EUDIW CERTIFICATION SYSTEM MANAGEMENT BOARD NL

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

NL-WSCA-conditions

NL EUDI Wallet Certification System

Conditions for the Underlying Platform Supporting AVA_VAN.5 Evaluation

RC1.1

2026.10.29

Document reference	NL-WSCA-condtions
Version	RC1.1
Status	Release Candidate
Date	29/10/2026
Document type	Conditions in support of a WSCA for the Underlying Platform Supporting AVA_VAN.5 Evaluation
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-WSCA NL-WI (-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

Abstract

The **Wallet Secure Cryptographic Application (WSCA)** is the security-critical software component of the EUDI Wallet responsible for protecting and performing cryptographic operations and managing security-sensitive assets. As a core security component, a compromise of the WSCA could directly affect the security and trustworthiness of the EUDI Wallet. Under Commission Implementing Regulation (EU) 2024/2981, the WSCA is therefore subject to a **high level of vulnerability analysis, with AVA_VAN.5 defined as the default assessment level**.

AVA_VAN.5 provides a rigorous analysis of potential vulnerabilities and resistance against attackers with high attack potential, providing assurance that the WSCA cannot be practically compromised through known or previously unknown vulnerabilities within the defined evaluation scope.

As the WSCA is implemented as a **software-only product**, its security also depends on the security properties of the underlying operating platform on which it executes. Weaknesses or vulnerabilities in the platform could undermine the security objectives of the WSCA, even where the WSCA itself is securely implemented.

Therefore, the security requirements for the underlying operating platform must be clearly defined and justified to ensure that the platform provides the necessary security properties to support the **AVA_VAN.5 vulnerability assessment level** claimed for the WSCA. This document defines these requirements and establishes the security conditions that the underlying platform must satisfy so that vulnerabilities or weaknesses in the platform do not undermine the security objectives or the overall assurance level of the WSCA.

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

Table of Contents

1 Introduction 6
1.1 Security problem definition 6
1.2 Document Identification 6
1.3 Structure of the document 6
2 Security requirements Operational Environment 8
3 High-Level layered Design of the Underlying Platform 10
3.1 Hardware / Host System 10
3.2 Boot and Initialisation Layer 10
3.3 Operating System Kernel 10
3.4 Operating System / System Services 11
3.5 Application Space 11
3.6 Container / Isolation Layer 11
3.7 WSCA Execution Environment 11
4 Security Principles for the Underlying Platform 12
5 Requirements for the Underlying Platform 14
5.1 Hardware Requirements 14
5.2 Boot and Initialisation Layer Requirements 15
5.3 Operating System Kernel Requirements 16
5.4 Operating System / System Services Requirements 17
5.5 Application Space Requirements 18
5.6 Container / Isolation Layer Requirements 19
5.7 WSCA Execution Environment Requirements 20
6 Application of the Requirements for the Linux-Based Underlying Platform 22
6.1 Linux and Hardware Security Mechanisms Mapping 22
6.2 Hardware / Host System for Linux 24
6.3 Boot and Initialisation for Linux 26
6.4 Operating System Kernel for Linux 26
6.5 Operating System / System Services for Linux 28
6.6 Application Space for Linux 28
6.7 Container / Isolation Layer for Linux 29
6.8 WSCA Execution Environment for Linux 30

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

1 Introduction

1.1 Security problem definition

The WSCA is a **software-only application** running on an underlying operating system. As the WSCA is evaluated against **AVA_VAN.5**, the security of the underlying platform must provide a level of protection commensurate with the security objectives and vulnerability resistance expected for the WSCA. A weakness in the underlying platform could provide an attacker with a means to bypass the security mechanisms of the WSCA and consequently undermine the AVA_VAN.5 assurance claim.

The objective of this document is to **define the security requirements and protection principles for the underlying platform on which the WSCA operates**. These requirements address the platform capabilities and configuration necessary to protect the WSCA against unauthorised access, modification, interference and exploitation, and to ensure that platform vulnerabilities do not provide an attack path that would undermine the WSCA security objectives.

The WSCA and its underlying platform must be deployed in a **secure and controlled operational environment**. The operational environment shall provide the physical, administrative and operational controls necessary to protect the WSCA and its underlying platform against unauthorised physical access, tampering, misuse and other environmental threats. These controls include as applicable, **physical protection of the servers and WSCD, controlled access to the environment, and protection of supporting infrastructure and network connections**. The operational environment therefore forms an additional layer of protection and shall be sufficiently secure to ensure that the security objectives of the WSCA and the protection provided by the underlying platform are not undermined during operation.

The requirements defined in this document establish the **minimum-security conditions for the underlying platform and operational environment to support the WSCA's AVA_VAN.5 evaluation**. They cover, as applicable, aspects such as minimal configuration, restricted execution, least privilege, process isolation, memory protection, controlled interfaces, platform integrity and vulnerability management.

1.2 Document Identification

Title: Security Requirements for the Underlying Platform of the WSCA Supporting AVA_VAN.5 Evaluation

Version: 0.7

Date: Tuesday, 27 October 2026

Authors: SGS Brightsight (ITSEF and CB)

1.3 Structure of the document

This document uses a **step-by-step, layered approach** to identify the security measures needed to protect the WSCA and support its **AVA_VAN.5 evaluation**. It starts by considering the operational environment in which the WSCA is deployed, then identifies the

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

components of the underlying platform, establishes the security principles that apply to these components, and finally translates these principles into detailed security requirements.

Section 2 — Security requirements Operational Environment: describes the conditions and security requirements that shall be in place around the WSCA and its underlying platform.

Section 3 —High-Level layered Design of the Underlying Platform: High-Level layered Design of the Underlying Platform: describes the different components that make up the underlying platform, including the hardware, bootloader, operating system kernel, system services, application space and container environment. This establishes the scope of the platform that supports the WSCA.

Section 4 — Security Principles for the Underlying Platform: defines the fundamental security principles that shall be applied to the underlying platform. These principles describe how the platform is to be protected, including minimising the attack surface, restricting execution and privileges, isolating processes, protecting memory and controlling interfaces.

Section 5 — Requirements for the Underlying Platform: translates the security principles into specific requirements for each applicable platform layer. These requirements define the security properties and configurations necessary to ensure that individual weaknesses in the underlying platform do not provide an attack path that could compromise the WSCA or undermine its **AVA_VAN.5 assurance claim**.

Section 6 — Application of the Requirements for the Linux-Based Underlying Platform: This chapter describes how the defined security requirements can be implemented in a **Linux-based environment**.

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

2 Security requirements Operational Environment

This section describes the **required operational environment** in which the WSCA shall be deployed and identifies the architectural, platform and deployment constraints that are required to support the security objectives and assurance expectations associated with an AVA_VAN.5 evaluation.

Section 5 defines the detailed security **requirements for the dedicated underlying platform**. These requirements are derived from the security principles defined in Section 4 and specify how these principles apply to the individual layers and components of the platform. The operational environment is depicted in ***Figure 1***.

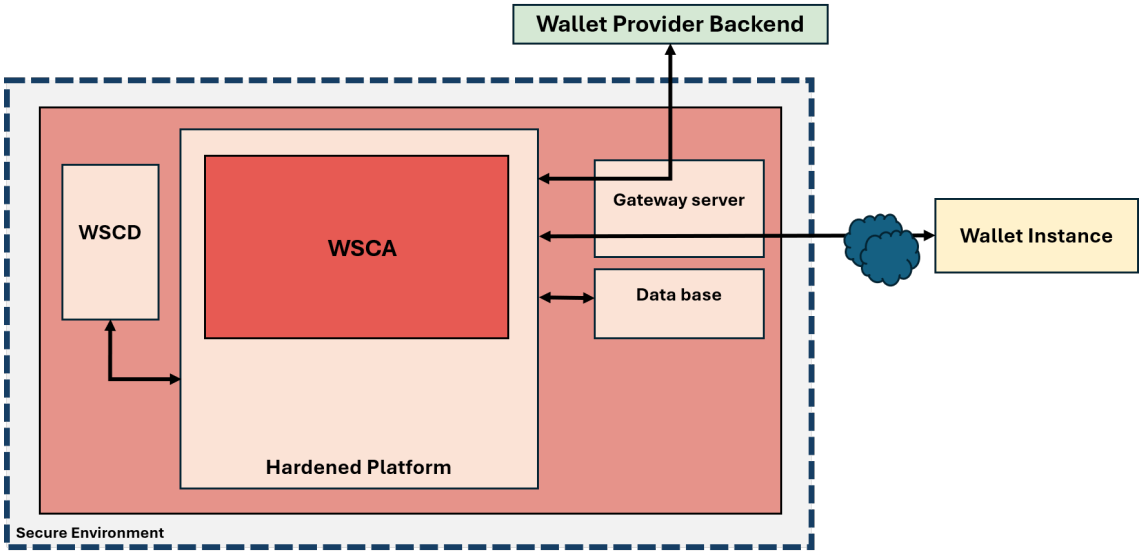


Figure 1 — WSCA in Its Environment

The security requirements for the operational environment in ***Table 1***, follow the principle of minimising exposure and **reducing the attack surface of the WSCA and its dedicated underlying platform** and thus reduce the likelihood that a vulnerability in the operational environment or the underlying platform can be used to compromise the WSCA.

Table 1 — Security requirements Operational Environment

ID	Security requirements Operational Environment
ENV-DEDICATED_PLATFROM	The WSCA operates on a **dedicated underlying hardened platform and hardware** that forms part of its trusted operational environment. This platform is responsible for providing the security mechanisms necessary to protect the WSCA against unauthorised access, modification, interference and exploitation (see detailed requirements in section 5). **No applications, services, workloads or processing functions unrelated to the WSCA shall be installed or operated** on the hardened platforms and on top **dedicated hardware**. Consequently, deployment of the WSCA as a shared cloud application is not permitted.

ID	Security requirements Operational Environment
ENV-TRUSTED-ADMIN	Administrators are appropriately trained, follow established policies and guidance, and are trusted to act without malicious intent. They ensure that passwords and credentials have sufficient strength and entropy. The platform enforces **dual control for administrative tasks** to prevent a single administrator from performing sensitive or potentially malicious actions alone.
ENV-GATEWAY	The WSCA platform shall not be directly exposed to the Internet. External communication is mediated through a **gateway platform positioned between the WSCA platform and the Internet**. This provides an additional security boundary for network communication and reduces the direct exposure of the WSCA execution environment. The gateway implements at least DDoS protection, intrusion detection, antivirus functionality and firewall functionality.
ENV-DATABASE	The **WSCD** and **security-relevant databases supporting the WSCA** shall be located within the same protected operational environment as the WSCA platform and shall not be directly accessible from untrusted networks.
ENV-WSCD-CHANNEL	**Communication channels** between the WSCA and the WSCD and the WSCA the gateway platform shall be **authenticated and encrypted.** The communication mechanism shall protect the confidentiality and integrity of all exchanged security-relevant data and cryptographic commands, and shall prevent unauthorised interception, modification, replay, or impersonation of communicating parties. The communication channel between the WSCA and the WSCD shall use a **dedicated physical connection**, isolated from all other computer systems.
ENV-ENVIRONMENT	The TOE and its underlying platform shall be located in a physically secured environment that protects against unauthorised physical access, tampering, theft, and physical interference. Access shall be restricted to authorised personnel through appropriate physical access controls. Physical security measures shall protect the **WSCA**, **WSCD** the **databases** and the **gateway** against attacks that could compromise their integrity, confidentiality, or availability, including attacks within the scope of the AVA_VAN.5 vulnerability assessment.

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

3 High-Level layered Design of the Underlying Platform

The underlying platform is the complete software and hardware environment required to execute and protect the software-only WSCA. The platform consists of several layers, where each layer provides functionality and security mechanisms required by the layers above it.

For each of these layers, Section 5 defines a set of security requirements derived from the security principles described in Section 4. The High-Level layered Design is depicted in ***Figure 2***.

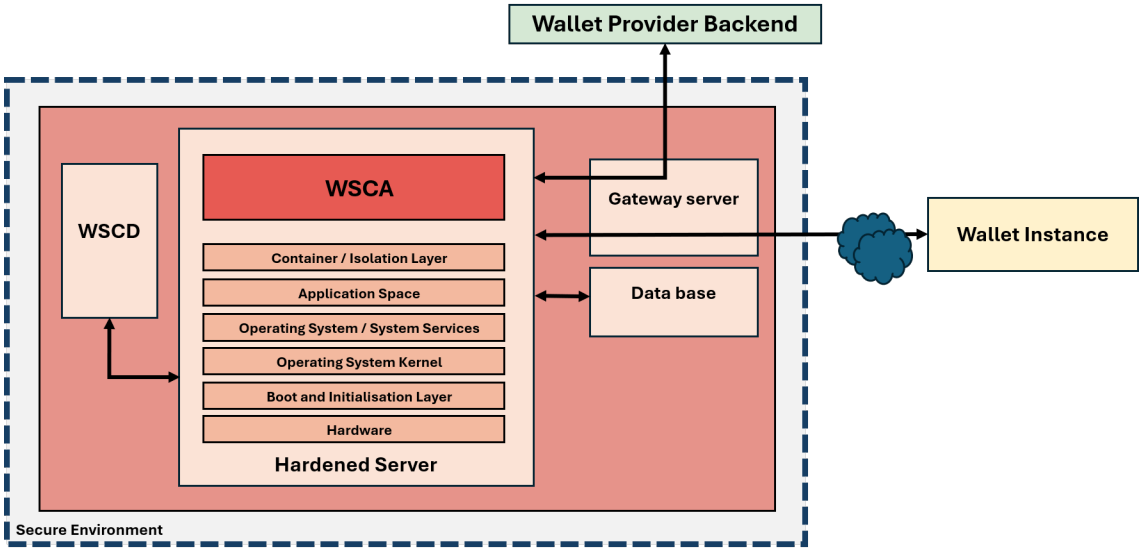


Figure 2 — High-Level Design of the Underlying Platform

3.1 Hardware / Host System

The hardware/host system provides the physical resources on which the underlying platform operates. This includes the processor, memory, storage, network interfaces and the physical interfaces used to connect to the WSCD. Where available, hardware-based security features such as hardware-assisted memory protection or trusted execution mechanisms may also contribute to the protection of the WSCA.

3.2 Boot and Initialisation Layer

The boot and initialisation layer is responsible for bringing the platform into a known and controlled state before the operating system starts. It typically includes firmware such as UEFI and a bootloader, and may include mechanisms such as Secure Boot and integrity verification. This layer is important because compromise of the boot process could allow malicious code to execute before the operating system and security controls are established.

3.3 Operating System Kernel

The operating system kernel provides the fundamental security and resource-management functions of the platform. It controls processes, memory, privileges, devices, system calls,

inter-process communication and network functionality. The kernel is particularly security-critical because a compromise of the kernel could allow an attacker to bypass the security controls protecting the WSCA.

3.4 Operating System / System Services

Operating system services provide the functionality required by the kernel and applications to operate the platform. These may include authentication and authorisation, file and storage services, cryptographic services, logging, time services, networking and system management. Only services required for the secure operation and administration of the WSCA environment should be enabled.

3.5 Application Space

Application space contains the software that runs above the operating system, excluding the WSCA itself. This may include supporting applications, utilities, management components and other software required to operate the platform. The application space should be minimised and controlled to reduce the possibility that vulnerabilities in unrelated applications can be used to compromise the WSCA.

3.6 Container / Isolation Layer

Where containers are used, the container runtime and its associated isolation mechanisms form part of the platform supporting the WSCA. Containers provide separation between applications and can restrict access to platform resources through mechanisms such as namespaces, resource controls and security policies. The security of the container runtime, container configuration and container images must therefore be considered where the WSCA executes within a container.

3.7 WSCA Execution Environment

The WSCA execution environment is the specific environment in which the WSCA software executes. It includes the runtime components, libraries, configuration, security policies and interfaces required by the WSCA. It also includes the interfaces through which the WSCA communicates with the WSCD and, where applicable, the gateway. The execution environment must provide the necessary isolation and protection to prevent other platform components from compromising the WSCA.

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

4 Security Principles for the Underlying Platform

Table 2 defines the security principles for the underlying platform. The principles are focused on **reducing the attack surface, restricting execution and privileges, isolating the WSCA from other platform components, protecting its memory and security-relevant data, controlling interfaces and network exposure, and maintaining the integrity of the platform**.

Table 2 — Security Principles for the Underlying Platform

ID	Security Principle	Security Principal description
01	**Minimal Configuration**	The underlying platform shall be configured with only those operating system components, services, drivers, interfaces and functions that are required for the secure operation of the WSCA. Unnecessary functionality shall be disabled or removed.
02	**Limited Execution**	The underlying platform shall restrict execution to authorised software, processes and services required for the operation of the WSCA. Execution of unauthorised software shall be prevented.
03	**Least Privilege**	The underlying platform shall ensure that the WSCA and its supporting processes operate with only the privileges necessary to perform their specified functions.
04	**Process Isolation**	The underlying platform shall provide isolation between the WSCA and other processes such that an unauthorised process cannot access, modify or interfere with the WSCA or its security-relevant data.
05	**Memory Protection**	The underlying platform shall protect memory allocated to the WSCA against unauthorised reading, modification and execution by other processes or users.
06	**Controlled Interfaces**	The underlying platform shall restrict access to system interfaces, devices, files, inter-process communication mechanisms and other resources to those required for the operation of the WSCA.
07	**Platform Integrity**	The underlying platform shall provide mechanisms to prevent or detect unauthorised modification of platform components that could affect the security of the WSCA.
08	**Network Protection**	The underlying platform shall expose only the network services, protocols and ports required for the operation of the WSCA. Direct network access from the WSCA to untrusted networks shall be prevented where it is not required for the WSCA's specified operation.
09	**WSCA Data Protection**	The underlying platform shall protect WSCA security-relevant data, including configuration data, credentials and other sensitive information, against unauthorised disclosure, modification or deletion.
10	**Privilege Escalation Protection**	The underlying platform shall prevent a process or user operating with limited privileges from obtaining privileges that could be used to compromise or interfere with the WSCA.
11	**Security-Relevant Logging**	The underlying platform shall provide logging of security-relevant events necessary to support the detection and investigation of events that could affect the security of the WSCA.

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

ID	Security Principle	Security Principal description
12	**Vulnerability Management**	The underlying platform shall be maintained using a defined vulnerability and security-update management process to ensure that known vulnerabilities that could affect the WSCA are assessed and appropriately remediated.
13	**Cyber Security Monitoring**	The underlying platform shall be continuously monitored for security-relevant events, anomalous behaviour, and indicators of compromise that could affect the security of the WSCA. Detected events shall be analysed and, where necessary, trigger appropriate alerts and incident response measures.

5 Requirements for the Underlying Platform

This section defines the **security requirements for each layer** of the underlying platform architecture. The requirements are derived from the **security principles** established in Section 4 and translate these principles into concrete security objectives and controls for the individual platform layers.

Together, these requirements ensure that the underlying platform provides the security properties necessary to protect the WSCA and to support its AVA_VAN.5 assurance objectives.

5.1 Hardware Requirements

The hardware/host system provides the physical foundation on which the underlying platform and WSCA operate. As the WSCA is a software-only application, compromise or manipulation of the hardware may bypass software-based security mechanisms and directly affect the confidentiality, integrity and availability of the WSCA. The hardware shall therefore be configured and operated to provide a controlled and trustworthy foundation for the platform.

ID	Security Principle	Hardware Requirement
HW-01	**Minimal Configuration**	The host system shall be configured with only the hardware components required for the operation of the WSCA platform. Unnecessary hardware devices and interfaces shall be disabled or removed where technically possible.
HW-02	**Limited Execution**	The hardware shall support the platform's mechanisms for restricting execution to authorised platform and WSCA software. Hardware mechanisms that could allow unauthorised code execution or bypass platform controls shall be disabled or appropriately controlled.
HW-03	**Least Privilege**	Hardware resources and privileged hardware functions shall only be accessible through authorised platform mechanisms and to components requiring such access for their specified functions.
HW-04	**Process Isolation**	The host hardware shall provide, where supported, hardware-assisted mechanisms for isolating processes and execution environments, including processor privilege levels and virtual memory protection.
HW-05	**Memory Protection**	The processor and memory subsystem shall support hardware-assisted memory protection mechanisms. These mechanisms shall be enabled and configured to prevent unauthorised access to memory allocated to the WSCA.
HW-06	**Controlled Interfaces**	Physical interfaces and peripheral devices shall be limited to those required for the operation and administration of the WSCA platform. Unnecessary interfaces, such as unused USB, storage or peripheral interfaces, shall be disabled or physically restricted where possible.
HW-07	**Platform Integrity**	Where supported by the hardware, mechanisms such as a hardware root of trust, Trusted Platform Module (TPM), measured boot or Secure Boot shall be used to support verification of the integrity of the platform during system initialisation.

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

ID	Security Principle	Hardware Requirement
HW-08	**Network Protection**	Network interfaces shall be limited to those required for WSCA operation and platform administration. Unused network interfaces shall be disabled, and the hardware configuration shall not provide an uncontrolled path to untrusted networks.
HW-09	**WSCA Data Protection**	Hardware resources used to store or process WSCA security-relevant data shall be protected against unauthorised physical and logical access. Where hardware security mechanisms are available, they shall be used where required to protect sensitive data.
HW-10	**Privilege Escalation Protection**	The hardware shall support processor and platform security mechanisms that prevent lower-privileged software from directly accessing privileged hardware resources or bypassing the access-control mechanisms of the underlying platform.
HW-11	**Security-Relevant Logging**	Where hardware security events are relevant to the security of the WSCA, the platform shall provide mechanisms to capture or expose such events to the platform logging and monitoring functions.
HW-12	**Vulnerability Management**	The hardware platform, including firmware and microcode, shall be subject to a defined vulnerability-management process. Security updates addressing vulnerabilities that could affect the WSCA shall be assessed and applied in accordance with the applicable security-update policy.
HW-13	**Cyber Security Monitoring**	The hardware platform shall support the collection and reporting of security-relevant hardware events, where technically available, including hardware faults, integrity violations, and indicators of unauthorised physical or logical access. Such events shall be made available to the platform's security monitoring mechanisms for analysis and, where necessary, trigger appropriate alerts.

5.2 Boot and Initialisation Layer Requirements

The boot and initialisation layer establishes the initial state of the underlying platform before the operating system and WSCA are executed. As this layer executes before the security controls of the operating system are fully established, a compromise of the boot process could allow an attacker to introduce malicious or modified components and subsequently bypass the protection mechanisms of the platform and WSCA.

ID	Security Principle	Boot and Initialisation Requirement
BOOT-01	**Minimal Configuration**	The boot and initialisation environment shall be configured with only the firmware, boot components and configuration required to initialise the platform and start the authorised operating system. Unnecessary boot functionality shall be disabled where technically possible.
BOOT-02	**Limited Execution**	The boot process shall permit execution only of authorised firmware, bootloader components and operating system components. Unauthorised code shall not be permitted to execute during platform initialisation.
BOOT-03	**Least Privilege**	Boot and initialisation components shall operate with only the privileges and access to hardware resources necessary to perform platform initialisation and start the authorised operating system.
BOOT-04	**Process Isolation**	The boot environment shall prevent unauthorised or untrusted components from gaining access to or interfering with security-relevant boot processes and initialisation components.

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

ID	Security Principle	Boot and Initialisation Requirement
BOOT-05	**Memory Protection**	The boot environment shall use available hardware and firmware mechanisms to protect memory used during initialisation against unauthorised access or modification.
BOOT-06	**Controlled Interfaces**	Access to firmware configuration, boot interfaces and hardware configuration functions shall be restricted to authorised personnel or authorised management mechanisms. Unnecessary boot interfaces shall be disabled or otherwise restricted.
BOOT-07	**Platform Integrity**	The boot process shall provide mechanisms to verify the authenticity and/or integrity of firmware, bootloader and operating system components before they are executed. Where supported, Secure Boot, measured boot or equivalent integrity mechanisms shall be enabled and appropriately configured.
BOOT-08	**Network Protection**	Network functionality during the boot and initialisation process shall be disabled unless required for a defined platform function. Where network boot or other pre-OS network functionality is required, it shall be restricted to authorised sources and protected against unauthorised modification or execution.
BOOT-09	**WSCA Data Protection**	The boot and initialisation process shall not expose or provide unauthorised access to WSCA security-relevant data stored on the platform. Access to storage containing WSCA data during boot shall be restricted to authorised platform components.
BOOT-10	**Privilege Escalation Protection**	The boot and initialisation mechanisms shall prevent lower-trust or unauthorised components from obtaining privileged execution through manipulation of the boot process, boot configuration or firmware interfaces.
BOOT-11	**Security-Relevant Logging**	Where supported, security-relevant boot and integrity events, including failed integrity checks and unauthorised boot attempts, shall be recorded and made available to the platform's security monitoring mechanisms.
BOOT-12	**Vulnerability Management**	Firmware, bootloader and other boot components shall be subject to a defined vulnerability-management process. Security vulnerabilities affecting the integrity or security of the WSCA platform shall be assessed and appropriately remediated.
BOOT-13	**Cyber Security Monitoring**	The boot and initialisation process shall support the detection of security-relevant anomalies, including unexpected boot configurations, failed integrity verification, and unauthorised changes to boot components. Detected events shall be reported to the platform's security monitoring mechanisms to enable timely investigation and response.

5.3 Operating System Kernel Requirements

The operating system kernel provides the fundamental security and resource-management functions on which the WSCA relies. It controls processes, memory, privileges, devices, system calls and inter-process communication. A compromise of the kernel could allow an attacker to bypass the security mechanisms protecting the WSCA. The kernel shall therefore be configured and operated to provide strong isolation, access control and protection of the WSCA execution environment.

ID	Security Principle	Operating System Kernel Requirement
KERN-01	**Minimal Configuration**	The kernel shall be configured with only the functionality, drivers, modules and interfaces required for the secure operation of the WSCA platform. Unnecessary kernel functionality shall be disabled or removed where technically possible.

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

ID	Security Principle	Operating System Kernel Requirement
KERN-02	**Limited Execution**	The kernel shall prevent unauthorised kernel modules, drivers or other privileged code from being loaded or executed. Only authorised and trusted kernel components shall be permitted.
KERN-03	**Least Privilege**	The kernel shall provide mechanisms to enforce separation between privileged and non-privileged execution and shall ensure that the WSCA operates with only the privileges required for its specified functions.
KERN-04	**Process Isolation**	The kernel shall provide process isolation mechanisms that prevent one process from accessing, modifying or interfering with another process unless explicitly authorised.
KERN-05	**Memory Protection**	The kernel shall enforce memory isolation and shall prevent unauthorised processes from reading, modifying or executing memory allocated to the WSCA. Hardware-assisted memory protection mechanisms shall be enabled where available.
KERN-06	**Controlled Interfaces**	Access to kernel interfaces, system calls, devices, IPC mechanisms and other kernel-managed resources shall be restricted according to the privileges and functions of the requesting process.
KERN-07	**Platform Integrity**	The kernel shall provide mechanisms to protect the integrity of its own code, security configuration and security-relevant data during operation.
KERN-08	**Network Protection**	The kernel network stack shall be configured to expose only the network functionality required by the WSCA environment. Unnecessary network protocols, interfaces and services shall be disabled.
KERN-09	**WSCA Data Protection**	The kernel shall enforce access controls preventing unauthorised processes or users from accessing WSCA security-relevant files, memory, devices and other resources.
KERN-10	**Privilege Escalation Protection**	The kernel shall provide mechanisms to prevent processes operating with limited privileges from obtaining unauthorised higher privileges or bypassing kernel access-control mechanisms.
KERN-11	**Security-Relevant Logging**	The kernel shall provide logging or security event mechanisms for security-relevant events that could indicate an attempt to bypass kernel security controls or compromise the WSCA.
KERN-12	**Vulnerability Management**	The kernel and its security-relevant modules and drivers shall be subject to vulnerability management. Known vulnerabilities that could affect the WSCA shall be assessed and appropriately remediated.
KERN-13	**Cyber Security Monitoring**	The operating system kernel shall support the detection and reporting of security-relevant anomalies, including unauthorised privilege escalation, unexpected kernel module loading, violations of access controls, and attempts to exploit kernel vulnerabilities. Detected events shall be made available to the platform's security monitoring mechanisms for analysis and incident response.

5.4 Operating System / System Services Requirements

Operating system services provide functionality used by the WSCA and other components of the platform. Because vulnerabilities in unnecessary or improperly configured services can provide attack paths to the WSCA, only required services shall be enabled and their access shall be appropriately restricted.

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

ID	Security Principle	Operating System / System Services Requirement
OS-01	**Minimal Configuration**	Only operating system services required for the secure operation, administration and maintenance of the WSCA shall be installed and enabled.
OS-02	**Limited Execution**	System services shall execute only authorised binaries and components. Unauthorised modification or replacement of service executables shall be prevented or detected.
OS-03	**Least Privilege**	Each system service required by the WSCA shall operate using the minimum privileges and system resources necessary for its function.
OS-04	**Process Isolation**	System services shall be isolated from the WSCA and from each other to the extent necessary to prevent compromise of one service from directly compromising the WSCA.
OS-05	**Memory Protection**	System services shall operate using the platform's available memory protection mechanisms so that they cannot access or modify WSCA memory without authorisation.
OS-06	**Controlled Interfaces**	Access to system services shall be restricted to authorised users, processes and network endpoints. Unused service interfaces shall be disabled.
OS-07	**Platform Integrity**	System service configuration and security policies shall be protected against unauthorised modification.
OS-08	**Network Protection**	Network-facing services shall be limited to those required by the WSCA environment. Services that are not required shall be disabled and their associated ports shall not be exposed.
OS-09	**WSCA Data Protection**	System services shall not provide unauthorised access to WSCA files, configuration, credentials or other security-relevant data.
OS-10	**Privilege Escalation Protection**	System services shall be configured to prevent their interfaces or vulnerabilities from being used to obtain privileges that could compromise the WSCA.
OS-11	**Security-Relevant Logging**	Security-relevant events generated by system services shall be logged where required to detect or investigate activity that could affect WSCA security.
OS-12	**Vulnerability Management**	System services and associated components shall be regularly assessed for vulnerabilities and security updates shall be applied where required.
OS-13	**Cyber Security Monitoring**	The operating system shall continuously monitor system services for security-relevant anomalies, including unexpected service behaviour, unauthorised configuration changes, suspicious process activity, and indicators of compromise. Detected events shall be reported to the platform's security monitoring mechanisms for analysis, alerting, and incident response.

5.5 Application Space Requirements

The application space contains software running above the operating system, excluding the WSCA itself. Unnecessary applications and utilities increase the attack surface and may provide an attacker with an indirect path towards the WSCA. The application space shall therefore be strictly controlled.

ID	Security Principle	Application Space Requirement
APP-01	**Minimal Configuration**	Only applications and utilities required for the operation, administration and maintenance of the WSCA environment shall be installed.

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

ID	Security Principle	Application Space Requirement
APP-02	**Limited Execution**	Execution of applications, scripts and utilities shall be restricted to authorised software. Application allow-listing or equivalent controls shall be applied where technically feasible.
APP-03	**Least Privilege**	Applications and utilities shall operate with the minimum privileges required for their intended functions.
APP-04	**Process Isolation**	Applications shall be isolated from the WSCA so that compromise of an application does not provide direct access to WSCA processes or resources.
APP-05	**Memory Protection**	Applications shall not be able to read, modify or execute WSCA memory unless explicitly authorised and required for operation.
APP-06	**Controlled Interfaces**	Applications shall have access only to the system interfaces, files, devices and services required for their intended functions.
APP-07	**Platform Integrity**	Application binaries and security-relevant configuration shall be protected against unauthorised modification.
APP-08	**Network Protection**	Applications shall not establish network connections to untrusted networks unless explicitly required and authorised.
APP-09	**WSCA Data Protection**	Applications shall be prevented from accessing WSCA security-relevant data unless explicitly authorised.
APP-10	**Privilege Escalation Protection**	Applications shall be configured and controlled to prevent exploitation of application vulnerabilities from resulting in unauthorised privilege escalation affecting the WSCA.
APP-11	**Security-Relevant Logging**	Security-relevant application events that could indicate an attempt to compromise the WSCA environment shall be logged where appropriate.
APP-12	**Vulnerability Management**	Applications installed on the platform shall be subject to vulnerability management. Unnecessary or unsupported applications shall be removed.
APP-13	**Cyber Security Monitoring**	Applications in the WSCA environment shall be monitored for security-relevant anomalies, including unauthorised execution, unexpected application behaviour, suspicious process activity, and indicators of compromise. Detected events shall be reported to the platform's security monitoring mechanisms for analysis, alerting, and incident response.

5.6 Container / Isolation Layer Requirements

Where the WSCA is deployed within a container, the container runtime and associated isolation mechanisms form part of the underlying platform. The container layer shall prevent the WSCA from being exposed to unnecessary host resources or to other containers and shall ensure that a compromise of a container does not provide an uncontrolled path to the host or WSCA environment.

ID	Security Principle	Container / Isolation Layer Requirement
CONT-01	**Minimal Configuration**	Container images shall contain only the software, libraries and configuration required for the intended WSCA function.
CONT-02	**Limited Execution**	Only authorised container images and workloads shall be permitted to execute. Container images shall originate from controlled and trusted sources.
CONT-03	**Least Privilege**	Containers shall execute with the minimum privileges required. Privileged containers shall not be used unless explicitly justified and authorised.

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

ID	Security Principle	Container / Isolation Layer Requirement
CONT-04	**Process Isolation**	Containers shall be isolated from the host, other containers and unrelated processes using the platform's available isolation mechanisms.
CONT-05	**Memory Protection**	Container resource controls shall prevent one container from accessing or exhausting memory resources in a manner that could compromise the WSCA or the underlying platform.
CONT-06	**Controlled Interfaces**	Container access to host devices, filesystems, kernel interfaces and other host resources shall be explicitly restricted. Unnecessary host mounts and device access shall not be permitted.
CONT-07	**Platform Integrity**	Container images, runtime components and container configuration shall be integrity-protected and changes shall be controlled.
CONT-08	**Network Protection**	Container network connectivity shall be restricted to the connections required for WSCA operation. Unnecessary external network access shall be prevented.
CONT-09	**WSCA Data Protection**	Containers shall not have access to WSCA security-relevant data unless such access is explicitly required for the WSCA function.
CONT-10	**Privilege Escalation Protection**	Container configuration and runtime controls shall prevent container processes from escaping their isolation boundary or obtaining unauthorised host privileges.
CONT-11	**Security-Relevant Logging**	Container runtime and security events relevant to the protection of the WSCA shall be logged and made available for monitoring where appropriate.
CONT-12	**Vulnerability Management**	Container images, runtimes and supporting components shall be regularly assessed for vulnerabilities. Vulnerable or unsupported images and components shall be updated or removed.
CONT-13	**Cyber Security Monitoring**	The container runtime and isolation mechanisms shall be monitored for security-relevant anomalies, including unauthorised container creation or modification, attempted isolation-boundary violations, unexpected resource access, and suspicious workload behaviour. Detected events shall be reported to the platform's security monitoring mechanisms for analysis, alerting, and incident response.

5.7 WSCA Execution Environment Requirements

The WSCA execution environment is the final platform layer immediately supporting the WSCA software. This layer includes the runtime components, libraries, configuration and security policies on which the WSCA directly depends. Particular care shall be taken to ensure that the WSCA has access only to the resources required for its security functions.

ID	Security Principle	WSCA Execution Environment Requirement
WSCA-01	**Minimal Configuration**	The WSCA execution environment shall contain only the runtime components, libraries, configuration and supporting functionality required by the WSCA.
WSCA-02	**Limited Execution**	Only authorised WSCA binaries, libraries and supporting components shall be permitted to execute within the WSCA execution environment.
WSCA-03	**Least Privilege**	The WSCA shall execute with the minimum operating system and platform privileges required to perform its specified security functions.

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

ID	Security Principle	WSCA Execution Environment Requirement
WSCA-04	**Process Isolation**	The WSCA shall be isolated from unrelated applications, services and processes such that compromise of another component does not provide direct access to the WSCA.
WSCA-05	**Memory Protection**	The WSCA execution environment shall enforce memory protection mechanisms preventing unauthorised access to WSCA memory and security-relevant runtime data.
WSCA-06	**Controlled Interfaces**	The WSCA shall have access only to the platform interfaces, devices, files and services required for its specified operation.
WSCA-07	**Platform Integrity**	WSCA binaries, libraries and security-relevant configuration shall be protected against unauthorised modification.
WSCA-08	**Network Protection**	Network communication from the WSCA shall be restricted to explicitly authorised endpoints and protocols required for its operation.
WSCA-09	**WSCA Data Protection**	Security-relevant WSCA data shall be protected against unauthorised disclosure, modification and deletion by the underlying platform and other processes.
WSCA-10	**Privilege Escalation Protection**	The execution environment shall prevent the WSCA or its supporting components from obtaining privileges beyond those required for their specified functions.
WSCA-11	**Security-Relevant Logging**	Security-relevant events affecting the WSCA execution environment shall be logged where necessary to support detection and investigation of security events.
WSCA-12	**Vulnerability Management**	All runtime components, libraries and supporting software on which the WSCA depends shall be subject to vulnerability management and security-update processes.
WSCA-13	**Cyber Security Monitoring**	The WSCA execution environment shall be continuously monitored for security-relevant anomalies, including unauthorised changes to runtime components or configuration, unexpected WSCA behaviour, suspicious resource access, and indicators of compromise. Detected events shall be reported to the platform's security monitoring mechanisms for analysis, alerting, and incident response.

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

6 Application of the Requirements for the Linux-Based Underlying Platform

This chapter describes how the defined security requirements can be implemented in a **Linux-based environment**. Section 6.1 identifies the relevant Linux security mechanisms that may be used to fulfil the requirements. The subsequent sections describe, for each requirement, **how the applicable mechanisms can be applied and configured** to provide the required security properties.

Not all mechanisms must be used. The developer shall select an appropriate combination of mechanisms based on the platform architecture. The selected mechanisms shall be **properly justified as providing the required security properties**; equivalent mechanisms may be used where appropriate.

6.1 Linux and Hardware Security Mechanisms Mapping

Table 3 identifies the **Linux and hardware security mechanisms** that can be used to implement the security principles and requirements defined for the underlying platform. It provides a technical link between the required security properties and their implementation in a Linux-based environment.

Table 3 — Linux and Hardware security mechanisms

Security Mechanism	Primary security purpose	Relevant platform layer
UEFI Secure Boot	Establishes an authenticated chain of trust from firmware to bootloader and operating system.	Hardware / Boot
TPM 2.0	Provides hardware-backed protection for keys and platform integrity measurements	Hardware / Boot
TPM measured boot	Records measurements of firmware, bootloader, kernel and other boot components to support platform integrity verification	Boot
UEFI firmware controls	Restricts hardware configuration, boot sources and pre-OS interfaces	Hardware / Boot
GRUB bootloader protection	Protects kernel selection, boot parameters and recovery functions against unauthorised modification	Boot
Signed Linux kernel / kernel modules	Prevents execution of unauthorised or modified privileged code	Boot / Kernel
Linux MMU / virtual memory	Provides memory isolation between processes and between user and kernel space	Hardware / Kernel
CPU privilege levels	Separates privileged kernel execution from unprivileged user-space execution	Hardware / Kernel
NX/XD	Prevents execution of code from memory regions designated as non-executable	Hardware / Kernel
ASLR	Randomises memory locations to make exploitation of memory-corruption vulnerabilities more difficult	Kernel / WSCA

Security Mechanism	Primary security purpose	Relevant platform layer
SMEP	Prevents the kernel from executing code located in user-space memory	Hardware / Kernel
SMAP	Prevents unintended kernel access to user-space memory	Hardware / Kernel
IOMMU	Restricts DMA-capable devices from accessing unauthorised memory regions	Hardware / Kernel
Hardware-assisted virtualisation	Provides hardware support for isolation between virtual machines or execution environments	Hardware / Isolation
Linux users and groups	Provides identity-based separation and basic access control	Kernel / Services / Applications
Linux filesystem permissions	Restricts read, write and execute access to files and directories	Kernel / Services / Applications / WSCA
Linux capabilities	Splits root privileges into individually controllable privileges and supports least privilege	Kernel / Services / Containers / WSCA
SELinux	Provides mandatory access control and confinement of processes and resources	Kernel / Services / Applications / Containers / WSCA
AppArmor	Provides mandatory application access-control and confinement policies	Kernel / Services / Applications / Containers / WSCA
seccomp	Restricts the system calls available to a process and reduces its kernel attack surface	Kernel / Applications / Containers / WSCA
Linux namespaces	Isolates processes and resources between execution environments	Kernel / Containers / WSCA
PID namespaces	Isolates process visibility and process identifiers	Containers / Isolation
User namespaces	Isolates user and group identities and allows restricted container root identities	Containers / Isolation
Mount namespaces	Isolates filesystem mount points and restricts visibility of host filesystems	Containers / Isolation / WSCA
Network namespaces	Isolates network interfaces, routes and network stacks	Containers / Isolation
IPC namespaces	Isolates inter-process communication resources	Containers / Isolation
cgroups	Controls and limits CPU, memory and other resource consumption	Kernel / Containers / Services
systemd service isolation	Restricts service privileges, resources, filesystem access and available system functionality	System Services / WSCA
systemd NoNewPrivileges	Prevents a process and its children from acquiring additional privileges	Services / WSCA
systemd filesystem restrictions	Restricts service access to host filesystems and sensitive directories	Services / WSCA
systemd device restrictions	Restricts access to hardware devices exposed through /dev	Services / WSCA
systemd namespace isolation	Creates additional isolation boundaries for system services	Services / WSCA

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

Security Mechanism	Primary security purpose	Relevant platform layer
Container runtime	Enforces and manages container isolation using Linux kernel security mechanisms	Container / Isolation
Container image signing / verification	Ensures that only authorised and trusted container images are deployed	Container / Isolation
Container capability restrictions	Removes unnecessary Linux privileges from container processes	Container / Isolation
Container device controls	Prevents unnecessary access to host hardware devices	Container / Isolation
Filesystem / volume controls	Restricts container and WSCA access to persistent data	Containers / WSCA
nftables	Restricts network connections, ports, protocols and communication paths	Kernel / Services / Applications / WSCA
Linux routing controls	Controls permitted network paths and prevents unintended network connectivity	Kernel / Network
LUKS2 / dm-crypt	Protects stored data against unauthorised access when storage is not otherwise protected	Hardware / Storage
auditd / Linux Audit	Records security-relevant system calls, authentication, privilege and policy events	Kernel / Services / Applications
systemd-journald	Collects and manages system and service security-relevant events	Services / Applications / WSCA
File integrity monitoring	Detects unauthorised modification of executables, libraries and configuration	Services / Applications / WSCA
Kernel hardening configuration	Reduces exploitable kernel functionality and strengthens kernel security boundaries	Kernel
Kernel module restrictions	Prevents unauthorised kernel extensions from introducing privileged code	Kernel
Device access controls	Restricts processes from directly accessing hardware devices	Kernel / Containers / WSCA
Package management and security updates	Maintains platform components against known vulnerabilities	All software layers
Vulnerability scanning / dependency management	Identifies known vulnerabilities in Linux packages, libraries, containers and dependencies	All software layers
Security monitoring / SIEM integration	Collects and correlates security events from logs, audit facilities, integrity monitoring and runtime sensors; supports alerting and incident response.	All layers

6.2 Hardware / Host System for Linux

The hardware layer provides security mechanisms that are either directly enforced by the processor and firmware or exposed to and controlled by Linux. Relevant mechanisms include CPU privilege levels, the Memory Management Unit (MMU), NX/XD, SMEP, SMAP, IOMMU,

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

hardware random-number generation, TPM 2.0, Secure Boot, measured boot, hardware-assisted virtualisation and hardware device controls.

The Linux kernel uses these mechanisms to establish separation between kernel and user space, protect process memory and control access to hardware resources. Where supported, a TPM can provide a hardware root of trust for platform integrity measurements and protected key operations.

ID	Security Principle	Requirement Implementation for Linux
HW-01	**Minimal Configuration**	Only hardware required for the WSCA platform shall be present or enabled. Unused devices and interfaces shall be disabled in firmware and shall not be exposed to Linux.
HW-02	**Limited Execution**	Processor execution-control mechanisms such as NX/XD, SMEP and SMAP shall be enabled where supported. These shall prevent execution or inappropriate access to memory in ways that could bypass Linux security controls.
HW-03	**Least Privilege**	Direct hardware access shall be mediated by the Linux kernel. User-space processes shall not have direct access to privileged hardware resources unless explicitly required.
HW-04	**Process Isolation**	The processor's privilege levels and MMU shall be used to maintain separation between kernel space and user space and between individual processes.
HW-05	**Memory Protection**	Hardware memory protection shall be enabled and used by Linux to prevent one process from accessing another process's memory.
HW-06	**Controlled Interfaces**	Unused physical interfaces such as USB, Bluetooth and wireless interfaces shall be disabled or physically restricted. Unauthorised boot devices shall be prevented.
HW-07	**Platform Integrity**	Secure Boot and TPM-based measured boot shall be used where supported to establish and verify the integrity of the Linux boot chain.
HW-08	**Network Protection**	Only required physical network interfaces shall be enabled. Unused interfaces shall be disabled at firmware or Linux level.
HW-09	**WSCA Data Protection**	Storage containing WSCA security-relevant data shall be physically protected. Linux disk encryption, such as LUKS2, shall be used where protection of data at rest is required.
HW-10	**Privilege Escalation Protection**	Hardware privilege separation, NX/XD, SMEP, SMAP and other available processor protections shall be enabled to prevent user-space software from bypassing kernel protection.
HW-11	**Security-Relevant Logging**	Relevant hardware and firmware security events exposed to Linux shall be collected through the system logging or auditing mechanisms.
HW-12	**Vulnerability Management**	Firmware, microcode, hardware drivers and other hardware-dependent components shall be included in the platform vulnerability-management process.
HW-13	**Cyber Security Monitoring**	Where supported, hardware and firmware security events—including integrity failures, hardware faults and indicators of unauthorised physical or logical access—shall be exposed to platform monitoring mechanisms. Events shall be analysed and, where warranted, trigger alerts and incident response.

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

6.3 Boot and Initialisation for Linux

A Linux-based platform normally uses UEFI firmware, UEFI Secure Boot, a bootloader such as GRUB, a signed Linux kernel, initramfs/early userspace and optionally TPM-based measured boot.

Secure Boot establishes a chain of authenticated executable components. Measured boot can record measurements of firmware, bootloader, kernel and other components in the TPM. Linux can subsequently use these measurements for platform integrity verification or attestation.

ID	Security Principle	Requirement Implementation for Linux
BOOT-01	**Minimal Configuration**	UEFI, GRUB and initramfs shall contain only components required to start the approved Linux system. Unnecessary boot functionality shall be disabled.
BOOT-02	**Limited Execution**	UEFI Secure Boot shall be enabled where supported. Only authorised firmware, bootloader and kernel components shall be executed.
BOOT-03	**Least Privilege**	Access to firmware configuration and boot configuration shall be restricted to authorised administrators.
BOOT-04	**Process Isolation**	The boot chain shall execute only trusted components and shall prevent unauthorised components from being introduced into the initialisation process.
BOOT-05	**Memory Protection**	Hardware memory-protection mechanisms shall be active during boot and shall be inherited by the Linux kernel.
BOOT-06	**Controlled Interfaces**	UEFI configuration, GRUB configuration and recovery/single-user boot functions shall be protected against unauthorised use.
BOOT-07	**Platform Integrity**	Secure Boot shall verify the authenticity/integrity of boot components. TPM measured boot shall be used where appropriate to provide additional integrity evidence.
BOOT-08	**Network Protection**	PXE and other pre-OS network functionality shall be disabled unless explicitly required.
BOOT-09	**WSCA Data Protection**	The boot environment shall not expose WSCA security-relevant data or cryptographic material to unauthorised users or components.
BOOT-10	**Privilege Escalation Protection**	GRUB command-line access, kernel parameters and recovery modes shall be protected against unauthorised modification or use.
BOOT-11	**Security-Relevant Logging**	Boot integrity failures and relevant boot security events shall be recorded where supported.
BOOT-12	**Vulnerability Management**	UEFI, GRUB, kernel, initramfs and associated boot components shall be included in vulnerability management and security updates.
BOOT-13	**Cyber Security Monitoring**	The boot and initialisation process shall support detection and reporting of security-relevant anomalies, including unexpected boot configurations, failed integrity verification and unauthorised changes to boot components. Events shall be made available to platform monitoring mechanisms for investigation and response.

6.4 Operating System Kernel for Linux

The Linux kernel provides the most important security mechanisms for protecting the WSCA. These include process isolation, virtual memory protection, Linux capabilities, namespaces,

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

seccomp, SELinux/AppArmor, kernel module controls, filesystem permissions, device access controls, kernel hardening mechanisms, network controls and audit facilities.

Linux also provides security mechanisms such as ASLR, NX, kernel address-space protection, read-only kernel memory protections, module signing and restrictions on privileged operations.

These mechanisms form the principal platform security boundary between the WSCA and other software executing on the host.

ID	Security Principle	Requirement Implementation for Linux
KERN-01	**Minimal Configuration**	The Linux kernel shall contain only required drivers, filesystems, protocols and functionality. Unnecessary kernel modules shall be removed or disabled.
KERN-02	**Limited Execution**	Kernel module loading and privileged kernel extensions shall be restricted. Module signing shall be enabled where supported, and unauthorised modules shall not be loaded.
KERN-03	**Least Privilege**	The WSCA shall execute as a dedicated non-root user wherever possible. Linux capabilities shall be used instead of unrestricted root privileges.
KERN-04	**Process Isolation**	Linux process isolation shall be enforced through normal process separation and, where appropriate, namespaces and SELinux/AppArmor.
KERN-05	**Memory Protection**	ASLR, NX/XD, SMEP, SMAP and other available kernel memory protections shall be enabled. Processes shall not be able to access WSCA memory belonging to another process.
KERN-06	**Controlled Interfaces**	Access to system calls, /proc, /sys, devices, IPC and other kernel interfaces shall be restricted. seccomp shall be used where appropriate to reduce the WSCA system-call attack surface.
KERN-07	**Platform Integrity**	Kernel configuration, modules and security settings shall be protected against unauthorised modification.
KERN-08	**Network Protection**	Linux networking shall expose only required interfaces, protocols and ports. nftables or equivalent firewall controls shall restrict network communication.
KERN-09	**WSCA Data Protection**	Linux permissions, capabilities and SELinux/AppArmor policies shall prevent unrelated processes from accessing WSCA files, devices and other protected resources.
KERN-10	**Privilege Escalation Protection**	Linux capabilities, SELinux/AppArmor, seccomp and kernel hardening mechanisms shall be used to reduce the possibility of privilege escalation.
KERN-11	**Security-Relevant Logging**	auditd, journald and kernel audit facilities shall record relevant security events.
KERN-12	**Vulnerability Management**	The Linux kernel and security-relevant modules and drivers shall be maintained using a controlled security-update process.
KERN-13	**Cyber Security Monitoring**	The operating system kernel shall support detection and reporting of security-relevant anomalies, including unexpected kernel-module loading, access-control violations, suspicious privileged activity and indicators of attempted kernel exploitation. Events shall be made available to platform monitoring mechanisms for analysis and incident response.

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

6.5 Operating System / System Services for Linux

Linux system services are commonly managed through systemd. Security mechanisms include systemd service isolation and sandboxing, Linux users and groups, capabilities, SELinux/AppArmor, filesystem permissions, network namespaces, cgroups, firewall controls and audit/logging facilities.

Systemd provides additional controls such as NoNewPrivileges, filesystem restrictions, device restrictions, private temporary directories and namespace-based isolation.

ID	Security Principle	Requirement Implementation for Linux
OS-01	**Minimal Configuration**	Only system services required for WSCA operation, administration or security monitoring shall be installed and enabled.
OS-02	**Limited Execution**	Systemd shall start only authorised services and binaries. Service executables shall be protected against modification.
OS-03	**Least Privilege**	Services shall run under dedicated non-root accounts. Linux capabilities and systemd NoNewPrivileges shall be used where applicable.
OS-04	**Process Isolation**	Services shall be isolated using SELinux/AppArmor and systemd sandboxing where appropriate.
OS-05	**Memory Protection**	Normal Linux process memory isolation shall prevent services from accessing WSCA memory.
OS-06	**Controlled Interfaces**	Service sockets, files, devices and IPC interfaces shall be restricted using Linux permissions and mandatory access controls.
OS-07	**Platform Integrity**	Service binaries and configuration files shall be writable only by authorised administrative processes.
OS-08	**Network Protection**	Only required network services shall listen on the host. nftables shall restrict access to permitted sources and destinations.
OS-09	**WSCA Data Protection**	Services shall have no access to WSCA security-relevant data unless explicitly required. SELinux/AppArmor shall enforce this where appropriate.
OS-10	**Privilege Escalation Protection**	Services shall use restricted capabilities, NoNewPrivileges and appropriate sandboxing to prevent privilege escalation.
OS-11	**Security-Relevant Logging**	Systemd/journald and audit facilities shall record security-relevant service events.
OS-12	**Vulnerability Management**	Installed system services and packages shall be regularly updated and unsupported services shall be removed.
OS-13	**Cyber Security Monitoring**	System services shall be monitored for security-relevant anomalies, including unexpected behaviour, unauthorised configuration changes, suspicious process activity and indicators of compromise. Detected events shall be forwarded to platform monitoring mechanisms for analysis, alerting and incident response.

6.6 Application Space for Linux

Linux provides several mechanisms to control application software, including Unix users/groups, filesystem permissions, Linux capabilities, SELinux/AppArmor, seccomp, systemd sandboxing, namespaces, cgroups and executable/file integrity controls.

The key objective at this layer is to ensure that an unrelated application cannot become an attack path towards the WSCA.

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

ID	Security Principle	Requirement Implementation for Linux
APP-01	**Minimal Configuration**	Only software required for WSCA operation and authorised administration shall be installed.
APP-02	**Limited Execution**	Execution shall be restricted to authorised applications. SELinux/AppArmor and appropriate filesystem permissions shall prevent unauthorised execution or modification.
APP-03	**Least Privilege**	Applications shall use dedicated users and shall not execute with root privileges unless explicitly justified.
APP-04	**Process Isolation**	Applications shall be isolated from the WSCA using Linux process isolation, SELinux/AppArmor and, where appropriate, namespaces.
APP-05	**Memory Protection**	Linux process memory isolation shall prevent applications from reading or modifying WSCA memory.
APP-06	**Controlled Interfaces**	Access to files, devices, sockets, /proc, /sys and other system interfaces shall be restricted.
APP-07	**Platform Integrity**	Application binaries, libraries and configuration shall be protected against unauthorised modification.
APP-08	**Network Protection**	Applications shall have no unrestricted network access. Linux firewall rules shall restrict network communication.
APP-09	**WSCA Data Protection**	SELinux/AppArmor and filesystem permissions shall prevent applications from accessing WSCA data unless explicitly authorised.
APP-10	**Privilege Escalation Protection**	Applications shall operate without unnecessary capabilities. seccomp and mandatory access controls shall be used where appropriate.
APP-11	**Security-Relevant Logging**	Security-relevant application events shall be recorded through journald, auditd or equivalent mechanisms.
APP-12	**Vulnerability Management**	Application packages and dependencies shall be maintained and vulnerable or unsupported applications shall be removed or updated.
APP-13	**Cyber Security Monitoring**	Applications in the WSCA environment shall be monitored for security-relevant anomalies, including unauthorised execution, unexpected application behaviour, suspicious process activity and indicators of compromise. Detected events shall be forwarded to platform monitoring mechanisms for analysis, alerting and incident response.

6.7 Container / Isolation Layer for Linux

Linux provides the fundamental mechanisms used by modern container technologies. These include namespaces, cgroups, Linux capabilities, seccomp, SELinux/AppArmor, filesystem namespaces, device cgroups and Linux network isolation.

The container runtime, such as containerd, Podman or Docker, combines these kernel mechanisms to create an isolated execution environment.

For an AVA_VAN.5 evaluation, the container boundary should therefore not be treated as an independent security mechanism. Its security depends substantially on the correct configuration of the Linux kernel and container runtime.

ID	Security Principle	Requirement Implementation for Linux
CONT-01	**Minimal Configuration**	Container images shall use a minimal base image and contain only software required by the WSCA.

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

ID	Security Principle	Requirement Implementation for Linux
CONT-02	**Limited Execution**	Only authorised and integrity-verified container images shall be executed.
CONT-03	**Least Privilege**	Containers shall run as non-root users where possible. Unnecessary Linux capabilities shall be dropped.
CONT-04	**Process Isolation**	PID, user, mount, network and IPC namespaces shall isolate the WSCA container from the host and other containers.
CONT-05	**Memory Protection**	cgroups shall limit container resource consumption. Container processes shall not be able to access host or other-container memory.
CONT-06	**Controlled Interfaces**	Host filesystem mounts and /dev access shall be minimised. The container shall not receive unnecessary host privileges.
CONT-07	**Platform Integrity**	Container images shall be authenticated and integrity-verified before deployment. Runtime configuration shall be protected against unauthorised modification.
CONT-08	**Network Protection**	Container networking shall permit only explicitly required connections. Unrestricted Internet connectivity shall be prevented.
CONT-09	**WSCA Data Protection**	Volumes containing WSCA data shall be accessible only to authorised containers and processes.
CONT-10	**Privilege Escalation Protection**	Privileged containers shall not be used. CAP_SYS_ADMIN and other unnecessary capabilities shall be removed. Seccomp and SELinux/AppArmor shall be applied where appropriate.
CONT-11	**Security-Relevant Logging**	Container runtime and security events shall be logged and made available to the platform monitoring infrastructure.
CONT-12	**Vulnerability Management**	Container images, runtime components, Linux kernel components and dependencies shall be subject to vulnerability management.
CONT-13	**Cyber Security Monitoring**	The container runtime and isolation mechanisms shall be monitored for security-relevant anomalies, including unauthorised container creation or modification, attempted isolation-boundary violations, unexpected resource access and suspicious workload behaviour. Detected events shall be forwarded to platform monitoring mechanisms for analysis, alerting and incident response.

6.8 WSCA Execution Environment for Linux

The WSCA execution environment can combine the Linux mechanisms described above into a dedicated security boundary around the WSCA. The principal mechanisms are dedicated Linux user/group identities, filesystem permissions, SELinux/AppArmor, Linux capabilities, seccomp, systemd sandboxing, namespaces, cgroups, firewall controls, ASLR/NX and integrity protection of the WSCA executable and libraries.

This layer should provide the closest and most direct protection of the WSCA. It is therefore particularly relevant when demonstrating that the underlying platform does not introduce an attack path that would undermine the WSCA's AVA_VAN.5 assessment.

ID	Security Principle	Requirement Implementation for Linux
WSCA-01	**Minimal Configuration**	The WSCA execution environment shall contain only the libraries, runtime components, configuration and supporting functionality required by the WSCA.

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

ID	Security Principle	Requirement Implementation for Linux
WSCA-02	**Limited Execution**	The WSCA executable and libraries shall be integrity-protected and only authorised versions shall be executed.
WSCA-03	**Least Privilege**	The WSCA shall run under a dedicated non-root Linux account with only the required capabilities.
WSCA-04	**Process Isolation**	The WSCA shall be isolated from unrelated processes using SELinux/AppArmor, namespaces and/or systemd sandboxing as appropriate.
WSCA-05	**Memory Protection**	ASLR and NX/XD shall be enabled. Linux process isolation shall prevent other processes from accessing WSCA memory.
WSCA-06	**Controlled Interfaces**	The WSCA shall have access only to explicitly required files, devices, sockets and system calls. seccomp and SELinux/AppArmor shall be used where appropriate to enforce this restriction.
WSCA-07	**Platform Integrity**	The WSCA executable, libraries and security configuration shall be protected against unauthorised modification.
WSCA-08	**Network Protection**	The WSCA shall communicate only with authorised endpoints. Linux firewall rules shall prevent unintended network connections.
WSCA-09	**WSCA Data Protection**	WSCA credentials, cryptographic material and security-relevant configuration shall be protected using restrictive filesystem permissions and mandatory access controls.
WSCA-10	**Privilege Escalation Protection**	The WSCA shall execute with no unnecessary capabilities and shall use NoNewPrivileges, seccomp and mandatory access control where appropriate.
WSCA-11	**Security-Relevant Logging**	Security-relevant events affecting the WSCA execution environment shall be recorded and protected against unauthorised alteration.
WSCA-12	**Vulnerability Management**	All WSCA runtime libraries and platform dependencies shall be included in the vulnerability-management process.
WSCA-13	**Cyber Security Monitoring**	The WSCA execution environment shall be monitored for security-relevant anomalies, including unauthorised changes to runtime components or configuration, unexpected WSCA behaviour, suspicious resource access and indicators of compromise. Detected events shall be forwarded to platform monitoring mechanisms for analysis, alerting and incident response.

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