentication data against unauthorized disclosure, modification, replacement and deletion; ■ ensure that User-assigned aliases or other information used by the User to manage pseudonyms are not disclosed to Relying Parties; and ■ ensure that pseudonyms contain sufficient entropy and are generated using the cryptographic mechanisms specified by the applicable technical specifications. The TOE shall ensure that pseudonym generation and presentation do not unnecessarily disclose the User's legal identity or enable the User to be tracked across different Relying Parties. The TOE shall support User-chosen and managed pseudonyms where required by the applicable ARF and technical specifications.
FSR_REVOCATION	The TOE shall regularly check the validity and revocation status of the security-relevant data objects used by the Wallet Unit, including: ■ PIDs and electronic attestations; ■ the Wallet Instance Attestation (WIA); ■ Key Attestations (KA); ■ the WSCD used by the Wallet Unit; and ■ the Wallet Unit, where its revocation status is available. The TOE shall perform such checks before using a data object for a security-sensitive operation where required by the applicable ARF and technical specifications. The TOE shall use the applicable revocation or status information provided by the authorised provider and shall verify the authenticity and integrity of that information before relying on it. If the TOE determines that a PID, electronic attestation, WIA, KA, WSCD, or Wallet Unit has been revoked or suspended, the TOE shall: ■ prevent the affected data object or security function from being used where required by the applicable ARF and technical specifications; and ■ inform the User of the relevant revocation or suspension and its consequences. If the TOE cannot obtain or validate the required revocation or status information, it shall follow the applicable offline/failure-handling rules and shall not incorrectly treat an unverifiable status as valid where this would compromise the security of the Wallet Unit.

FSR_ID	Functional Security Requirements
FSP_Q_SIGNATURE	**Qualified Electronic Signature and Seal Creation** The Wallet Unit shall support qualified electronic signature and seal creation using a **local, external or remotely managed Qualified Signature Creation Device (QSCD)**, as applicable. The Wallet Unit shall: ■ securely interface with the applicable Signature Creation Application (SCA) and QSCD; ■ ensure that signature or seal creation is performed only for the data approved by the User; ■ protect the integrity and confidentiality of the data and transaction information exchanged with the SCA/QSCD; ■ ensure that the signature or seal creation request is bound to the data approved by the User and cannot be replaced or manipulated; ■ where a remote QSCD is used, verify that the signature creation service is provided by an authorised QTSP, as applicable; and ■ provide the User with the result of the signature or seal creation, including notification of a failed operation. The WI-(TOE) shall support the mandatory signature and seal formats and interfaces applicable to the selected signature creation mechanism.
FSR_W2W	**Wallet-to-Wallet Interaction** The WI-(TOE) supports Wallet-to-Wallet (W2W) interactions between Wallet Units in accordance with the applicable ARF and technical specifications. The TOE shall: ■ establish an authenticated and protected communication channel with the other Wallet Unit; ■ verify the authenticity of the other Wallet Unit before processing a W2W request; ■ ensure that the User is informed of the requested attributes or electronic attestations before disclosure; ■ require User approval before disclosing any attributes or electronic attestations to the other Wallet Unit; ■ ensure that only the attributes or electronic attestations approved by the User are disclosed; ■ protect the W2W transaction against eavesdropping, modification, replay and transaction manipulation; and ■ log the W2W transaction in accordance with FSR_LOG_DATA. The TOE shall support both **Holder Wallet Unit mode** and **Verifier Wallet Unit mode**, where these functions are supported by the Wallet Unit. The TOE shall: ■ enter Holder Wallet Unit mode only following explicit User selection; ■ enter Verifier Wallet Unit mode only following explicit User selection; ■ inform the User when entering Holder Wallet Unit mode that this mode is intended for interaction with another natural person using a Wallet Unit; ■ return to normal Wallet Unit operation when the Wallet Unit is closed or after the applicable period of inactivity; ■ support W2W attestation presentation only through the applicable ISO/IEC 18013-5 proximity communication mechanisms; ■ when acting as a Verifier Wallet Unit, authenticate the Verifier and obtain the Verifier's approval before sending a presentation request; ■ limit the rate at which presentation requests can be sent in accordance with the applicable technical specification; ■ when acting as a Verifier Wallet Unit, verify received attestations and display the verified attributes to the Verifier; ■ not persistently store received attestations or attributes when acting as a Verifier Wallet Unit; and

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

FSR_ID	Functional Security Requirements
	▪ when acting as a Holder Wallet Unit, authenticate the Verifier Wallet Unit before presenting the requested attributes, where the applicable cryptographic proof mechanism is available. If authentication of the Verifier Wallet Unit fails or cannot be performed, the TOE shall inform the User and shall prevent disclosure or allow the User to explicitly decide whether disclosure may proceed, in accordance with the applicable ARF and technical specification.
FSR_MIGRATION	**Wallet Unit Migration** The WI-(TOE) shall support migration of the Wallet Unit in accordance with the applicable ARF and Technical Specification 10. The TOE shall: ▪ maintain an up-to-date **Migration Object** containing the transaction log information and the information required to restore the PIDs and electronic attestations present in the Wallet Unit; ▪ ensure that the Migration Object does **not contain WIA or KA**; ▪ for each PID and device-bound attestation, store in the Migration Object the information required to request re-issuance, including the attestation type, the PID Provider or Attestation Provider and the applicable service supply points; ▪ ensure that the Migration Object does not contain attribute identifiers, attribute values or private keys associated with a PID or device-bound attestation; ▪ for non-device-bound attestations, support the migration mechanism defined by the applicable ARF, including, where applicable, storing the information required for direct transfer or re-issuance; ▪ allow the User to choose whether attribute identifiers and attribute values associated with a non-device-bound attestation are included in the Migration Object, where applicable; ▪ delete the corresponding entry from the Migration Object when the User deletes a PID or attestation from the Wallet Unit; ▪ store or export the Migration Object to an external or remote storage location supported by the Wallet Unit and selected by the User; ▪ protect the confidentiality, integrity and authenticity of the Migration Object and prevent its use by anyone other than the User; ▪ after installation of a new Wallet Instance, allow the User to import a Migration Object from a supported storage location; ▪ allow the User to select the PIDs and attestations to be restored; ▪ request re-issuance of selected PIDs and device-bound attestations from the respective PID Provider or Attestation Provider, using the normal issuance process; ▪ request issuance of the PID before requesting issuance of other selected attestations; ▪ for non-device-bound attestations, either restore the attestation directly from the Migration Object or request re-issuance, as applicable to the contents of the Migration Object; ▪ ask the User whether the transaction log contained in the Migration Object shall be restored and, where approved, restore the log and append subsequent transactions to it; and ▪ ensure that the migration and re-issuance processes use the applicable security mechanisms and interfaces defined for normal issuance. The TOE shall protect the Migration Object during creation, storage, export, import and processing against unauthorized disclosure, modification, replacement and replay.

FSR_ID	Functional Security Requirements
FSR_DPA_REPORT	**Data Protection Authority Reporting** The WI-(TOE) shall provide the User with a function to initiate a complaint concerning a suspicious or inappropriate Relying Party presentation request to the competent Data Protection Authority (DPA), in accordance with the applicable ARF and national requirements. The TOE shall: ■ provide access to the applicable Data Protection Authority reporting function through the Wallet Instance; ■ allow the User to identify the relevant Relying Party and presentation request; ■ allow the User to provide the information required to substantiate the complaint; ■ protect the confidentiality and integrity of the report during preparation and transmission; and ■ record the submitted report in the transaction log so that it can subsequently be presented to the User.
FSR_TRST_MRK	**EU Digital Identity Wallet Trust Mark** The WI-(TOE) displays the official **EU Digital Identity Wallet Trust Mark** in the User interface. The Trust Mark view shall: ■ display the official Trust Mark graphics and/or logo; ■ provide information about the certification status of the Wallet Solution; and ■ provide links to the official list of certified EUDI Wallet Solutions and the certification information of the Wallet Solution. The TOE shall obtain and protect the Trust Mark information provided by the Wallet Provider and shall ensure that the displayed Trust Mark information cannot be modified by an unauthorized party. The TOE shall ensure that the Trust Mark and related certification information are only displayed for the applicable Wallet Solution and Wallet Instance.
FSR_LOG_DATA	**Transaction and Operational Logging** The WI-(TOE) shall maintain **transaction logs** for transactions with Wallet-Relying Parties and other Wallet Units, including unsuccessful transactions and electronic signing and sealing transactions. The transaction logs shall contain, as applicable: ■ the date and time of the transaction; ■ the name, contact details and unique identifier of the corresponding Wallet-Relying Party and the Member State in which it is established, or the relevant information from the Wallet Unit Attestation for transactions with another Wallet Unit; ■ the type of data requested and presented; and ■ the reason for an unsuccessful transaction. The TOE shall also log reports sent by the User to a Data Protection Authority through the Wallet Unit. The TOE shall protect transaction logs against unauthorized access, modification and deletion and shall ensure their confidentiality, integrity and authenticity. The TOE shall ensure that transaction logs do not contain the actual values of attributes presented, unless explicitly required by applicable legislation. The TOE shall allow the User to access and export the transaction logs in accordance with the applicable technical specifications. The TOE shall allow the User to delete individual transaction-log entries, where permitted by applicable legislation, and shall inform the User before logs are deleted when deletion is required. Transaction logs shall remain available for the period required by applicable Union or national law. Where access to transaction logs by the Wallet Provider is necessary for the provision of Wallet services, the TOE shall only permit such access based on the **explicit prior consent of the User**, as required by the applicable Regulation.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

FSR_ID	Functional Security Requirements
	The TOE shall additionally maintain operational and maintenance logs where required for the secure operation and maintenance of the Wallet Instance. Such logs shall be minimized and shall not contain the actual values of attributes presented.
FSR_DATA_DELETION	**User Data Deletion Requests** The WI-(TOE) shall support User-initiated requests for deletion of User attributes previously presented to Relying Parties in accordance with the applicable ARF and technical specifications. The TOE shall: ▪ allow the User to select one or more Relying Parties for which an attribute deletion request shall be submitted; ▪ generate and transmit the deletion request using the applicable protocol and security mechanisms; ▪ ensure that the deletion request identifies the relevant Relying Party and the data to which the request applies; ▪ protect deletion requests against unauthorized modification and replacement; and ▪ log the initiation and outcome of deletion requests in accordance with FSR_LOG_DATA.

3.2 WI-(TOE) Resilience Security Requirements

The following section defines the **Resilience Security Requirements (RSRs)** for the WI-(TOE). The RSRs have been split into two tables, ***Table*** 11 specifies the **security measures that are mandatory** to ensure that the protections required for each asset are not compromised, see list of assets in ***Table*** 1. ***Table*** 12 specifies **best-practice requirements** that can help developers protect the assets and security mechanisms sufficiently to meet the expected protection.

While the FSRs specify the security functionality provided by the WI-(TOE), the RSRs define resilience and defensive measures that support the secure implementation and continued operation of that functionality in the presence of active attacks or a compromised execution environment.

Table 11 ***— Mandatory Resilience Security Requirements***

RSR_ID	Mandatory Resilience Security Requirements
RSR_USERDEVICE	**Minimum User Device Security Platform** The User Device hosting the Wallet Instance shall provide a hardware-supported security platform consisting, at minimum, of a Trusted Execution Environment (TEE) or Secure Element (SE), integrated with a security-maintained operating system, to provide the security services required by the Wallet Instance. The TOE shall check that the User Device and OS shall at minimum: ▪ provide a **TEE or SE** providing an isolated execution and storage environment for security-relevant Wallet Instance assets; ▪ ensure that security-relevant device-bound private keys used by the Wallet Instance **cannot be exported or cloned** from the TEE/SE; ▪ ensure that cryptographic operations using such keys are performed within the protected environment and that the key material is not exposed to the normal OS or other applications; ▪ provide **hardware-supported isolation** between the protected security environment, the OS kernel and user applications; ▪ provide a **trusted boot mechanism** that ensures that only authenticated and authorized firmware and OS components are executed with privileged access;

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

RSR_ID	Mandatory Resilience Security Requirements
	▪ provide **OS-enforced application isolation/sandboxing** preventing one application from accessing another application's data, memory or processes without authorization; ▪ provide protection against **unauthorized privilege escalation**, including execution of unauthorized code with OS or kernel privileges; ▪ provide mechanisms for **secure and timely security updates** of the OS, firmware and security-critical platform components; ▪ provide **hardware-backed integrity/attestation mechanisms** allowing the security state of relevant platform components to be established; and ▪ provide a **secure device-access mechanism** protecting the User Device and the protected security environment against unauthorized local access.
RSR_PIN_ENTRY	**Secure PIN Entry** The WI (TOE) shall protect the knowledge-based authentication factor used for RUAP against **unauthorized disclosure, modification and reuse** During PIN entry, the TOE shall provide protection against: ▪ **screen capture and recording;** ▪ **overlay and tapjacking attacks;** ▪ side-channel leakage; ▪ accessibility-service abuse; ▪ malware attempting to **intercept or manipulate the PIN**; and ▪ unauthorized modification or substitution of the PIN-entry interface. The PIN shall **not be persistently stored or logged** by the TOE. Any PIN and security-sensitive data derived from it shall be **securely cleared immediately after use** and shall not remain recoverable in memory, temporary storage, caches or other TOE-controlled storage. The TOE shall ensure that the PIN is processed **only for the intended authentication operation** and shall not expose it to third-party applications or unnecessary platform services.
RSR_SCRN_PRT	**Protection of Sensitive User Interface Data** The WI (TOE) shall protect sensitive information presented through the User Device graphical user interface against **unauthorized capture, modification or disclosure**. At a minimum, the TOE shall: ▪ **prevent or appropriately restrict screenshots and screen recording** while sensitive information is displayed; ▪ protect sensitive screens against **overlay, tapjacking and other UI-injection attacks**; ▪ prevent sensitive information from being unnecessarily exposed through the **system clipboard**; ▪ prevent unauthorized accessibility services from accessing sensitive information or user input; ▪ prevent sensitive information from being unnecessarily exposed through **notifications, previews or other system UI**; ▪ provide secure handling of authentication input, including support for platform-provided secure authentication mechanisms where applicable; and ▪ verify that an appropriate **secure device lock mechanism** is configured before enabling security-critical functionality.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

RSR_ID	Mandatory Resilience Security Requirements
RSR_OPER_MOD	**Secure Operating Mode** The WI (TOE) shall establish that it is operating in a **security state supported by the Wallet Instance**. At a minimum, the TOE shall detect conditions indicating that the security properties of the underlying User Device or OS may have been compromised, including **rooting/jailbreaking, unauthorized developer or debug mode, execution in an emulator or app-virtualization environment**, or other conditions that invalidate the security assumptions of the TOE. When such a condition is detected, the TOE shall **terminate or disable all security-critical Wallet Instance functionality** and shall not perform security-critical operations until an acceptable security state has been restored.
RSR_MON	**Security Monitoring and Attestation** The WI (TOE) shall collect security-relevant **monitoring and attestation information** required to detect compromise of the TOE or its supporting User Device and shall securely provide this information to the Wallet Provider. At a minimum, the monitoring information shall include, where applicable: ▪ **rooted/jailbroken status**; ▪ **emulator, virtualization and developer/debug status**; ▪ detected modification or tampering attempts, including modification of TOE code, runtime instrumentation and interference with TOE security-relevant interfaces; ▪ **TOE integrity status**; ▪ relevant User Device and platform security state; ▪ **TOE and supported platform versions**; and ▪ the **unique identifier** required to associate the information with the corresponding Wallet Instance. Monitoring and attestation information transmitted to the Wallet Provider shall be protected against **unauthorized disclosure, modification and replay**. It shall contain sufficient freshness information, such as a **timestamp and/or nonce**, to detect replay of previously generated evidence. The TOE shall not persistently retain monitoring information unless explicitly required for a security purpose.
RSR_NO_DEBG	**Protection Against Debugging and Instrumentation** The WI (TOE) shall not contain production debug functionality or debug artifacts that could weaken its security The TOE shall **detect and prevent security-critical operation** when an unauthorized debugger, runtime instrumentation framework or equivalent debugging mechanism is attached or enabled. Security controls shall not rely solely on the absence of visible debug functionality and shall protect against commonly available mechanisms for **bypassing such controls**.
RSR_MAC	**Mandatory Access Control and Application Isolation** The WI (TOE) shall operate within the **security and isolation mechanisms provided by the underlying operating system**. The TOE shall not bypass or weaken the OS mandatory access-control, application sandboxing, process-isolation or permission mechanisms. The TOE shall use the **principle of least privilege** and shall request only the platform permissions required for its intended security and functional purposes.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

RSR_ID	Mandatory Resilience Security Requirements
RSR_SEC_CLR	**Secure Clearing of Sensitive Data** The WI (TOE) shall securely clear sensitive assets from memory when they are no longer required. Secure clearing shall use an **explicit and effective mechanism** appropriate to the asset and execution environment, such as zeroization, and shall not depend solely on garbage collection, memory deallocation or eventual reuse of the memory. The implementation shall ensure that the clearing operation is **not removed, optimized away or otherwise altered** by the compiler, runtime environment or platform. Where sensitive data may exist in multiple representations or memory areas, all relevant representations shall be appropriately cleared.
RSR_DAT_SAN	**Sanitization of Sensitive Data** The TOE shall sanitize **sensitive assets, PID attributes and other security-relevant personal data** from temporary or secondary storage when they are no longer required. This requirement shall apply, as applicable, to: ▪ **migration data;** ▪ **backups;** **logs;** ▪ **caches;** ▪ **temporary files;** ▪ operational data; ▪ crash reports; ▪ diagnostic data; and ▪ other data generated or retained during operation. Sanitization shall ensure that sensitive assets and PID attributes **cannot subsequently be recovered or disclosed** through such data, unless their retention is explicitly required for an authorized security or operational purpose and appropriate protection is applied.
RSR_KEY_MNG	**Secure Key Management** The WI (TOE) shall implement key-management mechanisms appropriate to the **security strength and intended use** of each cryptographic key. At a minimum: ▪ secret and private keys shall be generated, imported or injected only through mechanisms that protect their **confidentiality, integrity and authenticity**; ▪ keys requiring hardware-backed protection shall be generated or stored using the User Device's **hardware-backed keystore, TEE or SE**; ▪ security-relevant private keys shall **not be exportable** from their protected environment; ▪ private or secret keys that must temporarily exist outside the hardware-backed security environment shall be **unique per session** and shall provide appropriate protection against compromise of previous or subsequent sessions; ▪ key-encryption keys shall provide **security strength equal to or greater than** the keys they protect; ▪ public keys used for security-critical purposes shall be **integrity- and authenticity-protected and authenticated** before use; ▪ protection of keys in transit shall not rely solely on transport-channel security where compromise of the channel could expose or enable substitution of the key; ▪ keys shall have **explicitly defined usage restrictions** and shall not be used for purposes other than those for which they were generated; ▪ cryptographic keys shall be **securely destroyed or invalidated** when their lifecycle ends; and ▪ keys or security credentials dependent on the User Device's biometric enrollment state shall be invalidated or re-established when the relevant biometric enrollment changes, where required by the platform security model.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

RSR_ID	Mandatory Resilience Security Requirements
	Key-management mechanisms shall conform to the applicable **ECCG Agreed Cryptographic Mechanisms** and the security strength required by the TOE.
RSR_CRPTO	**Use of Approved Cryptography** The WI (TOE) shall use cryptographic mechanisms that provide **adequate security for the intended protection period and security strength** of the protected assets. At a minimum: ▪ cryptographic keys shall be generated using an **approved cryptographically secure random source**; ▪ key derivation shall use **approved key-derivation functions** and appropriate parameters and key sizes; ▪ encryption, decryption, digital signatures, MACs and hashing shall use **approved cryptographic algorithms and security parameters**; ▪ deprecated, broken or insufficiently secure algorithms and parameter sets shall not be used for security-critical functions; and ▪ cryptographic mechanisms shall be selected and configured in accordance with the applicable **ECCG Agreed Cryptographic Mechanisms**. Where a cryptographic mechanism required by the TOE is **not yet included in the ECCG Agreed Cryptographic Mechanisms**, its use shall be subject to an **appropriate security assessment and documented justification** demonstrating that the mechanism provides sufficient security for the intended purpose and security strength. Such a mechanism shall only be used where its security has been established in accordance with the **applicable certification or evaluation framework**, and shall be replaced or updated when an applicable ECCG mechanism becomes available. The use of cryptographic mechanisms not yet included in the ECCG Agreed Cryptographic Mechanisms shall therefore be **explicitly identified, justified and subject to approval by the applicable certification/evaluation process**.

RSR_RNG	**Cryptographically Secure Random Number Generation**
	The WI-(TOE) must generate random numbers using either:
	▪ A hardware RNG through an SE or TEE, and that RNG must be assessed as suitable for cryptographically strong random-number generation.
	▪ **Or** implement a DRBG compliant to NIST SP 800-90A or ISO/IEC 18031. This DRBG must be seeded by at least two sources of entropy:
	  ○ One external trusted source, (e.g. Wallet Provider HSM/WSCD)
	  ○ A source from the mobile device
	The combined entropy must be at least equal to the intended cryptographic key strength, and the entropy must be authenticated before use. Combining of entropy must preserve the entropy strength.
	The RNG (or DRNG) must be reseeded at least:
	▪ when the integrity/confidentiality of its state cannot be assured, such as at launch or after a halted state; and
	▪ at least every 24 hours of continuous operation.
	The reseeding process must add entropy to the existing RNG state, rather than simply replacing it.
	The DRNG shall generate random values with sufficient entropy and without statistically significant correlation, and shall be demonstrated to be fit for cryptographic purposes using an industry-recognized statistical test suite such as NIST SP 800-22 or AIS-31.

RSR_SEC_COD	**Secure Software Development and Supply Chain** The WI-(TOE) is developed using Secure coding practices, at a minimum: ▪ The TOE shall protect sensitive data and secrets by ensuring that secrets are not hardcoded in the TOE source code or package, and sensitive data is stored encrypted, and only in private storage. ▪ The TOE shall **securely handle untrusted or externally supplied data,** including intents, deep links, WebView content, and dynamically loaded code, such that: ○ untrusted content cannot access sensitive native functionality;

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

RSR_ID	Mandatory Resilience Security Requirements
	○ does not expose sensitive native functionality to WebView content and does not allow WebViews to load untrusted content; ○ deep links and intents cannot be abused to bypass security controls; ○ dynamic code is loaded safely, and only from trusted sources. ▪ The TOE shall use platform-provided security mechanisms, **including compiler-provided security features,** and shall not use deprecated APIs or functionality that could weaken the TOE's security. ▪ The TOE shall **adequately manage platform permissions** according to the principle of least privilege and shall use only trusted and appropriately maintained libraries for security-critical functionality. ▪ The TOE package shall not **contain malicious or unauthorized code** and third party components are regularly checked for known vulnerabilities.

Table** 12 — **Best Practices Resilience Security Requirements

RSR_ID	Recommended Resilience Security Requirements
RSR_OBF_COD	**Code and Resource Obfuscation** The WI-(TOE) may apply **code and resource obfuscation** to hinder reverse engineering and analysis of security-sensitive functionality and assets. At a minimum, obfuscation shall cover: ▪ **control-flow obfuscation** of code implementing security mechanisms or handling security-sensitive assets; ▪ **symbol renaming** of functions, methods, classes and other symbols associated with security-sensitive code; and **data and string obfuscation** of sensitive data, embedded secrets, strings and other security-relevant resources. Obfuscation shall also cover security-sensitive interfaces, including **interfaces to cryptographic functionality and inter-library calls**, where exposure of the implementation could facilitate compromise of the TOE. The TOE may additionally implement **anti-deobfuscation and anti-analysis techniques** to increase the resistance of the obfuscation against automated or manual reverse engineering.
RSR_PYL_ENC	**Application-Level Payload Encryption** The WI-(TOE) may protect security-sensitive assets transmitted between the TOE and external entities using **application-level encryption in addition to transport-layer protection**, where such assets require protection against compromise of the underlying communication channel. Application-level encryption shall provide **confidentiality and integrity protection** for the protected payload and shall ensure that the payload cannot be disclosed or modified without detection if the underlying transport protection is compromised. Cryptographic keys used for application-level payload protection shall be managed in accordance with the applicable **key-management and cryptographic requirements**.
RSR_ATT	**Device Attestation** The WI-(TOE) may support **device and platform attestation** sufficient to establish the relevant security state of the User Device on which the TOE operates. Where attestation evidence is used for security decisions, the TOE shall ensure that the evidence is **authentic, integrity protected and sufficiently fresh** and that it is associated with the corresponding User Device and Wallet Instance. The TOE shall not rely on attestation evidence that cannot be successfully authenticated, integrity verified or associated with the applicable device.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

RSR_ID	Recommended Resilience Security Requirements
RSR_INTG	**TOE Integrity Verification** The WI-(TOE) may periodically verify the **integrity and authenticity of its security-relevant software and resources**, including executable code, security-sensitive configuration and other resources required for secure operation. The integrity verification mechanism shall detect **unauthorized modification, replacement or removal** of the verified code or resources. If the integrity or authenticity verification fails, the TOE shall **prevent security-critical functionality** and shall not continue operation using the affected code or resources.
RSR_FOCS	**Protection Against Loss of User Interface Focus** The WI-(TOE) may protect security-sensitive user interactions and displayed sensitive assets against **loss or unauthorized transfer of user-interface focus**. If the TOE detects that focus has been lost while the User is performing a security-sensitive action or while sensitive information is being displayed, the TOE shall: ▪ **terminate the affected security-sensitive operation;** ▪ prevent the operation from continuing without appropriate re-authentication or renewed User interaction, where required; and ▪ prevent sensitive information from remaining exposed after focus has been lost.
RSR_DYN_TL	**Protection Against Dynamic Instrumentation and Runtime Manipulation** The WI-(TOE) may implement **countermeasures against runtime instrumentation and dynamic analysis** techniques that could be used to compromise security-sensitive assets or security functionality. Where such countermeasures are implemented, they should detect or resist techniques including **runtime code injection, function hooking, API interception, dynamic debugging and manipulation of security-relevant execution flows**. Where a detected instrumentation condition could compromise a security-critical operation, the TOE should **terminate the affected operation**.

A Requirement Coverage Rationale

This appendix demonstrates the coverage and completeness of the identified security requirements. It shows how the Functional Security Requirements (FSRs) and Resilience Security Requirements (RSRs) address the Wallet Unit lifecycle, identified assets, and their required security protection.

A.1 Coverage of all FSR through the life cycle

Table 13 maps each lifecycle and trust-relation described in section 1.3 to the FSRs and assets addressed by that functionality. This demonstrates that the identified security functionality and assets cover the full Wallet Unit lifecycle.

***Table** 13 — Coverage of all functionalities through the life cycle*

Lifecycle / trust-relation	Related FSR(s)	Related assets
1.3.1 Wallet Instance Installation		
1.3.1.1 Mutual Authentication Channel	FSR_SEC_CHANNEL	A.CERT, A.MUT_KEY, A.INIT, A.CMD
1.3.1.2 Initial Data Requested by Wallet Provider	FSR_WI_INSTALL, FSR_SEC_CHANNEL	A.INIT, A.MON, A.WIID
1.3.2 Wallet Activation		
1.3.2.1 Wallet Provider User Account	FSR_WI_INSTALL, FSR_SEC_CHANNEL, FSR_REVOCATION	A.WIID, A.INIT
1.3.2.2 User Authentications	FSR_LOCAL_AUTH, FSR_REMOTE_AUTH, FSR_SEC_CHANNEL	A.PIN, A.PUK, A.KEYAUTH, A.POSS, A.UAE, A.ARD, A.NONCE, A.CMD, A.MUT_KEY, A.KEYAUTH, A.PUK
1.3.2.3 Initial Generation of WIA and KA	FSR_WIA, FSR_KEY_ATTEST, FSR_REMOTE_AUTH, FSR_SEC_CHANNEL	A.WIA, A.KA, A.CMD, A.NONCE, A.UAE
1.3.2.3.1 Key Attestations (KA)	FSR_KEY_ATTEST, FSR_REMOTE_AUTH, FSR_SEC_CHANNEL, FSR_TRUST_ANCHOR	A.KA, A.MUT_KEY, A.CERT, A.ANCHOR, A.NONCE, A.CMD
1.3.2.3.2 Wallet Instance Attestation (WIA)	FSR_WIA, FSR_REMOTE_AUTH, FSR_SEC_CHANNEL, FSR_TRUST_ANCHOR	A.WIA, A.WIID, A.MUT_KEY, A.ANCHOR, A.NONCE
1.3.3 PID/Attestation Provider Authentication and Issuance		
1.3.3.1 PID/Attestation Provider Authentication to Wallet Unit	FSR_ISSUANCE, FSR_ACC_REG_CERT, FSR_TRUST_ANCHOR, FSR_SEC_CHANNEL	A.CERT, A.ANCHOR,
1.3.3.2 Wallet Unit Authentication to PID/Attestation Provider	FSR_WIA, FSR_KEY_ATTEST, FSR_ISSUANCE, FSR_SEC_CHANNEL, FSR_PROTOCOL	A.WIA, A.KA, A.MUT_KEY, A.NONCE, A.CMD
1.3.3.3 PID/Attestation Generation	FSR_ISSUANCE, FSR_KEY_ATTEST, FSR_WIA, FSR_TRUST_ANCHOR	A.PIDATT, A.KA, A.WIA, A.MUT_KEY, A.CERT, A.ANCHOR
1.3.4 Credential (PIDs/Attestation) Disclosure		
1.3.4.1 TOE Authenticates Relying Party	FSR_ACC_REG_CERT, FSR_TRUST_ANCHOR, FSR_SEC_CHANNEL	A.CERT, A.ANCHOR,

Lifecycle / trust-relation	Related FSR(s)	Related assets
1.3.4.2 User Approval and Selective Disclosure	FSR_SELECT_DISCLO, FSR_PRESENTATION, FSR_LOCAL_AUTH, FSR_REMOTE_AUTH, FSR_PROTOCOL	A.PIDATT, A.PRES, A.NONCE, A.MUT_KEY, A.UAE
1.3.4.3 Pseudonyms	FSR_PSEUDONYM, FSR_LOCAL_AUTH, FSR_REMOTE_AUTH	A.PSEUD, A.MUT_KEY, A.CMD, A.UAE
1.3.5 Revocation/Deletion	FSR_REVOCATION, FSR_DATA_DELETION, FSR_WI_INSTALL	A.WIA, A.KA, A.PIDATT, A.MUT_KEY, A.LOG, A.WIID
1.3.6 Additional Wallet Instance Security Functionality		
1.3.6.1 Qualified Electronic Signature and Seal	FSP_Q_SIGNATURE, FSR_LOCAL_AUTH, FSR_REMOTE_AUTH, FSR_SEC_CHANNEL, FSR_LOG_DATA	A.QESDATA, A.UAE, A.CERT, A.ANCHOR, A.MUT_KEY, A.LOG
1.3.6.2 Wallet-to-Wallet (W2W)	FSR_W2W, FSR_PROTOCOL, FSR_LOCAL_AUTH, FSR_SEC_CHANNEL, FSR_LOG_DATA	A.PIDATT, A.PRES, A.CERT, A.ANCHOR, A.NONCE, A.MUT_KEY, A.LOG
1.3.6.3 Migration	FSR_MIGRATION, FSR_ISSUANCE, FSR_DATA_DELETION, FSR_LOG_DATA	A.MIGR, A.PIDATT, A.LOG, A.WIA, A.KA, A.MUT_KEY
1.3.6.4 DPA Reporting	FSR_DPA_REPORT, FSR_LOG_DATA, FSR_SEC_CHANNEL	A.LOG, A.PIDATT, A.CERT, A.PRES
1.3.6.5 Trust Mark	FSR_TRST_MRK, FSR_TRUST_ANCHOR	A.TRUSTM, A.ANCHOR, A.CERT
1.3.6.6 Logging as an Explicit Lifecycle/Security Function	FSR_LOG_DATA	A.LOG

A.2 Coverage of the attack scenarios

Table 14 demonstrates how the identified security requirements **address the defined attack scenarios**. For each attack scenario, the relevant Functional Security Requirements (FSRs), Resilience Security Requirements (RSRs), and Environmental Security Objectives (OEs) are identified. The rationale explains how these requirements prevent or mitigate the attack and, where applicable, identifies cases where the protection is provided entirely by the environment. This provides the coverage argument that the identified requirements adequately address the security threats and required asset protection.

Table 14 ***— Coverage of the Required Asset protection***

Attack scenario	FSR	RSR	OE	Rationale
AS.UI.01 — Fake WSCA/WSCD during channel establishment	FSR_WI_INSTALL, FSR_SEC_CHANNEL, FSR_WIA	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.WP, OE.WSCA/WSCD, OE.APP_DISTRIBUTION	**FSR_WI_INSTALL** establishes the initial trusted relationship with the Wallet Provider and verifies the security environment before activation. **FSR_SEC_CHANNEL** requires mutual authentication and integrity protection of the WI–WSCA communication, preventing an attacker from substituting an unauthenticated WSCA/WSCD. **FSR_WIA** ensures that the Wallet Instance Attestation is authentic, valid and bound to the corresponding

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Attack scenario	FSR	RSR	OE	Rationale
				Wallet Instance with in the 24h time window, limiting the use of fraudulent attestation information. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.WP** protects the Wallet Provider trust relationship and WIA-related signing material, while **OE.WSCA/WSCD** provides the trusted endpoint and requires an authenticated protected relationship with the WI. **OE.APP_DISTRIBUTION** ensures that only authentic, signed versions of the Wallet Instance are installed, preventing an attacker from distributing a modified Wallet Instance that connects to an attacker-controlled WSCA/WSCD.
AS.UI.02 — Cloning of Wallet Instance authentication factors	FSR_WI_INSTALL, FSR_SEC_CHANNEL, FSR_LOCAL_AUTH, FSR_REMOTE_AUTH, FSR_WIA	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.USER, OE.WSCA/WSCD , OE.USER_DEVICE	**FSR_LOCAL_AUTH** requires possession of the User Device together with a second authentication factor before security-sensitive WI operations. **FSR_REMOTE_AUTH** requires two independent authentication factors for WSCA/WSCD access and protects the authentication process against guessing, replay and manipulation. **FSR_SEC_CHANNEL** protects the authentication exchange between the WI and WSCA/WSCD. **FSR_WI_INSTALL** ensures that the required authentication mechanisms are established during trusted activation. **FSR_WIA** ensures that the Wallet Instance Attestation is authentic, valid and bound to the corresponding Wallet Instance with in the 24h time window, limiting the use of fraudulent attestation information. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.USER** requires the User to retain control of the authentication factors.

Attack scenario	FSR	RSR	OE	Rationale
				OE.USER_DEVICE ensures that the possession factor is held in a hardware-backed keystore from which it cannot be exported or cloned. **OE.WSCA/WSCD** independently requires successful authentication before sensitive cryptographic operations.
AS.UI.03 — Theft of device-bound private key	-	-	OE.WSCA/WSCD, OE.PROVIDER, OE.RP	In the remote-HSM configuration, the critical private key is exclusively resident in the WSCA/WSCD. **OE.WSCA/WSCD** therefore provides the primary protection against private-key extraction and performs the required cryptographic operation without exposing the private key. **OE.PROVIDER** verifies the WIA/KA and the associated proof of possession during issuance, while **OE.RP** verifies the proof of possession during presentation.
AS.UI.04 — Bypass of User authentication	FSR_WI_INSTALL, FSR_LOCAL_AUTH, FSR_REMOTE_AUTH, FSR_SEC_CHANNEL, FSR_WIA	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.USER, OE.WSCA/WSCD, OE.WP, OE.WP, OE.USER_DEVICE, OE.APP_DISTRIBUTION	**FSR_LOCAL_AUTH** and **FSR_REMOTE_AUTH** require User authentication before security-sensitive operations. **FSR_SEC_CHANNEL** ensures that protected operations can only be performed through the authenticated WI–WSCA/WSCD channel. **FSR_WI_INSTALL** prevents activation where the required security conditions are not met. **FSR_WIA** provides evidence of the security status and identity of the Wallet Instance. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.WSCA/WSCD** provides an independent authorization boundary for protected cryptographic operations. **OE.USER_DEVICE** provides the OS-level authentication, trusted boot and application isolation on which local authentication relies. **OE.APP_DISTRIBUTION** prevents a modified Wallet Instance with removed or weakened authentication checks from being installed.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Attack scenario	FSR	RSR	OE	Rationale
AS.UI.05 — Manipulation of WIA/KA binding	-	-	OE.WP, OE.PROVIDER, OE.RP	This attack is primarily mitigated by the **environment**. **OE.WP** protects the issuance and signing of WIA/KA information. **OE.PROVIDER** verifies the authenticity and integrity of the WIA/KA and verifies that the attested public key is associated with the corresponding protected private key. **OE.RP** independently verifies the resulting proof of possession during presentation. Consequently, an attacker cannot successfully substitute a WIA/KA binding with an attacker-controlled key without failing the environmental verification performed during issuance or presentation.
AS.UI.06 — Replay or manipulation of authentication data	FSR_WI_INSTALL, FSR_LOCAL_AUTH, FSR_REMOTE_AUTH, FSR_SEC_CHANNEL, FSR_WIA	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.WSCA/WSCD, OE.USER_DEVICE	**FSR_LOCAL_AUTH** and **FSR_REMOTE_AUTH** require User authentication before security-sensitive operations. **FSR_SEC_CHANNEL** protects WI–WSCA communications against replay and modification. **FSR_REMOTE_AUTH** additionally requires fresh and transaction-bound authentication evidence, preventing captured authentication data from being reused. **FSR_WIA** provides evidence of the security status and identity of the Wallet Instance. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.WSCA/WSCD** provides an independent requirement that authentication and cryptographic operations are protected against replay and manipulation. **OE.USER_DEVICE** ensures that the possession-factor key used to create the authentication evidence cannot be extracted and reused outside the User Device.

Attack scenario	FSR	RSR	OE	Rationale
AS.UI.07 — Bypass via Migration	FSR_MIGRATION, FSR_WI_INSTALL, FSR_SEC_CHANNEL, FSR_LOCAL_AUTH, FSR_WIA, FSR_KEY_ATTEST FSR_REMOTE_AUTH	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.WSCA/WSCD	**FSR_MIGRATION** directly mitigates this attack by securing Wallet migration and restoration against unauthorized access. **FSR_WI_INSTALL**, **FSR_SEC_CHANNEL**, **FSR_LOCAL_AUTH** and **FSR_REMOTE_AUTH** support secure destination verification, protected communication and required User authentication. **FSR_WIA** and **FSR_KEY_ATTEST** ensure the Wallet Instance and its cryptographic key bindings can be verified. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.WSCA/WSCD** provides an independent requirement that authentication and cryptographic operations are protected against replay and manipulation.
AS.LPD.01 — Fake Relying Party	FSR_ACC_REG_CERT, FSR_TRUST_ANCHOR, FSR_PROTOCOL, FSR_SELECT_DISCLO, FSR_PRESENTATION	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.RP	**FSR_ACC_REG_CERT** requires the RP Access and Registration Certificates to be authenticated and validated before disclosure, verifies that the request is associated with the authenticated RP, and ensures that the RP and Service information shown to the User corresponds to the authenticated certificates. **FSR_TRUST_ANCHOR** ensures that the certificates and signed objects are validated against authorised and applicable trust anchors. **FSR_PROTOCOL** protects the presentation flow against phishing, relay, session-hijacking, replay and transaction manipulation and ensures that security-relevant messages are bound to the transaction and session. **FSR_SELECT_DISCLO** prevents disclosure until the applicable disclosure policies have been satisfied and the User has approved the presentation. **FSR_PRESENTATION** ensures that credentials are only presented to an authenticated and authorised entity. **The resilience security requirements in section 3.2 collectively ensure that the**

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Attack scenario	FSR	RSR	OE	Rationale
				assets are adequately protected. **OE.RP** requires the RP to authenticate using the applicable certificates and establish presentation trust.
AS.LPD.02 — Manipulation of presentation request	FSR_PROTOCOL, FSR_ACC_REG_CERT, FSR_SELECT_DISCLO, FSR_TRUST_ANCHOR, FSR_PRESENTATION, FSR_MDOC	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.USER, OE.RP	**FSR_PROTOCOL** verifies the authenticity and integrity of security-relevant protocol messages, binds them to the applicable transaction and session, and prevents their replay or reuse. **FSR_ACC_REG_CERT** authenticates the RP and verifies that the request is associated with the authenticated RP and is consistent with the applicable registration information. **FSR_TRUST_ANCHOR** provides the trusted basis for validating the RP certificates and other signed objects used during request processing. **FSR_SELECT_DISCLO** ensures that only attributes and attestations approved by the User are included in the presentation and prevents disclosure when the required approval or disclosure policy conditions are not satisfied. **FSR_PRESENTATION** ensures that only authentic, valid and authorised credentials are presented. **FSR_MDOC** provides the corresponding protections for mdoc requests, including verification of request authenticity and integrity and binding of security-relevant mdoc messages to the presentation session and transaction. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.RP** requires the RP to authenticate and verify the resulting transaction.
AS.LPD.03 — Bypass of User approval	FSR_LOCAL_AUTH, FSR_REMOTE_AUTH, FSR_PRESENTATION, FSR_SELECT_DISCLO, FSR_TRUST_ANCHO	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR,	OE.USER, OE.RP, OE.APP_DISTRIBUTION	**FSR_LOCAL_AUTH** requires User authentication before security-sensitive actions. **FSR_REMOTE_AUTH** provides the additional WSCA/WSCD-level authentication and ensures that protected assets cannot be used without the required User

Attack scenario	FSR	RSR	OE	Rationale
	R, FSR_PROTOCOL, FSR_ACC_REG_CERT	RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD		authentication factors. **FSR_PRESENTATION** prevents presentation unless the applicable authentication and authorization requirements have been satisfied. **FSR_SELECT_DISCLO** explicitly requires User approval before disclosure and ensures that the approval applies to the actual presentation transaction and transmitted attributes. **FSR_TRUST_ANCHOR** provides the trusted basis for validating the security-relevant certificates and signed objects used in the presentation process. **FSR_PROTOCOL** ensures that the transaction presented to the User corresponds to the transaction actually processed and transmitted, and rejects the transaction when the required security checks fail. **FSR_ACC_REG_CERT** ensures that the authenticated RP and Service are identified before the User approves the presentation. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.USER** is responsible for reviewing the information presented and approving the intended disclosure, while **OE.RP** provides the authenticated RP identity used by the WI. **OE.APP_DISTRIBUTION** prevents a modified Wallet Instance that skips the approval step from being installed.
AS.LPD.04 — Manipulation of User approval information	FSR_ACC_REG_CERT, FSR_SELECT_DISCLO, FSR_PRESENTATION, FSR_TRUST_ANCHOR, FSR_PROTOCOL	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.USER, OE.RP, OE.APP_DISTRIBUTION	**FSR_ACC_REG_CERT** authenticates the RP and Service and requires the RP and Service information presented to the User to correspond to the validated RP certificates. **FSR_SELECT_DISCLO** ensures that only the attributes and attestations actually approved by the User are included in the presentation and that approval applies to the actual transaction. **FSR_PRESENTATION** ensures that only valid credentials are presented to the authenticated and authorised recipient. **FSR_TRUST_ANCHOR** ensures

Attack scenario	FSR	RSR	OE	Rationale
				that the RP certificates and other signed objects used to establish the displayed identity are validated against trusted anchors. **FSR_PROTOCOL** binds the security-relevant messages and User-visible transaction to the actual presentation transaction, preventing a manipulated request or approval from being used for another transaction. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.USER** is responsible for reviewing the information presented and approving the intended disclosure, while **OE.RP** provides the authenticated RP identity used by the WI. **OE.APP_DISTRIBUTION** prevents a modified Wallet Instance that displays falsified approval information from being installed.
AS.LPD.06 — Manipulation of selective disclosure	FSR_SELECT_DISCLO, FSR_PRESENTATION, FSR_TRUST_ANCHOR, FSR_PROTOCOL, FSR_MDOC, FSR_ACC_REG_CERT	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.USER, OE.RP	**FSR_SELECT_DISCLO** is the primary control: it processes the applicable disclosure policies, requires User approval, and ensures that only the attributes and attestations approved by the User are included in the presentation. **FSR_PRESENTATION** ensures that the resulting presentation uses authentic, valid credentials and is only provided to an authenticated and authorised recipient. **FSR_TRUST_ANCHOR** provides the trusted basis for validating the certificates and signed objects used in the presentation process. **FSR_PROTOCOL** ensures that the User approval, request and resulting presentation are bound to the same transaction and prevents replay or reuse of security-relevant messages. **FSR_MDOC** applies equivalent selective-disclosure, request-integrity and transaction-binding protections to mdoc presentations. **FSR_ACC_REG_CERT** ensures that the disclosure policy is evaluated against the

Attack scenario	FSR	RSR	OE	Rationale
				authenticated RP and Service information and that the RP is identified before approval. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.USER** is responsible for reviewing the information presented and approving the intended disclosure, while **OE.RP** provides the authenticated RP identity used by the WI.
AS.LPD.07 — Presentation data interception	FSR_PROTOCOL, FSR_PRESENTATION	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.USER, OE.RP	**SR_PROTOCOL** protects the presentation protocol against manipulation, replay and session-hijacking and ensures that security-relevant messages are authenticated, integrity-protected and bound to the applicable transaction and session. **FSR_PRESENTATION** protects stored credentials against unauthorized disclosure and ensures that a PID or attestation is only presented to an authenticated and authorised recipient after the applicable authorization requirements have been satisfied. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.USER** is responsible for reviewing the information presented and approving the intended disclosure, while **OE.RP** provides the authenticated RP identity used by the WI.
AS.LPD.08 — Replay or transaction manipulation	FSR_PROTOCOL, FSR_SELECT_DISCLO, FSR_PRESENTATION, FSR_MDOC, FSR_REVOCATION	RSR_USERDEVICE, RSR_PIN_ENTRY, RSR_SCRN_PRT, RSR_OPER_MOD, RSR_MON, RSR_NO_DEBG, RSR_MAC, RSR_SEC_CLR, RSR_DAT_SAN, RSR_KEY_MNG, RSR_CRPTO, RSR_RNG, RSR_SEC_COD	OE.USER, OE.RP	**FSR_PROTOCOL** directly addresses replay and transaction manipulation by authenticating security-relevant messages, binding them to the current transaction and session, preventing reuse, and rejecting the transaction when the required checks fail. **FSR_SELECT_DISCLO** ensures that User approval applies to the actual presentation transaction and to the attributes and attestations transmitted, preventing approval from being reused for a different disclosure. **FSR_PRESENTATION** ensures

Attack scenario	FSR	RSR	OE	Rationale
				that only authentic, valid and authorised credentials are presented and preserves their binding to the Wallet Unit. **FSR_MDOC** provides the corresponding replay, modification and transaction-binding protections for mdoc presentations. **FSR_REVOCATION** ensures that the validity and revocation status of credentials and other security-relevant objects are checked before security-sensitive use and prevents the use of revoked or suspended objects where required. **The resilience security requirements in section 3.2 collectively ensure that the assets are adequately protected.** **OE.USER** is responsible for reviewing the information presented and approving the intended disclosure, while **OE.RP** provides the authenticated RP identity used by the WI.

B Modular and functional Decomposition of the WI-(TOE)

The functional decomposition in section B.1 identifies the main security functions of the WI-(TOE) and in ***Table*** 16 it groups the Functional Security Requirements (FSRs) and Resilience Security Requirements (RSRs) according to their primary security purpose. It provides a structured view of the security functionality and can be used by the developer as a starting point for defining the functional design of the TOE and allocating the security requirements to appropriate components and interfaces.

Section B.2 provides a reference modular decomposition of the WI-(TOE) into functional software modules. The modular decomposition provides the developer with a starting point for describing the actual TOE design and identifying how the FSRs and RSRs are implemented. It does not prescribe a specific software architecture; the developer may structure the implementation differently, provided that the required security functionality is implemented.

For the evaluation laboratory, both decompositions provide a traceable basis for verification and targeted testing. The functional decomposition can be used to identify the security functionality to be assessed, while the modular decomposition can be used to identify the TOE modules implementing that functionality. This supports focused testing and change-impact analysis but does not replace the detailed design evidence or evaluation activities required by the applicable methodology.

Table 16 provides an overview of the expected modules, their primary functionality, and their relationship to the functional and resilience security requirements.

B.1 Functional decomposition

The functional decomposition in ***Table*** 15 identifies the main security functions provided by the WI-(TOE) and groups the Functional Security Requirements (FSRs) and Resilience Security Requirements (RSRs) according to their primary security purpose. This decomposition provides a structured view of the security functionality of the WI-(TOE) and supports the subsequent mapping of security requirements to security objectives, threats and assets.

Table 15 ***— Functional decomposition***

Functional Decomposition	Description
Framework	Core wallet framework services that provide the overall application structure, orchestration, user interface, trust mark functionality, lifecycle management, and integration between functional components.
Wallet Provider Interaction	Functions that manage interactions between the Wallet Instance and the Wallet Provider ecosystem, including onboarding, attestation management, activation, monitoring, trust establishment, and lifecycle management.
PID Provider Interaction	Functions that establish and maintain trusted interactions with PID Providers, Attestation Providers, eID schemes, and other credential issuers, including authentication and verification of external entities.

Functional Decomposition	Description
Issuance	Functions involved in requesting, receiving, validating, and storing PIDs and Attestations, including both online and offline issuance processes.
Presentation	Functions responsible for the disclosure of credentials and attributes to Relying Parties, including selective disclosure, verification of requests, user consent, and presentation protocols.
Remote Signing	Functions supporting remote electronic signature and seal services, including authentication, signature initiation, signature verification, and integration with remote QSCD services.
PID Data Model	Functions responsible for the creation, processing, storage, management, and protection of credentials, attributes, claims, and related wallet data structures.
WI/WSCA/WSCD Low Level Functions	Security-critical functions that implement low-level interactions between the Wallet Instance and supporting security components such as the operating system, secure keystores, WSCA, and WSCD, including local and remote authentication mechanisms.
Wallet Functions	User-facing wallet capabilities such as dashboards, transaction history, pseudonym management, credential management, privacy controls, and related wallet services.
Supporting Functions	Auxiliary services that support the operation, security, maintenance, backup, recovery, and protection of the Wallet Instance without directly implementing wallet business functionality.

B.2 Modular decomposition

Security Requirements (FSRs) and Resilience Security Requirements (RSRs) implemented by the module. ***Figure*** 2 presents the decomposition of the WI-(TOE) into its constituent software modules. The following sections provide an overview of the functionality of each module and identify the Functional

Figure 2 ***— Modular decomposition***

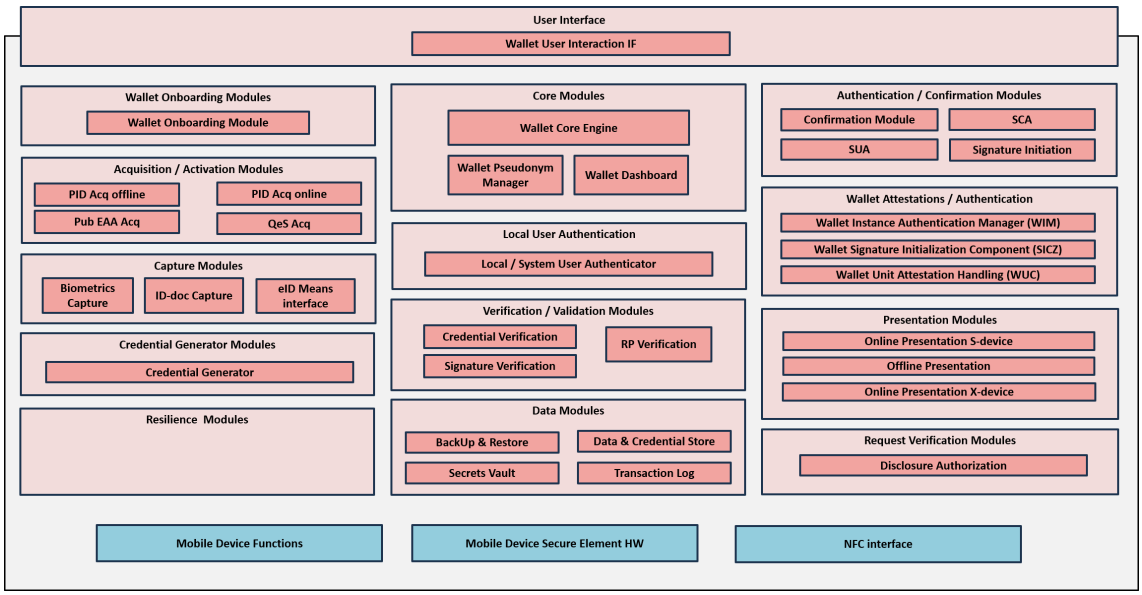


Table 16 provides a concise functional overview of each module and maps the modules to the functional security requirements defined in this Protection Profile. The table thereby

provides the link between the functional decomposition and the modular decomposition of the WI-(TOE).

Detailed descriptions of the individual modules are provided in [EUDI Oracle]. The module descriptions in [EUDI Oracle] further trace the requirements specified in regulation to the corresponding WI-(TOE) modules and their implemented security functionality.

***Table** 16 — WI-(TOE) decomposition into modules description*

Module	Description of Modules	Functional decomposition	Requirements	Assets
Wallet User Interaction IF	User interface visible on the User Device for login, authentication, consent, dashboard access and other Wallet functions.	Framework; Wallet Functions; WI/WSCA/WSCD Low Level Functions	**FSR_LOCAL_AUTH; FSR_REMOTE_AUTH; FSR_SELECT_DISCLO; FSR_PRESENTATION; FSR_ACC_REG_CERT; FSP_Q_SIGNATURE; FSR_TRST_MRK; FSR_REVOCATION; FSR_DATA_DELETION ; FSR_WI_INSTALL**	A.PIN; A.PUK; A.KEYAUTH; A.PIDATT; A.PRES; A.CERT; A.LOG; A.PSEUD; A.QESDATA; A.TRUSTM
Wallet Onboarding Module	Activates the first Wallet Instance and WSCA/WSCD, manages additional instances and deactivation.	Wallet Provider Interaction; WI/WSCA/WSCD Low Level Functions; Supporting Functions	**FSR_WI_INSTALL; FSR_SEC_CHANNEL; FSR_LOCAL_AUTH; FSR_REMOTE_AUTH; FSR_WIA; FSR_KEY_ATTEST; FSR_TRUST_ANCHOR**	A.INIT; A.WIID; A.MON; A.MUT_KEY; A.POSS; A.ARD; A.WIA; A.KA; A.CERT; A.ANCHOR
PID Acq Offline	Requests PID issuance via a physical registration desk.	PID Provider Interaction; Issuance; WI/WSCA/WSCD Low Level Functions	**FSR_ISSUANCE; FSR_PROTOCOL; FSR_TRUST_ANCHOR; FSR_WIA; FSR_KEY_ATTEST; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL**	A.PIDATT; A.CERT; A.ANCHOR; A.WIA; A.KA; A.NONCE; A.UAE; A.CMD; A.MUT_KEY
PID Acq Online	Requests PID issuance online from a PID Provider.	PID Provider Interaction; Issuance	**FSR_ISSUANCE; FSR_PROTOCOL; FSR_TRUST_ANCHOR; FSR_WIA; FSR_KEY_ATTEST; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL**	A.PIDATT; A.CERT; A.ANCHOR; A.WIA; A.KA; A.NONCE; A.UAE; A.CMD; A.MUT_KEY
Pub EAA Acq	Requests public EAA issuance online.	PID Provider Interaction; Issuance	**FSR_ISSUANCE; FSR_PROTOCOL; FSR_TRUST_ANCHOR; FSR_WIA; FSR_KEY_ATTEST; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL**	A.PIDATT; A.CERT; A.ANCHOR; A.WIA; A.KA; A.NONCE; A.UAE; A.CMD
Biometrics Capture	Captures biometric data for issuance, enrolment or presentation.	PID Provider Interaction; Issuance; Presentation; WI/WSCA/WSCD Low Level Functions	**FSR_ISSUANCE; FSR_LOCAL_AUTH; FSR_REMOTE_AUTH; FSR_SELECTIVE_DISCLO; FSR_PRESENTATION**	A.UAE; A.ARD; A.PIN, A.PUK, A.KEYAUTH; A.POSS; A.PIDATT

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Module	Description of Modules	Functional decomposition	Requirements	Assets
ID-doc Capture	Reads ID cards/passports during offline enrolment.	PID Provider Interaction; Issuance	**FSR_ISSUANCE; FSR_TRUST_ANCHOR**	A.PIDATT; A.CERT; A.ANCHOR; A.UAE
QeS Acq	Connects the Wallet to a remote Qualified Electronic Signature service.	Wallet Provider Interaction; Remote Signing; WI/WSCA/WSCD Low Level Functions	**FSP_Q_SIGNATURE; FSR_PROTOCOL; FSR_TRUST_ANCHOR; FSR_LOCAL_AUTH; FSR_REMOTE_AUTH; FSR_LOG_DATA**	A.QESDATA; A.CERT; A.ANCHOR; A.NONCE; A.UAE; A.LOG
eID Means Interface	Invokes eID identification and authentication services.	PID Provider Interaction; Wallet Provider Interaction	**FSR_ISSUANCE; FSR_PROTOCOL; FSR_TRUST_ANCHOR**	A.PIDATT; A.CERT; A.ANCHOR; A.NONCE
Wallet Core Engine	Orchestrates interfaces between Wallet components and the platform.	Framework; WI/WSCA/WSCD Low Level Functions; Supporting Functions	**FSR_WI_INSTALL; FSR_SEC_CHANNEL; FSR_PROTOCOL; FSR_W2W**	A.INIT; A.WIID; A.CMD; A.MUT_KEY; A.NONCE
Wallet Dashboard	User interface for data and log interaction.	Framework; Wallet Functions	**FSR_LOG_DATA; FSR_PRESENTATION; FSR_DATA_DELETION ; FSR_DPA_REPORT; FSR_PSEUDONYM; FSR_REVOCATION; FSR_LOCAL_AUTH**	A.PIDATT; A.LOG; A.PSEUD; A.CERT
Wallet Pseudonym Manager	Manages pseudonyms and their assignment to services.	Wallet Functions; PID Data Model; Presentation	**FSR_PSEUDONYM; FSR_SELECT_DISCLO; FSR_LOCAL_AUTH; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL**	A.PSEUD; A.NONCE; A.UAE; A.CMD
Local / System User Authenticator	PIN-based local Wallet unlock.	WI/WSCA/WSCD Low Level Functions; Wallet Functions	**FSR_LOCAL_AUTH**	A.PIDATT; A.LOG; A.PSEUD
Credential Verification	Validates attestations from Providers or Relying Parties.	PID Provider Interaction; Presentation; PID Data Model	**FSR_ISSUANCE; FSR_PRESENTATION; FSR_TRUST_ANCHOR; FSR_REVOCATION**	A.PIDATT; A.CERT; A.ANCHOR; A.NONCE
RP Verification	Validates RP certificates.	Presentation; PID Provider Interaction	**FSR_ACC_REG_CERT; FSR_TRUST_ANCHOR; FSR_REVOCATION; FSR_PROTOCOL**	A.CERT; A.ANCHOR; A.NONCE
Signature Verification	Verifies electronic signatures and seals.	Presentation; Remote Signing; PID Provider Interaction	**FSP_Q_SIGNATURE; FSR_TRUST_ANCHOR; FSR_REVOCATION**	A.QESDATA; A.CERT; A.ANCHOR
Backup & Restore	Backs up and restores Wallet data and credentials.	Supporting Functions; PID Data Model; Wallet Functions	**FSR_MIGRATION; FSR_ISSUANCE; FSR_LOG_DATA**	A.MIGR; A.PIDATT; A.LOG

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Module	Description of Modules	Functional decomposition	Requirements	Assets
Data & Credential Store	Stores and protects credentials, PID data and attributes.	PID Data Model; Supporting Functions	**FSR_PRESENTATION; FSR_ISSUANCE; FSR_PSEUDONYM; FSR_WIA; FSR_KEY_ATTEST; FSR_TRUST_ANCHOR; FSR_MIGRATION; FSR_WI_INSTALL**	A.PIDATT; A.PSEUD; A.WIA; A.KA; A.ANCHOR
Secrets Vault	Secure storage of non-key confidential assets.	WI/WSCA/WSCD Low Level Functions; PID Data Model; Supporting Functions	**FSR_MIGRATION; FSR_PSEUDONYM; FSR_WI_INSTALL**	A.MIGR; A.PSEUD; A.INIT
Transaction Log	Maintains protected logs of Wallet interactions.	Supporting Functions; Wallet Functions	**FSR_LOG_DATA; FSR_DPA_REPORT; FSR_MIGRATION; FSR_WI_INSTALL**	A.LOG
Confirmation Module	User confirmation of RP requests not requiring SCA/signature.	Presentation; Wallet Functions	**FSR_SELECT_DISCLO; FSR_PRESENTATION; FSR_PROTOCOL; FSR_LOCAL_AUTH**	A.PRES; A.PIDATT; A.PSEUD; A.NONCE
SCA	Handles online strong customer authentication flows.	Wallet Provider Interaction; WI/WSCA/WSCD Low Level Functions	**FSR_REMOTE_AUTH; FSR_LOCAL_AUTH; FSR_SEC_CHANNEL; FSR_PROTOCOL; FSR_LOG_DATA**	A.PIN; A.PUK; A.KEYAUTH; A.POSS; A.UAE; A.NONCE; A.CMD; A.MUT_KEY; A.PRES
SUA	Handles online strong User authentication flows.	Wallet Provider Interaction; PID Provider Interaction; WI/WSCA/WSCD Low Level Functions	**FSR_REMOTE_AUTH; FSR_SEC_CHANNEL; FSR_WIA; FSR_KEY_ATTEST**	A.PIN; A.KEYAUTH; A.POSS; A.UAE; A.ARD; A.NONCE; A.CMD; A.MUT_KEY; A.WIA; A.KA
Signature Initiation	Confirms signing sessions with remote e-signing services.	Remote Signing; Wallet Provider Interaction	**FSP_Q_SIGNATURE; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL; FSR_PROTOCOL**	A.QESDATA; A.UAE; A.CMD; A.NONCE; A.MUT_KEY; A.CERT
WIM	Obtains and uses Wallet Instance Attestations.	Wallet Provider Interaction; WI/WSCA/WSCD Low Level Functions	**FSR_WIA; FSR_REVOCATION; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL; FSR_TRUST_ANCHOR**	A.WIA; A.WIID; A.MON; A.CERT; A.ANCHOR; A.MUT_KEY; A.NONCE
SICZ	Obtains and uses Wallet Unit Attestations.	Wallet Provider Interaction; WI/WSCA/WSCD Low Level Functions	**FSR_WIA; FSR_KEY_ATTEST; FSR_REVOCATION; FSR_W2W; FSR_SEC_CHANNEL**	A.WIA; A.KA; A.CERT; A.ANCHOR; A.NONCE; A.MUT_KEY

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Module	Description of Modules	Functional decomposition	Requirements	Assets
WUC	Handles Wallet Unit Attestations.	Wallet Provider Interaction; PID Data Model; WI/WSCA/WSCD Low Level Functions	**FSR_WIA; FSR_KEY_ATTEST; FSR_REVOCATION; FSR_ISSUANCE**	A.WIA; A.KA; A.WIID
Offline Presentation	Presents PID/attestations via ISO 18013-5.	Presentation; Wallet Functions; PID Data Model	**FSR_MDOC; FSR_W2W; FSR_PRESENTATION; FSR_SELECT_DISCLO; FSR_PROTOCOL; FSR_TRUST_ANCHOR; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL; FSR_LOG_DATA**	A.PIDATT; A.PRES; A.CERT; A.ANCHOR; A.NONCE; A.UAE; A.CMD; A.LOG
Online Presentation S-device	OID4VP presentation on the same device.	Presentation; PID Provider Interaction; Wallet Functions	**FSR_PROTOCOL; FSR_PRESENTATION; FSR_SELECT_DISCLO; FSR_ACC_REG_CERT; FSR_PSEUDONYM; FSR_TRUST_ANCHOR; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL; FSR_LOG_DATA**	A.PIDATT; A.PRES; A.PSEUD; A.CERT; A.ANCHOR; A.NONCE; A.UAE; A.CMD; A.LOG
Online Presentation X-device	OID4VP presentation via redirect or QR code.	Presentation; PID Provider Interaction; Wallet Provider Interaction	**FSR_PROTOCOL; FSR_PRESENTATION; FSR_SELECT_DISCLO; FSR_ACC_REG_CERT; FSR_PSEUDONYM; FSR_TRUST_ANCHOR; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL; FSR_LOG_DATA**	A.PIDATT; A.PRES; A.PSEUD; A.CERT; A.ANCHOR; A.NONCE; A.UAE; A.CMD; A.LOG
Disclosure Authorization	Checks intended use and disclosure policy compliance.	Presentation; PID Data Model; Wallet Functions	**FSR_SELECT_DISCLO; FSR_ACC_REG_CERT; FSR_PRESENTATION; FSR_PSEUDONYM**	A.PIDATT; A.PRES; A.PSEUD; A.CERT; A.NONCE
Request Verification	Verifies RP request integrity and authenticity.	Presentation; PID Provider Interaction	**FSR_PROTOCOL; FSR_ACC_REG_CERT; FSR_TRUST_ANCHOR; FSR_MDOC**	A.PRES; A.CERT; A.ANCHOR; A.NONCE
Credential Generator	Generates Wallet-level attestations such as mandates.	PID Data Model; Issuance; Wallet Functions	**FSR_ISSUANCE; FSR_TRUST_ANCHOR; FSR_REMOTE_AUTH; FSR_SEC_CHANNEL**	A.PIDATT; A.CERT; A.ANCHOR; A.UAE; A.CMD; A.MUT_KEY

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Module	Description of Modules	Functional decomposition	Requirements	Assets
Resilience	Provides security mechanisms protecting the Wallet against compromise, analysis, insecure operating environments and attacks on sensitive assets.	Supporting Functions; WI/WSCA/WSCD Low Level Functions; Framework	**FSR_WI_INSTALL; RSR_USERDEVICE; RSR_PIN_ENTRY; RSR_SCRN_PRT; RSR_OPER_MOD; RSR_MON; RSR_NO_DEBG; RSR_MAC; RSR_SEC_CLR; RSR_DAT_SAN; RSR_KEY_MNG; RSR_CRPTO; RSR_RNG; RSR_SEC_COD;**	ALL ASSETS

C OWASP MAS mapping to the Security Requirements

This appendix provides a mapping between the requirements defined in this document and the corresponding requirements of the OWASP Mobile Application Security (MAS) standard. The mapping identifies the relevant OWASP MASVS requirements that provide equivalent security objectives and is intended to facilitate comparison and demonstrate coverage of the security requirements.

C.1 Functional security requirement mapping

ID Requirement	OWASP Requirements
FSR_REVOCATION	#9 - The app rotates cryptographic keys regularly.
FSR_LOCAL_AUTH	#17 - The app enforces step-up authentication before granting access to sensitive functionality. #21 - The app implements local authentication securely.
FSP_REMOTE_AUTH	#17 - The app enforces step-up authentication before granting access to sensitive functionality. #22 - The app does not allow fallback to non-biometric credentials for sensitive transactions.
FSR_SEC_CHANNEL	#27 - The app correctly implements identity pinning.
FSR_WI_INSTALL	#40 - The app terminates if an outdated app version is detected. #42 - The app terminates if an unsupported OS version is detected. #43 - The app targets a recent OS version.

C.2 Resilience security requirement mapping

ID Requirement	OWASP Requirements
RSR_MON	#1 -The app excludes sensitive data from application logs. #19 - The app ensures non-repudiation for critical actions. #76 - The app detects attacks by malware.
RSR_DAT_SAN	#1 - The app excludes sensitive data from application logs. #2 - The app excludes sensitive data from backups.

ID Requirement	OWASP Requirements
RSR_SEC_COD	#3 - The app encrypts sensitive data stored in private storage. #6 - The app does not hardcode sensitive data in the application package. #11 - The app properly encrypts sensitive data. #18 - The app makes sensitive data inaccessible after session termination. #20 - The app enforces authentication and authorization on its components. #32 - The app securely handles deep links. #34 - The app only permits trusted app extensions to interact with it. #35 - The app securely handles intents. #36 - The app does not expose sensitive native functionality to WebView content. #37 - The app does not allow WebViews loading untrusted content to access local resources. #38 - The app does not allow WebViews to load untrusted content. #41 - The app's third-party components are regularly checked for known vulnerabilities. #44 - The app loads dynamic code safely from trusted sources. #45 - The app uses the compiler-provided security features of the platform. #46 - The app does not use deprecated APIs or functionality. #47 - The app does not use non-standard libraries for security-critical functionality. #48 - The app securely handles untrusted data. #72 - The app adequately manages permissions. #75 - The app package does not contain malicious code.
RSR_KEY_MNG	#5 - The app stores cryptographic keys inside the platform-provided secure keystore. #7 - The app securely generates cryptographic keys. #10 - The app restricts access to cryptographic keys. #23 - The app invalidates keys after any enrolment of new biometric data. #26 - The app validates certificates for all network traffic.
RSR_CRPTO	#7 - The app securely generates cryptographic keys. #8 - The app derives cryptographic keys using approved key derivation functions. #12 - The app properly hashes sensitive data. #11 - The app properly encrypts sensitive data. #13 - The app properly uses Message Authentication Codes (MACs). #14 - The app properly generates cryptographic signatures. #15 - The app properly verifies cryptographic signatures.
RSR_RNG	#16 - The app properly generates random numbers.
RSR_SCRN_PRT	#24 - The app enables auto-fill support for authenticators. #28 - The app does not unnecessarily expose sensitive data through the app user interface. #29 - The app does not unnecessarily expose sensitive data through system notifications. #31 - The app protects its sensitive screens against overlay attacks. #33 - The app avoids placing sensitive data on the system clipboard. #39 - The app prevents sensitive data from being exposed to, or captured by, accessibility services. #76 - The app detects attacks by malware.
RSR_PYL_ENC	#25 - The app encrypts all network traffic. #55 - The app applies application-level payload encryption in addition to transport-layer encryption.
RSR_FOCS	#30 - The app removes sensitive data from views when moved to the background or when being recorded.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

ID Requirement	OWASP Requirements
RSR_PIN_ENTRY	#31 - The app protects its sensitive screens against overlay attacks. #39 - The app prevents sensitive data from being exposed to, or captured by, accessibility services. #76 - The app detects attacks by malware.
RSR_NO_DEBG	#50 - The app disables debug mechanisms. #54 - The app removes debug artifacts. #60 - The app terminates if a debugger is detected.
RSR_OBF_COD	#51 - The app applies code obfuscation to hinder reverse engineering. #52 - The app applies resource obfuscation to hinder reverse engineering
RSR_DYN_TL	#53 - The app implements anti-deobfuscation techniques to protect its obfuscation. #61 - The app detects dynamic analysis tools. #76 - The app detects attacks by malware.
RSR_OPER_MOD	#56 - The app terminates if a rooted/jailbroken device is detected. #57 - The app detects when it is running inside an app-virtualization environment. #58 - The app detects when it is running on an emulator.
RSR_ATT	#59 - The app implements device attestation.
RSR_INTG	#62 - The app verifies its own integrity. #63 - The app verifies the integrity of its resources. #64 - The app verifies the integrity of its code. #76 - The app detects attacks by malware.
Privacy	#66 - The app does not use persistent or unique identifiers in a way that enables user tracking. #67 - The app provides an adequate privacy policy. #68 - The app adequately declares all user collected data. #69 - The app provides adequate controls to manage user data. #70 - The app provides adequate controls over the visibility of user data. #71 - The app requests user consent prior to any data processing. #73 - The app adequately declares all tracking domains it connects to. #74 - The app uses privacy-preserving functionality.

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme

Colofon

FITCEM Protection Profile for Wallet Instance in the remote HSM architecture

EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL
Ensuring High Assurance of EUDI Wallets & the NL National eID Scheme