WM-02-01/Annex B

PL EUDI Wallet Certification System

Test plan

VERSION 1.01

2026.07.21

Document reference	WM-02-01/Annex B
Version	1.01
Status	Final
Date	21.07.2026
Document type	Test Plan
Parent framework	WM-02-01
Scheme Owner	Republic of Poland / Minister of Digital Affairs
Certification target	EUDI Wallet Unit cybersecurity functionality
Primary audience	Conformity Assessment Bodies (CABs)
Secondary audience	EUDI Wallet Providers, PID Providers
Assurance level <identity proofing CIR 2015/1502> or <evaluation, EU Regulation 2019/881>	High (EU Regulation 2019/881)
Subsidiary Documents	None

Table of Content

AUTH — Authentication and authorization **4**
LOG — Logging **27**
SRV — Services **35**
STRG — Local storage **53**
NET — Network & Interfaces **59**
SUG — User guidance documents **70**
SECDOC — Secure use of the TOE **79**
HRDWSEC — Additional security measures **87**
CRYPTO — Cryptography **92**

AUTH — Authentication and authorization

Table — VE and TE of AS.Auth1 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth1	**The TOE shall authenticate the user before he/she can access the TOE's functions or data provided by the TOE.**
Required vendor information	
VE.Auth1.01	The vendor shall identify the TOE functions and TOE-provided data that require User authentication, including the evaluated locked/authenticated states.
VE.Auth1.02	The vendor shall provide SUG and test instructions sufficient to install the TOE, register a test User, lock the TOE, authenticate the User, and trigger representative protected access paths.
Required test procedures	
TE.Auth1.01	• **Test description with test expectation:** The TOE shall block access to protected TOE functions/data before successful User authentication. The expected behaviour is authentication prompt or safe denial before any protected data is shown or function executed. • **Test preparation:** Install the TOE in the evaluated secure configuration and prepare one registered test User. Select a representative protected function/data item from VE.Auth1.01. • **Testing steps:** The tester records the TOE version, platform, evaluated configuration, selected protected function/data item and initial unauthenticated/locked state in the test evidence. The tester attempts to access the selected protected function/data item through the normal in-app path and observes the TOE response before any successful User authentication. • **Test result:** The actual result is the observed pre-authentication access behaviour: authentication prompt, safe denial, redirection to login/unlock flow, protected data hidden, protected function blocked, or protected data/function accessible before authentication. The result shall describe the observed response and whether any protected content or operation became available. • **Assessment (pass/fail):** PASS if access is blocked until successful User authentication. FAIL if protected data is disclosed or the protected function is executed before authentication.
TE.Auth1.02	• **Test description with test expectation:** The same protected TOE function/data shall become available after successful User authentication. The expected behaviour is that access is granted only in the authenticated state. • **Test preparation:** Use the same TOE, User and protected sample as in TE.Auth1.01. Ensure that the TOE is ready for User authentication using the evaluated authentication method. • **Testing steps:** The tester records the TOE version, platform, selected protected function/data item and authentication method in the test evidence. The tester authenticates the User successfully and repeats the same access attempt used in TE.Auth1.01 through the normal in-app path. • **Test result:** The actual result is the observed post-authentication access behaviour: the protected function/data item becomes available, remains blocked with a documented reason, remains blocked without clear reason, or behaves inconsistently with TE.Auth1.01. The result shall describe the authenticated state and the access outcome. • **Assessment (pass/fail):** PASS if access is granted after successful User authentication. FAIL if access remains unavailable without documented reason or if access was already available before authentication.
TE.Auth1.03	• **Test description with test expectation:** Protected TOE access shall remain authentication-gated when started from declared alternative entry paths. This is limited to entry paths documented by the vendor for the evaluated configuration, not to vulnerability discovery. • **Test preparation:** From VE.Auth1.02 select each declared alternative entry path that can start access to protected TOE function/data. Use the same protected sample type as in TE.Auth1.01 where possible. • **Testing steps:** The tester records the TOE version, platform, selected declared alternative entry path, selected protected function/data item and initial unauthenticated/locked state in the test evidence. The tester starts the protected action through the selected declared alternative entry path and observes whether the TOE requires authentication before protected access or execution. • **Test result:** The actual result is the observed alternative-entry-path behaviour: authentication prompt, safe denial, safe failure, protected data hidden, protected function blocked, or protected

Element	Content
	data/function accessible before authentication. The result shall identify the tested entry path and whether it produced a weaker authentication gate than the normal in-app path. • **Assessment (pass/fail):** PASS if every tested declared alternative entry path requests authentication or fails safely before access. FAIL if any tested entry path exposes protected data or executes a protected function without authentication.
Test Verdict	PASS iff TE.Auth1.01, TE.Auth1.02 and TE.Auth1.03 all PASS. FAIL if any TE fails or if missing VE evidence prevents the tester from executing the test.

Table — VE and TE of AS.Auth2 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth2	**The TOE shall support registration with a wallet provider backend service that is secured in accordance with the security requirements of the assets. Authentication factor generated during registration, e.g. cryptographic key material, may not be susceptible to manipulation, duplication, or extraction. Application Note: The registration is intended to ensure that the wallet backend can unambiguously identify the TOE during subsequent operations and verify its integrity during authentication.**
Required vendor information	
VE.Auth2.01	The vendor shall provide the registration specification, including the expected secured registration flow, asset-security requirements, and lifecycle of the registration authentication factor.
VE.Auth2.02	The vendor shall provide a test backend or test access, test accounts/devices, and conformance-level negative test instructions for registration, subsequent TOE recognition, and factor misuse checks.
Required test procedures	
TE.Auth2.01	• **Test description with test expectation:** A fresh TOE instance shall register with the Wallet Provider backend only through the specified secured registration flow. The expected result is a successful registration bound to the tested TOE instance. • **Test preparation:** Prepare a fresh TOE instance, backend test account, and test backend access according to VE.Auth2.02. Enable only the evaluated registration configuration. • **Testing steps:** The tester records the TOE version, platform, evaluated registration configuration, backend/test-backend identifier and test account/device identifier in the test evidence. The tester executes the vendor-specified secured registration flow and observes both the TOE-side registration outcome and the backend-side registration state using non-secret evidence only. • **Test result:** The actual result is the observed registration behaviour: registration succeeds through the specified secured flow, registration fails, backend does not recognise the TOE instance, or registration succeeds outside the specified flow. The result shall describe the TOE-side status, backend-side status and non-secret evidence category. • **Assessment (pass/fail):** PASS if registration succeeds through the specified secured flow and produces a backend-recognised TOE registration. FAIL if registration cannot be completed or succeeds outside the specified flow.
TE.Auth2.02	• **Test description with test expectation:** After registration, the backend shall recognise the registered TOE during a subsequent operation requiring TOE identification or integrity/authentication verification. An unregistered TOE shall not be accepted for the same operation. • **Test preparation:** Use the TOE registered in TE.Auth2.01 and prepare a second fresh unregistered TOE instance. Select one backend operation that requires registered TOE recognition. • **Testing steps:** The tester records the TOE version, platform, registered TOE instance identifier, unregistered TOE instance identifier and selected backend operation in the test evidence. The tester executes the selected operation from the registered TOE and then from the fresh unregistered TOE, and compares the backend decision and TOE-side outcome for both attempts. • **Test result:** The actual result is the observed TOE-recognition behaviour: registered TOE accepted and unregistered TOE rejected, both accepted, both rejected, or backend decision ambiguous. The result shall describe the backend decision category and TOE-side outcome for each TOE instance without recording secrets or reusable authentication artefacts.

Element	Content
	• **Assessment (pass/fail):** PASS if the registered TOE is accepted and the unregistered TOE is rejected for the selected operation. FAIL if the backend cannot distinguish the registered TOE from the unregistered TOE.
TE.Auth2.03	• **Test description with test expectation:** The registration authentication factor shall not be usable when manipulated, duplicated, or extracted through conformance-level test means. The expected result is safe rejection and no reusable factor obtained through supported interfaces. • **Test preparation:** Prepare the registered TOE, a second fresh TOE, and the vendor-defined negative test method from VE.Auth2.02. Do not use exploit development, reverse engineering, or source-code analysis in this test. • **Testing steps:** The tester records the TOE version, platform, selected negative check, target factor-misuse category and test method from VE.Auth2.02 in the test evidence. The tester executes the vendor-defined manipulation, duplication or supported export/extraction check and then attempts registration or subsequent authentication using the resulting artefact or state. • **Test result:** The actual result is the observed factor-misuse behaviour: no artefact/state obtained, artefact/state obtained but rejected, manipulated/duplicated/exported state accepted, or outcome inconclusive. The result shall describe whether backend registration or authentication accepted the resulting artefact/state, without recording authentication-factor values or reusable secrets. • **Assessment (pass/fail):** PASS if each negative check fails safely and no reusable authentication factor is obtained or accepted. FAIL if a manipulated, duplicated or supported-exported factor is accepted.
Test Verdict	PASS iff TE.Auth2.01, TE.Auth2.02 and TE.Auth2.03 all PASS. FAIL if any TE fails or if missing VE evidence prevents the tester from executing the test.

Table — VE and TE of AS.Auth3 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth3	**The TOE shall support creation and maintenance of binding between the Wallet Instance and user assigned keys managed by the WSCA.**
Required vendor information	
VE.Auth3.01	The vendor shall provide the binding specification describing how the Wallet Instance creates and maintains binding with user assigned keys managed by the WSCA. The specification shall identify the observable binding states and the operations proving that the binding is usable.
VE.Auth3.02	The vendor shall provide a test configuration with a suitable WSCA or WSCA test environment, test User, and instructions for creating, observing, and reusing the binding.
Required test procedures	
TE.Auth3.01	• **Test description with test expectation:** The TOE shall create a binding between the Wallet Instance and user assigned keys managed by the WSCA. The expected result is an observable bound state and successful completion of one operation that requires the bound key. • **Test preparation:** Prepare a fresh TOE instance, test User, and WSCA test configuration according to VE.Auth3.02. Ensure that no previous binding exists for the selected test User and TOE instance. • **Testing steps:** The tester records the TOE version, platform, WSCA test configuration, test User, selected binding creation flow and selected key-dependent operation in the test evidence. The tester executes the vendor-defined binding creation flow and then executes one vendor-defined operation that requires the Wallet Instance to use the WSCA-managed user assigned key through the created binding. • **Test result:** The actual result is the observed binding-creation behaviour: binding created and key-dependent operation completed, binding creation failed, bound state ambiguous, or key-dependent operation failed after binding creation. The result shall describe the binding state and operation outcome without recording private key material or other secrets. • **Assessment (pass/fail):** PASS if the binding is created and the key-dependent operation succeeds through the specified binding. FAIL if the TOE cannot create the binding or the operation does not use the created binding as specified.

Element	Content
TE.Auth3.02	• **Test description with test expectation:** The TOE shall maintain the created binding across normal TOE state changes defined for the evaluated configuration. The expected result is that the same binding remains usable after such state changes. • **Test preparation:** Use the TOE instance and binding created in TE.Auth3.01. Select one normal state change defined by the vendor, such as application restart or user re-authentication. • **Testing steps:** The tester records the TOE version, platform, selected normal state change, pre-change binding state, selected key-dependent operation and any User authentication or re-authentication condition in the test evidence. The tester performs the selected state change and repeats the same key-dependent operation used in TE.Auth3.01. • **Test result:** The actual result is the observed binding-maintenance behaviour: binding remains usable, re-authentication is required and the operation then succeeds, binding is lost, binding is corrupted, binding is unusable, or binding state is ambiguous. The result shall describe the selected state change, post-change binding state and repeated operation outcome. • **Assessment (pass/fail):** PASS if the binding remains valid and the key-dependent operation succeeds after the selected state change. FAIL if the binding is lost, corrupted, or unusable without a documented reason.
TE.Auth3.03	• **Test description with test expectation:** The TOE shall not treat an unbound Wallet Instance as bound to WSCA-managed user assigned keys. The expected result is safe denial or a binding creation requirement before any key-dependent operation proceeds. • **Test preparation:** Prepare a second fresh TOE instance or reset the first instance to a vendor-supported unbound state. Use the same test User only if this is supported by the test configuration. • **Testing steps:** The tester records the TOE version, platform, selected unbound TOE state, selected key-dependent operation and WSCA test configuration in the test evidence. The tester attempts the selected key-dependent operation without completing the binding creation flow and observes whether the TOE requires binding creation or denies the operation safely. • **Test result:** The actual result is the observed unbound-state behaviour: binding creation required, safe denial, operation blocked, operation completed as if bound, or binding state ambiguous. The result shall describe the TOE state, displayed message/status and whether the key-dependent operation was completed. Do not attempt exploitation or unauthorized WSCA manipulation. • **Assessment (pass/fail):** PASS if the TOE does not complete the key-dependent operation before a valid binding exists. FAIL if the TOE proceeds as if the binding existed when it has not been created.
Test Verdict	PASS iff TE.Auth3.01, TE.Auth3.02 and TE.Auth3.03 all PASS. FAIL if any TE fails or if missing VE evidence prevents the tester from executing the test.

Table — VE and TE of AS.Auth4 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth4	**The TOE shall enable users to change authentication factors they have set themselves and to protect them from unauthorized access.**
Required vendor information	
VE.Auth4.01	The vendor shall identify all authentication factors that Users can set themselves and describe how each such factor can be changed in the evaluated configuration.
VE.Auth4.02	The vendor shall provide SUG/test instructions and test accounts allowing the tester to set, change, verify, and attempt unauthorized access to the selected authentication factor. The SUG should match the actual UI and support secure operation of the certified configuration.
Required test procedures	
TE.Auth4.01	• **Test description with test expectation:** An authenticated User shall be able to change an authentication factor that the User has set themselves. The expected result is that the new factor works and the old factor no longer works, where technically applicable. • **Test preparation:** Prepare a TOE instance with one User-set authentication factor selected from VE.Auth4.01. Ensure the tester knows the valid current factor and can complete User authentication. • **Testing steps:** The tester records the TOE version, platform, selected factor type, evaluated configuration and vendor-defined change flow in the test evidence, without recording factor values.

Element	Content
	The tester authenticates as the User, executes the change flow, locks or resets the authentication state, and then attempts authentication with the old factor and with the new factor. • **Test result:** The actual result is the observed factor-change behaviour: change completed and only the new factor is accepted, change failed, old factor remains accepted, new factor is rejected, or outcome is ambiguous. The result shall describe the factor type and authentication outcomes without recording the actual factor value. • **Assessment (pass/fail):** PASS if the factor can be changed by the authenticated User and only the new factor is accepted afterwards. FAIL if the change cannot be completed or the old factor remains accepted without documented reason.
TE.Auth4.02	• **Test description with test expectation:** An unauthenticated or unauthorized actor shall not be able to change the User-set authentication factor. The expected result is authentication prompt, safe denial, or unchanged factor state. • **Test preparation:** Use the TOE instance from TE.Auth4.01 and put it in unauthenticated/locked state. Prepare an incorrect current factor or a second test User if the TOE supports multiple Users. • **Testing steps:** The tester records the TOE version, platform, selected factor type, unauthenticated/locked or unauthorized state, attempted change path and expected denial outcome in the test evidence. The tester attempts to access or complete the factor change flow without successful authorization and then verifies whether the previously valid factor state has changed. • **Test result:** The actual result is the observed unauthorized-change behaviour: authentication prompt, safe denial, unchanged factor state, factor changed, factor reset, factor disabled, or outcome ambiguous. The result shall describe the attempted change path and factor-state outcome. Do not use exploit development or storage inspection in this test. • **Assessment (pass/fail):** PASS if the TOE prevents the unauthorized change and preserves the previous valid factor state. FAIL if the factor is changed, reset, or disabled without proper authorization.
TE.Auth4.03	• **Test description with test expectation:** The TOE shall not disclose the existing User-set authentication factor through normal factor-management screens or supported export/help flows. The expected result is that the factor can be changed or reset only through the specified protected procedure, not viewed in clear form. • **Test preparation:** Use a TOE instance with the selected User-set factor configured. Identify the normal UI or supported flows related to viewing, changing, resetting, or exporting security settings. • **Testing steps:** The tester records the TOE version, platform, selected factor type, tested normal flows and tested authorization state in the test evidence. The tester navigates through the identified normal flows as an authenticated User and, where applicable, as an unauthenticated User, and observes whether the existing factor value or equivalent reusable secret is displayed or exported. • **Test result:** The actual result is the observed factor-disclosure behaviour: no factor value disclosed, only reset/change option available, reusable secret exposed, factor value displayed, factor exported, or outcome ambiguous. The result shall identify the tested flows and disclosure category; screenshots shall avoid capturing any secret value if exposure unexpectedly occurs. • **Assessment (pass/fail):** PASS if no existing factor value or reusable secret is disclosed through tested normal flows. FAIL if the TOE displays or exports such information in a reusable form.
Test Verdict	PASS iff TE.Auth4.01, TE.Auth4.02 and TE.Auth4.03 all PASS. FAIL if any TE fails or if missing VE evidence prevents the tester from executing the test.

Table — VE and TE of AS.Auth5 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth5	**The TOE shall support deletion of binding between the Wallet Instance and user assigned keys managed by the WSCA, when TOE is installed on another device or when the user deactivates the application via a dedicated deactivation option within the user interface.**
Required vendor information	
VE.Auth5.01	The vendor shall provide the binding deletion specification, including the deactivation UI flow and the vendor-defined flow for installation or activation on another device. The specification shall define the observable “bound”, “unbound/deactivated”, and “re-bound” states.

Element	Content
VE.Auth5.02	The vendor shall provide test instructions, two test devices or equivalent test instances, a test User, and WSCA/test backend access allowing the tester to create, delete, and verify the binding state.
Required test procedures	
TE.Auth5.01	• **Test description with test expectation:** The TOE shall delete or invalidate the binding when the User deactivates the application through the dedicated UI option. After deactivation, an operation requiring the previous WSCA-managed user assigned key shall not proceed through the deleted binding. • **Test preparation:** Prepare one TOE instance in a bound state with a test User and WSCA/test backend. Select one key-dependent operation that proves the binding is usable before deletion. • **Testing steps:** The tester records the TOE version, platform, initial binding state, selected key-dependent operation and deactivation UI path in the test evidence. The tester executes the selected key-dependent operation before deactivation, then uses the dedicated UI deactivation option and repeats the same key-dependent operation. • **Test result:** The actual result is the observed UI-deactivation behaviour: pre-deactivation operation succeeds, deactivation completes, previous binding becomes unusable, TOE remains bound, post-deactivation operation still completes, or binding state is ambiguous. The result shall describe the deactivation outcome and post-deactivation operation result without recording private key material or authentication secrets. • **Assessment (pass/fail):** PASS if the UI deactivation completes and the previous binding cannot be used afterwards. FAIL if the TOE remains bound or completes the key-dependent operation using the previous binding.
TE.Auth5.02	• **Test description with test expectation:** The TOE shall delete or invalidate the previous binding when the TOE is installed or activated on another device according to the evaluated flow. The previous device shall not remain usable as a bound Wallet Instance for the same binding. • **Test preparation:** Prepare Device A with a bound TOE instance and Device B as a fresh device or fresh test instance. Prepare the vendor-defined “another device” installation or activation scenario from VE.Auth5.01. • **Testing steps:** The tester records the TOE version, platform, Device A identifier, Device B identifier, initial binding state, selected key-dependent operation and vendor-defined another-device scenario in the test evidence. The tester confirms that Device A can perform the selected operation before the scenario, installs or activates the TOE on Device B according to vendor instructions, and then attempts the relevant operation from both Device A and Device B. • **Test result:** The actual result is the observed another-device behaviour: Device A initially operates as bound, Device B activation completes, Device A binding is deleted or invalidated, Device A remains accepted as bound, Device B becomes bound as specified, or the binding state is ambiguous. The result shall describe the binding status and operation outcome for each device without recording authentication factors, private key material or secret tokens. • **Assessment (pass/fail):** PASS if the previous binding on Device A is deleted or invalidated and Device A cannot continue using it. FAIL if Device A remains accepted as bound when the evaluated “another device” scenario requires deletion.
Test Verdict	PASS iff TE.Auth5.01 and TE.Auth5.02 both PASS. FAIL if any TE fails or if missing VE evidence prevents the tester from executing either trigger scenario.

Table — VE and TE of AS.Auth6 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth6	**The TOE shall not disclose any confidential information or allow any inferences to be drawn regarding such information in response to an authentication request.**
Required vendor information	
VE.Auth6.01	The vendor shall identify confidential information relevant to authentication requests and specify the allowed response contents for successful, failed, cancelled, and invalid authentication requests.
VE.Auth6.02	The vendor shall provide test instructions and tools or test endpoints allowing the tester to issue authentication requests and observe TOE-side and requester-side responses.

Element	Content
Required test procedures	
TE.Auth6.01	• **Test description with test expectation:** A successful authentication response shall contain only the information specified as allowed for the evaluated authentication flow. It shall not include confidential information beyond the intended authentication result and required protocol data. • **Test preparation:** Prepare a registered TOE instance, test User, and authentication requester or test endpoint according to VE.Auth6.02. Identify the expected successful response content from VE.Auth6.01. • **Testing steps:** The tester records the TOE version, platform, request type, requester/test-endpoint identifier, expected allowed response categories and observation method in the test evidence. The tester issues a valid authentication request, completes User authentication successfully, and observes the response returned to the requester and any TOE-side message shown to the User. • **Test result:** The actual result is the observed successful-response behaviour: response limited to allowed content, response contains only required protocol data, confidential information exposed, unrelated sensitive fields included, or response content ambiguous. The result shall describe response categories and disclosure status without recording secret values. • **Assessment (pass/fail):** PASS if the response is limited to the specified allowed content and no confidential information is disclosed. FAIL if the TOE returns secret, sensitive, or unrelated confidential information in the response.
TE.Auth6.02	• **Test description with test expectation:** Failed or cancelled authentication requests shall not disclose confidential information. The expected response is a safe failure status or equivalent generic outcome defined by the vendor. • **Test preparation:** Use the same TOE and requester setup as in TE.Auth6.01. Prepare one failed authentication case and one User-cancelled case, where supported by the evaluated flow. • **Testing steps:** The tester records the TOE version, platform, selected failed/cancelled outcome, requester/test-endpoint identifier, expected safe response category and observation method in the test evidence. The tester issues an authentication request, causes the selected failed or cancelled outcome, and observes requester-side response, TOE-side messages and any externally visible status. • **Test result:** The actual result is the observed failed/cancelled-response behaviour: generic failure, cancellation status, safe denial, confidential information disclosed, overly specific error detail exposed, or response status ambiguous. The result shall describe the outcome category and disclosure status without including secret values in screenshots or logs. • **Assessment (pass/fail):** PASS if failed and cancelled outcomes do not disclose confidential information. FAIL if the TOE exposes confidential information in the response, message, status, or externally visible error details.
TE.Auth6.03	• **Test description with test expectation:** Negative authentication responses shall not allow simple inference about confidential information from distinguishable response contents. Comparable negative cases shall return only the vendor-defined safe distinctions. • **Test preparation:** Select a small representative set of negative authentication request variants from VE.Auth6.01. Define which response differences are allowed because they are necessary and non-confidential. • **Testing steps:** The tester records the TOE version, platform, selected negative variants, allowed response differences and observation method in the test evidence. The tester executes the selected negative variants, compares the externally visible responses, and checks whether response differences reveal confidential information or confidential state not intended for the requester. • **Test result:** The actual result is the observed inference-resistance behaviour: responses are equivalent, differences are allowed and non-confidential, differences reveal confidential state, differences are unexplained, or outcome is ambiguous. The result shall describe tested variants, response categories and relevant distinguishable differences; timing micro-analysis is not included unless the scheme later defines it for conformance testing. • **Assessment (pass/fail):** PASS if observed differences do not reveal confidential information or confidential state. FAIL if response differences allow a direct inference about such information.
Test Verdict	PASS iff TE.Auth6.01, TE.Auth6.02 and TE.Auth6.03 all PASS. FAIL if any TE fails or if missing VE evidence prevents the tester from determining allowed and prohibited response contents.

Table — VE and TE of AS.Auth7 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth7	**The TOE shall require user re-authentication after a defined period of inactivity.**
Required vendor information	
VE.Auth7.01	The vendor shall specify the defined inactivity period and the TOE state changes that start, reset, or end this period. The specification shall identify the TOE functions/data for which re-authentication is required after inactivity.
VE.Auth7.02	The vendor shall provide SUG/test instructions enabling the tester to authenticate the User, create an inactive state, trigger re-authentication, and observe the result. If a shortened test timeout is used, the vendor shall justify that it is equivalent to the evaluated timeout mechanism.
Required test procedures	
TE.Auth7.01	• **Test description with test expectation:** After continuous inactivity longer than the defined period, the TOE shall require User re-authentication before protected TOE access continues. The expected result is that no protected function/data is available before successful re-authentication. • **Test preparation:** Install the TOE in the evaluated secure configuration, prepare one registered test User, and determine the inactivity period from VE.Auth7.01. Select one representative protected function/data item. • **Testing steps:** The tester records the TOE version, platform, defined inactivity period, actual wait time, selected protected function/data item and initial authenticated state in the test evidence. The tester authenticates the User, confirms that the selected protected access works, leaves the TOE inactive for longer than the defined period, and attempts the same protected access again. • **Test result:** The actual result is the observed post-inactivity behaviour: re-authentication required, protected data/function blocked, protected data/function available before re-authentication, no re-authentication prompt, or ambiguous state. The result shall state whether any protected function/data became available before successful re-authentication. • **Assessment (pass/fail):** PASS if the TOE requires re-authentication before access and does not expose protected data or execute the protected function. FAIL if protected access continues without re-authentication after the inactivity period.
TE.Auth7.02	• **Test description with test expectation:** Failed or cancelled re-authentication after inactivity shall not restore protected access. The expected result is safe denial or remaining in the re-authentication-required state. • **Test preparation:** Use the TOE state obtained after timeout in TE.Auth7.01. Prepare an incorrect factor or a User-cancel action supported by the evaluated authentication method. • **Testing steps:** The tester records the TOE version, platform, selected protected function/data item, failed/cancelled re-authentication method and expected blocked state in the test evidence. The tester attempts re-authentication and intentionally fails or cancels it, then attempts the same protected access again. • **Test result:** The actual result is the observed failed/cancelled re-authentication behaviour: protected access remains blocked, TOE remains in re-authentication-required state, safe denial is shown, protected access is granted, or outcome is ambiguous. The result shall not include authentication-factor values. • **Assessment (pass/fail):** PASS if protected access remains blocked after failed or cancelled re-authentication. FAIL if the TOE grants protected access despite unsuccessful re-authentication.
TE.Auth7.03	• **Test description with test expectation:** Successful re-authentication after inactivity shall restore access to the protected TOE function/data. The expected result is that the same access denied before re-authentication succeeds afterwards. • **Test preparation:** Use the TOE after the inactivity timeout from TE.Auth7.01. Ensure that the tester can complete the evaluated User re-authentication method. • **Testing steps:** The tester records the TOE version, platform, selected protected function/data item, successful re-authentication method and pre-test timeout state in the test evidence. The tester completes re-authentication successfully and repeats the same protected access attempt used in TE.Auth7.01.

Element	Content
	• **Test result:** The actual result is the observed successful re-authentication behaviour: protected access restored, access still blocked with documented reason, access still blocked without documented reason, access restored before successful re-authentication, or session state unexpectedly reset/lost. The result shall describe the access decision for the selected protected sample. • **Assessment (pass/fail):** PASS if protected access is restored only after successful re-authentication. FAIL if access is not restored without documented reason or was restored before successful re-authentication.
Test Verdict	PASS iff TE.Auth7.01, TE.Auth7.02 and TE.Auth7.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to determine the inactivity period or protected sample.

Table — VE and TE of AS.Auth8 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth8	**Before the TOE transmits authentication data, the authenticity of the remote party (e.g., WSCA or wallet provider backend) shall be verified.**
Required vendor information	
VE.Auth8.01	The vendor shall identify the authentication-data transmission flows and the remote party authenticity checks applied before transmission. The vendor shall define the observable success and failure outcomes for those checks.
VE.Auth8.02	The vendor shall provide test endpoints, test configuration, or equivalent test support for valid and invalid remote-party authenticity scenarios. The evidence shall allow the tester to observe whether authentication data was transmitted without exposing secret values.
Required test procedures	
TE.Auth8.01	• **Test description with test expectation:** For each selected Auth8 flow, the TOE shall verify the authenticity of the remote party before transmitting authentication data. The expected result is successful transmission only after successful remote-party authenticity verification. • **Test preparation:** Configure a valid remote party according to VE.Auth8.02 and select one flow that transmits authentication data. Prepare non-secret observation evidence, such as test endpoint logs or protocol metadata. • **Testing steps:** The tester records the TOE version, platform, selected Auth8 flow, valid remote-party identity/test endpoint, expected authenticity check and observation method in the test evidence. The tester initiates the selected flow from the TOE and observes the remote-party authenticity result before observing the authentication-data transmission decision. • **Test result:** The actual result is the observed valid-remote-party behaviour: authenticity verification succeeds and authentication data is transmitted, verification succeeds but transmission is blocked with documented reason, transmission occurs before verification, or verification/transmission state is ambiguous. The result shall describe only response/status categories, not authentication-data values or secrets. • **Assessment (pass/fail):** PASS if the selected authentication data is transmitted only after successful verification of the remote party authenticity. FAIL if authentication data is transmitted before the authenticity check succeeds or if the authenticity state cannot be evidenced.
TE.Auth8.02	• **Test description with test expectation:** If remote-party authenticity verification fails, the TOE shall not transmit authentication data. The expected result is safe rejection, safe error, retry requirement, or another documented safe outcome before authentication data is transmitted. • **Test preparation:** Use the selected Auth8 flow from TE.Auth8.01 or another representative authentication-data flow from VE.Auth8.01. Prepare one invalid, unauthenticated, untrusted, expired or mismatched remote-party scenario according to VE.Auth8.02. • **Testing steps:** The tester records the TOE version, platform, selected Auth8 flow, invalid remote-party scenario, expected failure outcome and observation method in the test evidence. The tester initiates the selected flow against the invalid remote party and observes the TOE UI, protocol metadata, test endpoint log or backend decision. • **Test result:** The actual result is the observed invalid-remote-party behaviour: transmission blocked, safe error shown, retry/reconfiguration required, authentication data transmitted, or outcome

Element	Content
	ambiguous. The result shall state whether authentication data was transmitted without exposing its values. • **Assessment (pass/fail):** PASS if authentication data is not transmitted when remote-party authenticity verification fails. FAIL if the TOE transmits authentication data to or through an invalid, unauthenticated, untrusted, expired or mismatched remote party.
TE.Auth8.03	• **Test description with test expectation:** The TOE shall not transmit authentication data until the remote-party authenticity check has completed successfully. The expected result is delayed or blocked transmission while the authenticity result is pending, failed, inconclusive or not evidenced. • **Test preparation:** Prepare a selected Auth8 flow and a vendor-supported delayed, interrupted or inconclusive authenticity-check scenario. Define the expected safe outcome and approved observation method before executing the test. • **Testing steps:** The tester records the TOE version, platform, selected Auth8 flow, delayed/inconclusive authenticity condition, expected safe outcome and observation method in the test evidence. The tester initiates the selected flow while remote-party authenticity is pending, failed to complete, or cannot be evidenced, and observes whether transmission is delayed, blocked or performed. • **Test result:** The actual result is the observed ordering behaviour: transmission occurs only after successful authenticity verification, remains blocked while verification is pending/inconclusive, proceeds without evidenced verification, or outcome is ambiguous. The result shall identify the ordering category and whether any authentication-data transmission occurred. • **Assessment (pass/fail):** PASS if authentication data is transmitted only after the authenticity check has completed successfully. FAIL if authentication data is transmitted while the check is pending, failed, inconclusive or not evidenced.
Test Verdict	PASS iff TE.Auth8.01, TE.Auth8.02 and TE.Auth8.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify authentication-data flows, remote-party authenticity checks, valid/invalid scenarios, expected outcomes or observation methods.

Table — VE and TE of AS.Auth9 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth9	**The TOE shall require active confirmation by the authenticated user for the execution of a security-relevant function.**
Required vendor information	
VE.Auth9.01	The vendor shall identify the TOE security-relevant functions that require active User confirmation in the evaluated configuration. The vendor shall define what counts as active confirmation and what observable event proves that the function was executed.
VE.Auth9.02	The vendor shall provide SUG/test instructions, test accounts, and test endpoints or test data allowing the tester to trigger, confirm, decline, and observe execution of representative security-relevant functions.
Required test procedures	
TE.Auth9.01	• **Test description with test expectation:** The TOE shall require active confirmation by an authenticated User before executing the selected security-relevant function. The expected result is that execution occurs only after the User performs the defined confirmation action. • **Test preparation:** Install the TOE in the evaluated secure configuration and authenticate one test User. Select one representative security-relevant function from VE.Auth9.01 and identify its observable execution result. • **Testing steps:** The tester records the TOE version, platform, selected security-relevant function, authenticated User state, defined confirmation action and expected observable execution result in the test evidence. The tester initiates the selected function, observes the confirmation step, performs the defined active confirmation action and then observes whether the function is executed. • **Test result:** The actual result is the observed positive-confirmation behaviour: confirmation step shown, active confirmation performed, function executed after confirmation, function executed before confirmation, function not executed after confirmation, or execution state ambiguous. The

Element	Content
	result shall describe the confirmation action and execution outcome without recording secrets or personal data beyond what is necessary to evidence the result. • **Assessment (pass/fail):** PASS if the function is executed only after successful User authentication and active User confirmation. FAIL if the function executes before active confirmation or without an authenticated User.
TE.Auth9.02	• **Test description with test expectation:** The selected security-relevant function shall not be executed when the authenticated User declines, cancels, or does not perform the defined active confirmation action. The expected result is safe non-execution and no completed security-relevant effect. • **Test preparation:** Use the same TOE, User and selected function as in TE.Auth9.01. Prepare the vendor-defined decline/cancel/no-confirmation path for that function. • **Testing steps:** The tester records the TOE version, platform, selected security-relevant function, negative confirmation path, expected non-execution outcome and observation method in the test evidence. The tester initiates the selected function, reaches the confirmation step, declines, cancels or leaves the step without active confirmation, and observes the TOE state and any backend/test endpoint result. • **Test result:** The actual result is the observed negative-confirmation behaviour: operation cancelled, function not executed, no security-relevant effect completed, partial state change occurred, function executed despite decline/cancel/no confirmation, or outcome ambiguous. The result shall describe the negative action used, TOE-side status and whether any data disclosure, state change or cryptographic/security-relevant operation occurred. • **Assessment (pass/fail):** PASS if the TOE does not execute the function and does not complete the security-relevant effect. FAIL if the function executes despite decline, cancellation, or missing active confirmation.
TE.Auth9.03	• **Test description with test expectation:** The TOE shall not accept active confirmation for the selected security-relevant function from an unauthenticated or expired User state. The expected result is authentication requirement before confirmation can be effective. • **Test preparation:** Put the TOE in unauthenticated, locked, or expired-session state according to vendor instructions. Use the same selected function or an equivalent representative function from VE.Auth9.01. • **Testing steps:** The tester records the TOE version, platform, selected security-relevant function, unauthenticated/locked/expired state and expected authentication requirement in the test evidence. The tester initiates the selected function and attempts to proceed to confirmation or execution without completing User authentication, then observes whether the TOE requires authentication before accepting confirmation. • **Test result:** The actual result is the observed unauthenticated-confirmation behaviour: authentication required before confirmation, confirmation unavailable, operation blocked, confirmation accepted while unauthenticated, function executed, or state ambiguous. The result shall describe the TOE state, authentication requirement and whether the selected function was executed, using only non-secret evidence. • **Assessment (pass/fail):** PASS if the TOE requires User authentication before confirmation can lead to execution. FAIL if the TOE executes the function or accepts confirmation while the User is not authenticated.
Test Verdict	PASS iff TE.Auth9.01, TE.Auth9.02 and TE.Auth9.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify security-relevant functions or active confirmation behaviour.

Table — VE and TE of AS.Auth10 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth10	**Wallet Instance requires re-authentication if the period of inactivity is longer than defined value.**
Required vendor information	
VE.Auth10.01	The vendor shall specify the defined inactivity value, the TOE states that start or reset inactivity measurement, and the protected functions/data affected by this rule.

Element	Content
VE.Auth10.02	The vendor shall provide SUG/test instructions allowing the tester to authenticate the User, create inactivity longer than the defined value, trigger re-authentication, and observe access decisions. If a shortened test value is used, equivalence to the evaluated timeout mechanism shall be justified.
Required test procedures	
TE.Auth10.01	• **Test description with test expectation:** If inactivity lasts longer than the defined value, the Wallet Instance shall require User re-authentication before protected access continues. The expected result is blocked protected access until successful re-authentication. • **Test preparation:** Install the TOE in the evaluated secure configuration and prepare one registered test User. Determine the defined inactivity value from VE.Auth10.01 and select one representative protected function/data item. • **Testing steps:** The tester records the TOE version, platform, defined inactivity value, actual wait time, selected protected function/data item and initial authenticated state in the test evidence. The tester authenticates the User, confirms that the selected protected access works, leaves the TOE inactive for longer than the defined value, and then attempts the same protected access. • **Test result:** The actual result is the observed post-threshold inactivity behaviour: re-authentication required, protected access blocked, protected access available before re-authentication, no re-authentication prompt, or timeout state ambiguous. The result shall state whether protected access was possible before successful re-authentication. • **Assessment (pass/fail):** PASS if the TOE requires re-authentication and blocks protected access after inactivity longer than the defined value. FAIL if protected access continues without re-authentication.
TE.Auth10.02	• **Test description with test expectation:** After inactivity longer than the defined value, failed or cancelled re-authentication shall keep protected access blocked, while successful re-authentication shall restore it. The expected result is access only after successful re-authentication. • **Test preparation:** Use the TOE state obtained after timeout in TE.Auth10.01. Prepare one failed/cancelled re-authentication outcome and one successful outcome using the evaluated authentication method. • **Testing steps:** The tester records the TOE version, platform, selected protected function/data item, failed/cancelled re-authentication method, successful re-authentication method and expected access decisions in the test evidence. The tester first fails or cancels re-authentication and attempts the selected protected access, then completes re-authentication successfully and repeats the same protected access attempt. • **Test result:** The actual result is the observed re-authentication-gate behaviour: access remains blocked after failed/cancelled re-authentication, access is restored after successful re-authentication, access is restored without successful re-authentication, access remains blocked after successful re-authentication, or outcome is ambiguous. The result shall describe the access decision after each outcome without recording authentication-factor values. • **Assessment (pass/fail):** PASS if protected access remains blocked after failed/cancelled re-authentication and is restored only after successful re-authentication. FAIL if access is restored without successful re-authentication or remains unavailable after successful re-authentication without documented reason.
Test Verdict	PASS iff TE.Auth10.01 and TE.Auth10.02 both PASS. FAIL if any TE fails or if VE evidence is insufficient to determine the defined inactivity value or the affected protected access sample.

Table — VE and TE of AS.Auth11 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth11	**Before executing functionality that requires active confirmation by the user, Wallet Instance shall ensure that: • the user has been transparently informed about the intended use. • the user was given the opportunity to confirm or decline the execution.**
Required vendor information	
VE.Auth11.01	The vendor shall identify the Wallet Instance functionalities that require active confirmation and shall specify the intended-use information shown to the User before execution. The vendor shall also define the observable event proving that the functionality was executed or not executed.

Element	Content
VE.Auth11.02	The vendor shall provide SUG/test instructions, test accounts, and test data or endpoints allowing the tester to trigger the selected functionality, observe the User information screen, confirm execution, and decline execution.
Required test procedures	
TE.Auth11.01	• **Test description with test expectation:** The Wallet Instance shall present intended-use information and confirm/decline options before executing the selected functionality. The expected result is that execution occurs only after the User has been shown the intended use and has actively confirmed execution. • **Test preparation:** Install the TOE in the evaluated secure configuration and authenticate one test User. Select one representative functionality from VE.Auth11.01 and prepare the related test data or endpoint. • **Testing steps:** The tester records the TOE version, platform, selected functionality, authenticated User state, expected intended-use information, confirm/decline controls and observation method in the test evidence. The tester initiates the selected functionality, observes the pre-execution screen or flow, performs the vendor-defined confirmation action, and observes the execution event through the TOE UI, test endpoint or backend result. • **Test result:** The actual result is the observed positive intended-use and confirmation behaviour: intended-use information is shown before execution, confirm and decline choices are available, the function executes only after confirmation, the function executes before/without the required information or confirmation, or the execution state is ambiguous. The result shall describe information categories, choice availability and execution outcome without recording secrets or unnecessary personal data. • **Assessment (pass/fail):** PASS if the intended use is shown before execution, both confirm and decline choices are available, and the function executes only after confirmation. FAIL if the function executes before the intended use is shown or without active confirmation.
TE.Auth11.02	• **Test description with test expectation:** The Wallet Instance shall not execute the selected functionality when the User declines, cancels, or does not confirm execution. The expected result is safe non-execution and no completed security-relevant effect. • **Test preparation:** Use the same TOE, User and selected functionality as in TE.Auth11.01. Prepare the vendor-defined decline/cancel/no-confirmation path for the selected functionality. • **Testing steps:** The tester records the TOE version, platform, selected functionality, negative action type, expected non-execution outcome and observation method in the test evidence. The tester initiates the selected functionality, reaches the intended-use/confirmation step, declines, cancels or leaves the step without confirmation, and observes the TOE state and the test endpoint or backend result. • **Test result:** The actual result is the observed negative intended-use and confirmation behaviour: operation cancelled, function not executed, no security-relevant effect completed, partial state change occurred, function executed despite decline/cancel/no confirmation, or outcome is ambiguous. The result shall describe the negative action used, TOE-side status and whether any data disclosure, state change or other security-relevant effect occurred. • **Assessment (pass/fail):** PASS if the selected functionality is not executed and no security-relevant effect is completed after decline, cancellation or missing confirmation. FAIL if the functionality executes despite decline, cancellation or missing confirmation.
Test Verdict	PASS iff TE.Auth11.01 and TE.Auth11.02 both PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the selected functionality, intended-use information, confirm/decline behaviour, or execution evidence.

Table — VE and TE of AS.Auth12 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth12	**Knowledge factors shall be stored securely so that their confidentiality and integrity are preserved.**
Required vendor information	

Element	Content
VE.Auth12.01	The vendor shall identify the knowledge factors supported or stored by the TOE in the evaluated configuration and describe how their confidentiality and integrity are preserved. The specification shall identify whether each factor state is TOE-managed, platform-managed, OS-managed, WSCA/WSCD-managed or backend-managed.
VE.Auth12.02	The vendor shall provide SUG/test instructions, test accounts, selected knowledge factors, normal state-change scenarios and one vendor-supported negative storage-integrity scenario allowing the tester to verify confidentiality and integrity without recording the factor value.
Required test procedures	
TE.Auth12.01	• **Test description with test expectation:** The selected knowledge factor or reusable equivalent shall not be disclosed through normal TOE-supported flows. The expected result is that the factor can be used for authentication but cannot be viewed, exported, or recovered in reusable form. • **Test preparation:** Configure one selected knowledge factor from VE.Auth12.01 for a test User. Identify the normal TOE-supported flows related to authentication settings, backup, export, logs, diagnostics, and account/security management. • **Testing steps:** The tester records the TOE version, platform, selected knowledge-factor type, tested supported flows, authentication state and prohibited disclosure categories in the test evidence. The tester navigates through the identified flows as an authenticated User and, where applicable, as an unauthenticated User, and observes whether the knowledge-factor value or reusable equivalent is displayed, exported, logged, recovered or otherwise made available. • **Test result:** The actual result is the observed confidentiality behaviour: no knowledge-factor value or reusable equivalent is disclosed, only safe masking or non-reusable status information is shown, the factor or reusable equivalent is displayed/exported/logged/recovered, or the disclosure state is ambiguous. The result shall identify the tested flow categories and disclosure category without capturing the actual secret value in screenshots or logs. • **Assessment (pass/fail):** PASS if no knowledge factor value or reusable equivalent is disclosed through the tested supported flows. FAIL if the TOE exposes the factor or a reusable equivalent in clear or directly usable form.
TE.Auth12.02	• **Test description with test expectation:** The stored knowledge factor state shall remain correct after normal TOE state changes defined for the evaluated configuration. The expected result is that the valid factor is accepted and an invalid factor is rejected after the selected state change. • **Test preparation:** Use the TOE instance with the selected knowledge factor configured. Select one normal persistence state change from VE.Auth12.02, such as application restart, device restart, or lock/unlock cycle. • **Testing steps:** The tester records the TOE version, platform, selected knowledge-factor type, selected state change, invalid-factor attempt, valid-factor attempt and observation method in the test evidence. The tester performs the selected state change and then attempts authentication first with an invalid factor and then with the valid factor, unless a lockout state requires handling under Auth14. • **Test result:** The actual result is the observed stored-state integrity behaviour: the invalid factor is rejected and the valid factor is accepted, the valid factor is rejected without documented reason, the invalid factor is accepted, the stored factor state is lost/corrupted/confused with another factor, or the outcome is ambiguous. The result shall describe only access-decision categories and shall not record the actual knowledge-factor value. • **Assessment (pass/fail):** PASS if the invalid factor is rejected and the valid factor is accepted after the selected state change. FAIL if the stored factor state is lost, corrupted, confused with another factor, or accepts an invalid factor.
TE.Auth12.03	• **Test description with test expectation:** The TOE shall not accept a knowledge-factor storage state that fails the vendor-defined conformance-level integrity check. The expected result is safe rejection, reset through a protected procedure, or another documented safe state. • **Test preparation:** Prepare the vendor-supported negative storage-integrity scenario from VE.Auth12.02. Ensure that the method does not require extraction of the actual factor value or uncontrolled tampering outside the agreed test setup. • **Testing steps:** The tester records the TOE version, platform, selected knowledge-factor type, negative storage-integrity scenario, expected safe handling path and observation method in the test evidence. The tester applies the negative scenario, attempts to use the affected knowledge factor for

Element	Content
	authentication or factor management, and observes whether the TOE accepts the affected state or enters a documented safe handling path. • **Test result:** The actual result is the observed negative storage-integrity handling: the TOE rejects the affected state, requires reset through a protected procedure, enters another documented safe state, accepts the affected state as valid, allows unauthorized authentication/factor management, or outcome is ambiguous. The result shall describe TOE-side status, authentication/factor-management result and any safe recovery instruction without recording secrets or reusable authentication artefacts. • **Assessment (pass/fail):** PASS if the TOE rejects the affected state or handles it through a documented safe path without accepting an invalid factor. FAIL if the TOE accepts the affected state as valid or allows unauthorized authentication/factor management.
Test Verdict	PASS iff TE.Auth12.01, TE.Auth12.02 and TE.Auth12.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the knowledge factors, their storage responsibility, or the supported integrity-check scenario.

Table — VE and TE of AS.Auth13 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth13	**The input of the knowledge factor shall be protected against eavesdropping or manipulation.**
Required vendor information	
VE.Auth13.01	The vendor shall identify the knowledge-factor input flows in the evaluated configuration and describe the expected input-protection behaviour for each flow, including masking, echo prevention, supported paste/autofill/accessibility behaviour and any normal TOE-supported observation paths.
VE.Auth13.02	The vendor shall provide SUG/test instructions, test accounts, selected knowledge factors, representative input flows and vendor-supported manipulation variants allowing the tester to verify input protection without recording the actual factor value.
Required test procedures	
TE.Auth13.01	• **Test description with test expectation:** The TOE shall protect the selected knowledge-factor input against eavesdropping through normal TOE-supported observation paths. The expected result is that the factor value or reusable equivalent is not displayed, echoed, exported, or otherwise made available beyond the vendor-defined safe masking behaviour. • **Test preparation:** Configure one selected knowledge factor for a test User and identify one representative input flow from VE.Auth13.01. Prepare evidence capture in a way that does not record the actual factor value. • **Testing steps:** The tester records the TOE version, platform, selected knowledge-factor type, selected input flow, expected masking/protection behaviour and observation method in the test evidence. The tester starts the selected input flow, enters the valid knowledge factor, and observes the input screen, TOE messages and normal supported evidence or diagnostic outputs related to that flow. • **Test result:** The actual result is the observed input-confidentiality behaviour: input is masked or otherwise protected as specified, no reusable equivalent is disclosed, the factor is echoed/displayed/exported/logged, or the disclosure state is ambiguous. The result shall identify the input flow and protection category without recording the actual factor value; if unexpected disclosure occurs, screenshots or logs shall avoid capturing the secret value. • **Assessment (pass/fail):** PASS if no knowledge-factor value or reusable equivalent is disclosed through the tested normal TOE-supported paths. FAIL if the TOE exposes the factor in clear or directly reusable form.
TE.Auth13.02	• **Test description with test expectation:** The TOE shall not accept manipulated knowledge-factor input as valid. The expected result is safe rejection for the manipulated input and normal acceptance for the valid input in the unchanged authorised state. • **Test preparation:** Use the selected input flow from TE.Auth13.01 and prepare one vendor-defined representative manipulation variant. Ensure that the valid factor is known to the tester but is not recorded in the report.

Element	Content
	• **Testing steps:** The tester records the TOE version, platform, selected input flow, manipulation variant, expected rejection outcome and positive-control condition in the test evidence. The tester attempts the selected input flow using the manipulated or invalid input, observes the TOE decision and state change, and then repeats the same flow using the valid input unless the TOE has entered a lockout state that must be handled under Auth14. • **Test result:** The actual result is the observed manipulation-handling behaviour: manipulated input is rejected, fails safely, changes no authentication state, is accepted as valid, changes the factor/state, or causes an ambiguous outcome. The result shall also state whether the valid input is accepted in the unchanged authorised state; the actual factor value shall not be recorded. • **Assessment (pass/fail):** PASS if the manipulated input is rejected or fails safely and the valid input is accepted only in the authorised state. FAIL if manipulated input is accepted, changes the factor, or grants access as if it were valid.
Test Verdict	PASS iff TE.Auth13.01 and TE.Auth13.02 both PASS. FAIL if any TE fails or if VE evidence is insufficient to identify knowledge-factor input flows or the expected input-protection behaviour.

Table — VE and TE of AS.Auth14 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth14	**The Wallet Instance shall implement measures against brute-force attacks.**
Required vendor information	
VE.Auth14.01	The vendor shall specify the brute-force protection measures for each relevant authentication flow, including the applicable threshold, counter scope, mitigation effect, reset or recovery conditions, and observable User/TOE outcomes.
VE.Auth14.02	The vendor shall provide SUG/test instructions, test accounts and test reset/recovery instructions allowing the tester to trigger and observe the brute-force protection without revealing the valid factor. If shortened test thresholds are used, the vendor shall justify equivalence to the evaluated mechanism.
Required test procedures	
TE.Auth14.01	• **Test description with test expectation:** Repeated failed authentication attempts shall trigger the vendor-specified brute-force protection measure. The expected result is that the TOE limits further guessing according to the specified threshold and mitigation behaviour. • **Test preparation:** Configure one test User with a valid authentication factor and determine the applicable threshold and mitigation effect from VE.Auth14.01. Prepare incorrect input values that do not reveal or reuse the valid factor. • **Testing steps:** The tester records the TOE version, platform, selected authentication flow, declared threshold, counter scope, mitigation effect and observation method in the test evidence. The tester attempts authentication repeatedly with incorrect values until the specified threshold or mitigation condition is reached and observes whether the TOE applies the declared brute-force protection measure. • **Test result:** The actual result is the observed brute-force-threshold behaviour: mitigation activates at the specified threshold, activates earlier with documented reason, does not activate, allows continued guessing, or produces an ambiguous state. The result shall describe the number/category of failed attempts, observed mitigation and TOE/User-visible status without recording the valid authentication factor. • **Assessment (pass/fail):** PASS if the declared brute-force protection measure activates as specified and limits further guessing. FAIL if repeated failed attempts can continue without the specified mitigation.
TE.Auth14.02	• **Test description with test expectation:** Once brute-force mitigation is active, the TOE shall maintain or clear that mitigation only according to the vendor-specified policy. The expected result is that normal state changes do not bypass the measure, and authorised recovery restores normal authentication as specified. • **Test preparation:** Use the TOE state after mitigation activation from TE.Auth14.01. Select one normal state change and the vendor-defined recovery or reset condition from VE.Auth14.01.

Element	Content
	• **Testing steps:** The tester records the TOE version, platform, selected normal state change, active mitigation state, recovery condition and expected authentication decisions in the test evidence. The tester performs the selected normal state change and attempts authentication again according to the mitigation policy, then performs the vendor-defined recovery or wait condition and attempts valid authentication. • **Test result:** The actual result is the observed mitigation-persistence and recovery behaviour: mitigation remains active after the selected state change, mitigation is cleared as specified, mitigation is bypassed, recovery restores normal authentication, recovery fails without documented reason, or the state is ambiguous. The result shall describe the authentication decision after the state change and after recovery without recording the valid factor value. • **Assessment (pass/fail):** PASS if the mitigation is not bypassed by the selected normal state change and normal authentication is restored only according to the specified recovery policy. FAIL if the mitigation is cleared or bypassed contrary to the specification, or if recovery cannot be completed without documented reason.
Test Verdict	PASS iff TE.Auth14.01 and TE.Auth14.02 both PASS. FAIL if any TE fails or if VE evidence is insufficient to determine the brute-force threshold, mitigation effect, counter scope, or recovery rule.

Table — VE and TE of AS.Auth15 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth15	**The Wallet Instance ensures that the possession factor is used in a secure manner.**
Required vendor information	
VE.Auth15.01	The vendor shall identify the possession factor used by the Wallet Instance in the evaluated configuration and describe the security-relevant operations that require it. The vendor shall define the observable evidence that the possession factor was used successfully.
VE.Auth15.02	The vendor shall provide SUG/test instructions, test accounts/devices, and conformance-level negative test support allowing the tester to perform a valid use, an unavailable/unbound possession-factor case, and one supported misuse or duplication case.
Required test procedures	
TE.Auth15.01	• **Test description with test expectation:** The Wallet Instance shall use the declared possession factor for the selected security-relevant operation. The expected result is successful completion only through the vendor-specified possession-factor flow. • **Test preparation:** Prepare one registered TOE instance, test User, and selected operation from VE.Auth15.01. Prepare non-secret evidence showing the use of the possession factor, such as TOE status, test endpoint result, or protocol/session metadata. • **Testing steps:** The tester records the TOE version, platform, selected operation, possession-factor type/category, expected possession-factor evidence and observation method in the test evidence. The tester authenticates the User if required, executes the selected operation through the normal evaluated flow, and observes whether the operation completes with the declared possession-factor use. • **Test result:** The actual result is the observed valid possession-factor behaviour: operation completed through the declared possession-factor flow, operation blocked with documented reason, operation completed without observable possession-factor use, or possession-factor evidence ambiguous. The result shall describe operation and evidence categories without recording secret material, private keys or reusable authentication artefacts. • **Assessment (pass/fail):** PASS if the selected operation succeeds only through the specified possession-factor flow. FAIL if the operation succeeds without observable use of the declared possession factor.
TE.Auth15.02	• **Test description with test expectation:** The selected operation shall not complete when the required possession factor is unavailable, unbound, or not established for the TOE instance. The expected result is safe denial or a vendor-defined recovery/binding requirement before execution.

Element	Content
	• **Test preparation:** Prepare a second fresh TOE instance or put the tested TOE into a vendor-supported state where the possession factor is not available or not bound. Use the same selected operation as in TE.Auth15.01 where possible. • **Testing steps:** The tester records the TOE version, platform, selected operation, unavailable/unbound possession-factor state, expected denial or recovery behaviour and observation method in the test evidence. The tester attempts the selected operation without completing the required possession-factor setup or with the possession factor unavailable according to VE.Auth15.02, and observes the TOE-side and test endpoint result. • **Test result:** The actual result is the observed unavailable/unbound possession-factor behaviour: operation denied, recovery/binding required, safe error shown, operation completed as if the possession factor were valid, or outcome ambiguous. The result shall state whether the selected operation completed and shall not include exploit development or technical extraction evidence. • **Assessment (pass/fail):** PASS if the operation is denied or safely redirected to the required binding/recovery procedure. FAIL if the operation completes as if the possession factor were validly present.
TE.Auth15.03	• **Test description with test expectation:** A possession-factor artefact or state shall not be accepted when reused or duplicated through the vendor-supported conformance-level negative method. The expected result is rejection and no completed security-relevant operation. • **Test preparation:** Prepare the registered TOE from TE.Auth15.01, a second TOE instance, and the vendor-supported negative method from VE.Auth15.02. The method shall not require extraction of secret material outside the agreed test setup. • **Testing steps:** The tester records the TOE version, platform, negative misuse/duplication scenario, involved TOE instances, selected operation and expected rejection outcome in the test evidence. The tester executes the supported misuse or duplication scenario and attempts the selected operation using the resulting artefact or state. • **Test result:** The actual result is the observed misuse/duplication behaviour: duplicated or reused artefact/state rejected, operation not completed, safe recovery required, artefact/state accepted, operation completed, or outcome ambiguous. The result shall identify the scenario and acceptance decision without recording secret values or reusable authentication material. • **Assessment (pass/fail):** PASS if the duplicated/reused artefact or state is rejected and the operation is not completed. FAIL if the operation is accepted using a duplicated, replayed, or otherwise unsupported possession-factor state.
Test Verdict	PASS iff TE.Auth15.01, TE.Auth15.02 and TE.Auth15.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the possession factor, the operation requiring it, or the supported negative test method.

Table — VE and TE of AS.Auth16 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth16	**The TOE shall establish and maintain unique, authenticated sessions for each security-relevant wallet operation. These sessions must be bound to specific transactions. The TOE shall prevent concurrent or overlapping sessions where such concurrency could compromise authentication, consent, authorization, or replay protection. Application Note: Upon completion, cancellation, logout, inactivity, channel loss, security failure, or significant context change related to a transaction, the TOE shall terminate the associated session and reject attempts to reuse any previously generated confirmation data, nonces, tokens, or session identifiers.**
Required vendor information	
VE.Auth16.01	The vendor shall specify the session model for security-relevant wallet operations, including session creation, authentication, transaction binding, termination events, and observable session outcomes. The vendor shall define how the tester can observe session uniqueness and transaction binding without exposing secrets.
VE.Auth16.02	The vendor shall provide SUG/test instructions, test endpoints or test backend support, and test data allowing the tester to create valid sessions, start concurrent or overlapping operations, terminate sessions, and attempt reuse of prior confirmation data or session artefacts.

Element	Content
Required test procedures	
TE.Auth16.01	• **Test description with test expectation:** The TOE shall create an authenticated session for the selected security-relevant operation and bind it to the specific transaction being executed. The expected result is that confirmation or authorization for one transaction is not accepted for a different transaction. • **Test preparation:** Prepare one authenticated test User, a test endpoint/backend, and two distinguishable transactions for the same operation type. Identify the non-secret evidence of session uniqueness and transaction binding from VE.Auth16.01. • **Testing steps:** The tester records the TOE version, platform, operation type, transaction A/B non-secret labels, expected session/confirmation behaviour and observation method in the test evidence. The tester starts transaction A and completes the required authentication/confirmation for transaction A, then starts transaction B and verifies whether transaction B requires its own valid session/confirmation instead of reusing the confirmation result from transaction A. • **Test result:** The actual result is the observed transaction-binding behaviour: each transaction uses its own authenticated session, transaction B requires its own confirmation, transaction A confirmation is rejected for transaction B, one session/confirmation authorizes another transaction, or the binding state is ambiguous. The result shall describe only non-secret transaction/session categories and endpoint/backend decisions. • **Assessment (pass/fail):** PASS if each transaction is handled under its own authenticated session and confirmation for one transaction is not accepted for another. FAIL if one session or confirmation authorizes a different transaction without a vendor-defined safe reason.
TE.Auth16.02	• **Test description with test expectation:** The TOE shall prevent or safely handle concurrent or overlapping sessions for selected security-relevant operations where concurrency could compromise the transaction. The expected result is no cross-authorization, no mixed consent, and no replay-usable session outcome. • **Test preparation:** Select one operation pair or repeated operation from VE.Auth16.02 where overlapping sessions are relevant. Prepare the TOE and test endpoint/backend so that two operations can be started before the first one is completed, where supported by the UI/protocol. • **Testing steps:** The tester records the TOE version, platform, selected operation pair, overlap condition, expected safe handling and observation method in the test evidence. The tester starts operation/session A and leaves it pending, starts operation/session B or a second instance of the same operation, then attempts to confirm or complete one or both operations according to the vendor-supported flow. • **Test result:** The actual result is the observed overlapping-session behaviour: second session blocked, sessions safely separated, one session invalidated, both operations completed with correct independent consent, operation completed with wrong session/consent, replay-usable session outcome produced, or outcome ambiguous. The result shall describe TOE-side handling and endpoint/backend decisions using only non-secret session evidence. • **Assessment (pass/fail):** PASS if the TOE blocks overlap or handles it safely without cross-authorization or replay effect. FAIL if overlapping sessions allow one confirmation, token, nonce, or session outcome to complete the wrong operation.
TE.Auth16.03	• **Test description with test expectation:** After a vendor-defined session termination event, the TOE shall terminate the associated session and reject reuse of previously generated confirmation data, nonces, tokens, or session identifiers. The expected result is safe rejection and no completed operation from the old session. • **Test preparation:** Prepare a selected security-relevant operation and at least one termination event from VE.Auth16.01, such as completion, cancellation, logout, inactivity, channel loss, security failure, or significant transaction context change. Prepare non-secret observation support for reuse rejection. • **Testing steps:** The tester records the TOE version, platform, selected operation, termination event, expected reuse-rejection behaviour and observation method in the test evidence. The tester starts the selected operation and reaches the relevant session state or confirmation step without recording secrets, triggers the selected termination event, and then attempts to continue or reuse the prior session/confirmation artefact according to the vendor-supported negative method. • **Test result:** The actual result is the observed termination and reuse-rejection behaviour: session terminated, continuation blocked, prior artefact rejected, safe error shown, old artefact accepted, operation completed from old session, or outcome ambiguous. The result shall identify the

Element	Content
	termination event and acceptance decision without recording nonce, token, session identifier or confirmation-data values. • **Assessment (pass/fail):** PASS if the terminated session cannot be continued or reused and the old artefact is rejected. FAIL if the TOE or endpoint accepts previously generated session artefacts after the termination event.
Test Verdict	PASS iff TE.Auth16.01, TE.Auth16.02 and TE.Auth16.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the session model, transaction-binding evidence, concurrency handling, or termination/reuse-rejection method.

Table — VE and TE of AS.Auth17 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth17	**TOE shall provide a logically isolated trusted user interaction path for all sensitive operations, including user authentication, re-authentication, transaction consent, credential disclosure authorization, credential capture or generation, presentation and signature-initiation operations. Application Note: The trusted path shall display User the verified operation context and shall protect the user decision against overlay, UI spoofing, unauthorized screen capture, modification by and disclosure to/by untrusted applications to the extent enforceable by the supported platform.**
Required vendor information	
VE.Auth17.01	The vendor shall identify all sensitive operations requiring the trusted user interaction path and define the trusted-path behaviour for each operation class. The vendor shall specify the verified operation context shown to the User and the observable evidence that the User decision was taken through the trusted path.
VE.Auth17.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints and platform-specific test support allowing the tester to trigger trusted-path flows, observe context binding, and perform conformance-level negative checks for context mismatch and untrusted UI/capture interaction.
Required test procedures	
TE.Auth17.01	• **Test description with test expectation:** For each selected sensitive-operation class, the TOE shall use the declared trusted user interaction path and display the verified operation context before accepting the User decision. The expected result is that the operation proceeds only after the User decision is taken through that trusted path. • **Test preparation:** Install the TOE in the evaluated secure configuration and prepare one authenticated test User. Select representative sensitive operations from VE.Auth17.01, covering the mandatory operation classes where applicable to the evaluated TOE. • **Testing steps:** The tester records the TOE version, platform, selected sensitive-operation class, expected verified operation context, defined User decision action and observation method in the test evidence. The tester initiates each selected sensitive operation, observes the trusted-path step, confirms the operation using the vendor-defined User action and observes whether the operation is executed afterwards. • **Test result:** The actual result is the observed trusted-path behaviour: trusted path shown with verified context, User decision action available, operation executed only after the trusted-path decision, operation executed without trusted path, operation executed before verified context is shown, or outcome ambiguous. The result shall describe the operation class, visible context category and execution outcome without recording secrets, authentication factors, private keys or unnecessary personal data. • **Assessment (pass/fail):** PASS if every tested operation uses the declared trusted path, displays the verified context before decision, and executes only after the User decision. FAIL if any tested operation executes without the trusted path or before the verified context is shown.
TE.Auth17.02	• **Test description with test expectation:** The TOE shall not complete a sensitive operation if the operation context accepted by the User does not match the verified context bound to the operation. The expected result is safe rejection, re-prompting with the corrected context, or another documented safe handling path.

Element	Content
	• **Test preparation:** Prepare one selected sensitive operation and the vendor-supported context-mismatch scenario from VE.Auth17.02. Ensure that the tester can distinguish the context shown to the User from the context requested or completed at the test endpoint. • **Testing steps:** The tester records the TOE version, platform, selected sensitive operation, displayed context category, attempted execution context category, mismatch scenario and expected safe handling path in the test evidence. The tester initiates the selected operation, creates the vendor-defined mismatch, attempts to confirm or complete the operation and observes the TOE UI and endpoint/backend result. • **Test result:** The actual result is the observed context-mismatch behaviour: operation rejected, corrected context re-prompted, another documented safe path used, mismatched operation completed, User decision for one context accepted for another, or outcome ambiguous. The result shall describe TOE-side handling and endpoint/backend decision without recording secret values or reusable transaction artefacts. • **Assessment (pass/fail):** PASS if the TOE does not complete the operation under a mismatched or unverified context. FAIL if the User decision for one context authorizes a different context.
TE.Auth17.03	• **Test description with test expectation:** The trusted path shall apply the vendor-specified platform-enforceable protections against untrusted UI/capture interaction. The expected result is blocking, masking, warning, safe denial, or another documented safe behaviour before the User decision is exposed or modified. • **Test preparation:** Prepare the selected trusted-path flow and the vendor-supported untrusted UI/capture conformance scenario from VE.Auth17.02. Use only agreed platform features or test applications, without exploit development. • **Testing steps:** The tester records the TOE version, platform, selected trusted-path flow, selected untrusted UI/capture scenario, declared platform-enforceable protection and expected safe behaviour in the test evidence. The tester starts the trusted-path flow, attempts the selected untrusted interaction during the User decision step and observes whether the TOE applies the declared protection or enters the documented safe handling path. • **Test result:** The actual result is the observed untrusted-interaction behaviour: interaction blocked, screen masked, warning displayed, safe denial applied, documented safe handling used, User decision/context exposed, User decision/context modified, UI spoofing/capture succeeds, or outcome ambiguous. The result shall state whether the User decision or operation context was exposed or modified. • **Assessment (pass/fail):** PASS if the TOE applies the declared platform-enforceable protection or safe handling and the User decision is not exposed or modified through the tested scenario. FAIL if the tested untrusted interaction can view, alter, spoof, or capture the User decision contrary to the specification.
Test Verdict	PASS iff TE.Auth17.01, TE.Auth17.02 and TE.Auth17.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify sensitive operations, trusted-path behaviour, verified context, or platform-enforceable protections.

Table — VE and TE of AS.Auth18 (FITCEM PP: Authentication and authorization)

Element	Content
Authentication and authorization — AS.Auth18	**The TOE shall define and enforce the lifecycle of wallet user accounts, wallet instance identities, device bindings including identity proofing or equivalent evidence validation before activation or high-assurance use. Application Note: The TOE requires User to provide new knowledge factors if it is reinstalled or installed on another mobile device.**
Required vendor information	
VE.Auth18.01	The vendor shall specify the lifecycle of wallet user accounts, wallet instance identities and device bindings in the evaluated configuration, including lifecycle states, activation criteria, high-assurance use criteria, identity proofing or equivalent evidence validation, device-binding evidence and observable TOE/backend outcomes.
VE.Auth18.02	The vendor shall provide SUG/test instructions, test accounts, test devices or equivalent test instances, and test backend support allowing the tester to observe valid activation, invalid or incomplete lifecycle

Element	Content
	states, high-assurance use, device binding, reinstall behaviour and another-device behaviour without exposing secrets.
Required test procedures	
TE.Auth18.01	• **Test description with test expectation:** The TOE shall complete activation and permit selected high-assurance use only after the required lifecycle conditions, including identity proofing or equivalent evidence validation and device binding, are satisfied. The expected result is that activation/high-assurance use becomes available only in the valid lifecycle state. • **Test preparation:** Prepare a fresh TOE instance, test User, test backend or equivalent lifecycle test environment, successful validation condition, device-binding flow and one selected high-assurance operation from VE.Auth18.01. • **Testing steps:** The tester records the TOE version, platform, lifecycle state, validation condition, device-binding evidence category, selected high-assurance operation and observation method in the test evidence. The tester completes the vendor-defined activation flow, including required identity proofing or equivalent evidence validation and device binding, then attempts the selected high-assurance operation and observes the TOE and backend/test-backend decision. • **Test result:** The actual result is the observed valid-lifecycle behaviour: validation completed, device binding established, activation completed, selected high-assurance operation available, operation blocked with documented reason, lifecycle state ambiguous, or activation/use occurred without required conditions. The result shall describe lifecycle and backend/test-backend decisions without recording identity-proofing secrets, knowledge factors, private keys or unnecessary personal data. • **Assessment (pass/fail):** PASS if activation and high-assurance use occur only after the required validation and device binding are completed. FAIL if activation or high-assurance use succeeds without the specified lifecycle conditions.
TE.Auth18.02	• **Test description with test expectation:** The TOE shall block activation-dependent or high-assurance use when the wallet user account, wallet instance identity or device binding is not yet valid. The expected result is safe denial, activation requirement, or evidence-validation requirement. • **Test preparation:** Prepare a fresh, inactive, unbound or evidence-validation-failed TOE state according to VE.Auth18.02. Use the same selected high-assurance operation as in TE.Auth18.01 where possible. • **Testing steps:** The tester records the TOE version, platform, selected invalid lifecycle state, validation condition, selected high-assurance operation and expected blocking behaviour in the test evidence. The tester attempts the selected operation before completing activation or under the vendor-defined invalid evidence-validation/device-binding condition and observes the TOE UI and backend/test-backend decision. • **Test result:** The actual result is the observed invalid-lifecycle behaviour: operation blocked, activation/evidence-validation requirement shown, device-binding requirement shown, safe denial returned, operation succeeds despite invalid lifecycle state, or outcome ambiguous. The result shall describe TOE-side status and backend/test-backend decision; do not use exploit development or unauthorized backend manipulation. • **Assessment (pass/fail):** PASS if the TOE blocks the operation and requires the correct lifecycle step before use. FAIL if the operation succeeds while the account, instance identity, validation condition or device binding is not valid.
TE.Auth18.03	• **Test description with test expectation:** After reinstalling the TOE or installing it on another mobile device, the TOE shall require the User to provide new knowledge factors before activation or high-assurance use. The expected result is that previous local knowledge-factor state is not silently reused as sufficient for the new lifecycle state. • **Test preparation:** Prepare an already activated TOE instance with a configured knowledge factor and device binding. Prepare a reinstall scenario or a second device scenario according to VE.Auth18.02. • **Testing steps:** The tester records the TOE version, platform, scenario type, previous lifecycle/device-binding state, post-reinstall or second-device state, expected new-knowledge-factor requirement and selected activation/high-assurance operation in the test evidence. The tester performs the reinstall or another-device scenario, attempts activation or selected high-assurance use without creating new knowledge factors, completes the vendor-defined new knowledge-factor setup and repeats the operation. • **Test result:** The actual result is the observed reinstall/another-device lifecycle behaviour: new knowledge factors required, activation/high-assurance use blocked before new factor setup, operation succeeds only after new factor setup, previous local factor state silently reused, high-assurance use

Element	Content
	permitted without new factor setup, or outcome ambiguous. The result shall describe operation result before and after new factor setup without recording actual knowledge-factor values. • **Assessment (pass/fail):** PASS if the TOE requires new knowledge factors and blocks activation/high-assurance use until the new factor setup and required lifecycle steps are completed. FAIL if the TOE silently reuses previous knowledge-factor state or permits high-assurance use without the required new factor setup.
Test Verdict	PASS iff TE.Auth18.01, TE.Auth18.02 and TE.Auth18.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify lifecycle states, activation criteria, high-assurance use, device-binding evidence, or reinstall/another-device behaviour.

LOG — Logging

Table — VE and TE of AS.Log1 (FITCEM PP: Logging)

Element	Content
Logging — AS.Log1	**The TOE shall log wallet activation credential management, and presentation processes. However, specific attributes presented during these processes may not be recorded in the log. This log shall be presented to and made accessible by the respective authenticated users through the TOE's user interface.**
Required vendor information	
VE.Log1.01	The vendor shall identify the activation, credential-management and presentation processes that are logged in the evaluated configuration. The vendor shall specify the log-entry content and identify which presented attribute values or specific attributes are prohibited from being recorded.
VE.Log1.02	The vendor shall provide SUG/test instructions, test accounts, test credentials or attestations, and test endpoints allowing the tester to trigger the logged processes and access the log through the TOE user interface. The SUG should match the actual UI used by the tester, including labels and steps.
Required test procedures	
TE.Log1.01	• **Test description with test expectation:** The TOE shall create user-visible log entries for the selected activation, credential-management and presentation processes. The expected result is that the respective authenticated User can access the corresponding log entries through the TOE user interface. • **Test preparation:** Prepare a TOE instance, one authenticated test User, and the required test backend/endpoints according to VE.Log1.02. Select one representative event for each applicable logged process class from VE.Log1.01. • **Testing steps:** The tester records the TOE version, platform, test User, selected process classes and selected events in the test evidence. The tester executes the selected events, authenticates as the respective User, opens the TOE log UI, and observes the entries displayed for those events. • **Test result:** The actual result is the observed UI behaviour: the log screen either displays corresponding entries for the executed events, displays incomplete entries, or displays no corresponding entries. The result shall identify which selected process classes were visible in the log UI. • **Assessment (pass/fail):** PASS if each tested applicable process class produces a corresponding log entry accessible to the respective authenticated User through the TOE UI. FAIL if an applicable tested process is not logged or the authenticated User cannot access the log through the TOE UI.
TE.Log1.02	• **Test description with test expectation:** The TOE shall not record specific attributes presented during the selected presentation process. The expected result is that the log may show the occurrence and result of the presentation process, but not the presented attribute values prohibited by VE.Log1.01. • **Test preparation:** Prepare a presentation scenario using known test attributes and a test Relying Party or test endpoint. Identify the prohibited attribute values/categories from VE.Log1.01 before executing the test. • **Testing steps:** The tester records the TOE version, platform, selected presentation scenario and the prohibited log-content categories in the test evidence, without recording real personal data. The tester performs the presentation process, opens the TOE log UI and any supported user-accessible log view/export, and observes whether the prohibited specific attributes appear. • **Test result:** The actual result is the observed log content: the UI/export either omits the prohibited specific attributes, shows only allowed generic event information, or exposes prohibited presented attribute information. The result shall describe the observed categories, not copy secret or personal values. • **Assessment (pass/fail):** PASS if the tested log view does not contain prohibited specific attributes or their reusable equivalents. FAIL if the TOE records or displays prohibited specific presented attributes in the log.
Test Verdict	PASS iff TE.Log1.01 and TE.Log1.02 both PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the logged process classes, expected log entries, or prohibited attribute content.

Table — VE and TE of AS.Log2 (FITCEM PP: Logging)

Element	Content
Logging — AS.Log2	**The TOE may only grant the authorized user access to the log files. Log data shall be protected against unauthorized deletion, tampering, and unauthorized access.Application Note: If national or EU regulations contain applicable provisions regarding the retention period for log data, these shall be observed.**
Required vendor information	
VE.Log2.01	The vendor shall define the authorized user for log access and identify all TOE-supported log access paths in the evaluated configuration. The vendor shall also specify whether log access is UI-only, export-based, API-based, diagnostic, or otherwise supported.
VE.Log2.02	The vendor shall define the conformance-level protection and retention behaviour for log data, including supported deletion, retention, recovery or reset cases. The vendor shall provide test instructions for unauthorized access, unauthorized deletion/tampering checks, and retention-policy observation without requiring storage forensics.
Required test procedures	
TE.Log2.01	• **Test description with test expectation:** The TOE shall grant access to log data only to the authorized User. The expected result is that the authorized User can view the log, while unauthenticated or unauthorized access is denied or redirected to authentication. • **Test preparation:** Prepare a TOE instance with at least one existing log entry and one authorized test User. Prepare an unauthenticated state and, where supported, a second User or unauthorized test state. • **Testing steps:** The tester records the TOE version, platform, selected log access path and tested user states in the test evidence. The tester accesses the log as the authorized authenticated User, then repeats the access attempt from the unauthenticated or unauthorized state and observes the UI or supported access path response. • **Test result:** The actual result is the observed application behaviour: the authorized User sees the log, while the unauthenticated or unauthorized state receives an authentication prompt, denial, or documented safe response. The result shall identify the visible access decision for each tested state. • **Assessment (pass/fail):** PASS if log access is granted only to the authorized authenticated User and denied or safely gated otherwise. FAIL if an unauthenticated or unauthorized state can view log data.
TE.Log2.02	• **Test description with test expectation:** The TOE shall prevent unauthorized deletion or tampering of log data through TOE-supported log management paths. The expected result is denial, absence of unauthorized controls, or another documented safe response, with the original log entry preserved. • **Test preparation:** Prepare a known existing log entry created during Log1 testing and identify the supported log-management paths from VE.Log2.02. Prepare an unauthorized state or user for the negative attempt. • **Testing steps:** The tester records the TOE version, platform, selected log entry, selected management path and unauthorized state in the test evidence. The tester attempts the vendor-defined unauthorized deletion/tampering scenario and then accesses the log as the authorized User to observe the state of the selected entry. • **Test result:** The actual result is the observed application behaviour: the unauthorized attempt is denied, unavailable, or safely handled, and the selected log entry remains present and unchanged. If the TOE provides a message, the result shall describe the message category without copying sensitive log content. • **Assessment (pass/fail):** PASS if unauthorized deletion/tampering is prevented and the selected log entry remains protected. FAIL if an unauthorized state can delete, alter, hide, or corrupt the selected log entry through supported TOE paths.
TE.Log2.03	• **Test description with test expectation:** The TOE shall apply the defined log-retention behaviour where a national or EU retention period applies to the evaluated configuration. The expected result is that log data is retained or removed only according to the applicable documented retention rule. • **Test preparation:** Identify the applicable retention rule or confirm that no specific retention rule applies in the evaluated configuration. Prepare a log entry and a vendor-supported retention observation method, such as a test retention interval or controlled retention-state transition.

Element	Content
	• **Testing steps:** The tester records the TOE version, platform, selected log entry, applicable retention rule and retention observation method in the test evidence. The tester observes the log before and after the vendor-supported retention condition and checks the resulting log availability through the authorized log access path. • **Test result:** The actual result is the observed retention behaviour: the log entry remains available before expiry, is removed only after the documented retention condition, or shows a documented "not applicable" outcome. The result shall describe what the authorized User sees in the TOE UI or supported log access path. • **Assessment (pass/fail):** PASS if the observed retention behaviour matches the documented applicable retention rule or the documented non-applicability decision. FAIL if log data is removed too early, retained contrary to an applicable rule, or behaves inconsistently with the documented retention behaviour.
Test Verdict	PASS iff TE.Log2.01, TE.Log2.02 and TE.Log2.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify authorized users, log access paths, deletion/tampering behaviour, or applicable retention rules.

Table — VE and TE of AS.Log3 (FITCEM PP: Logging)

Element	Content
Logging — AS.Log3	**For the events listed above, the TOE shall store at least the following information:• timestamp• event type• result of the action that triggered the event.Application Note: log files shall not contain authentication credentials or sensitive authentication data.**
Required vendor information	
VE.Log3.01	The vendor shall define the log-entry schema for the events covered by Log1/Log2, including timestamp, event type, and action result. The vendor shall also define timestamp format, event-type taxonomy, result values, and prohibited authentication-related log content.
VE.Log3.02	The vendor shall provide SUG/test instructions, test accounts, test data, and log UI/export access allowing the tester to trigger logged events and observe the required fields without recording secrets. The SUG should match the actual UI labels and steps used by the tester.
Required test procedures	
TE.Log3.01	• **Test description with test expectation:** The TOE shall store timestamp, event type and action result for selected events covered by Log1/Log2. The expected result is that each tested log entry contains these three required information elements with values matching the executed event. • **Test preparation:** Prepare a TOE instance, one authenticated test User, and at least one existing log access path according to VE.Log3.02. Select one successful event and one failed/denied event from the events covered by Log1/Log2. • **Testing steps:** The tester records the TOE version, platform, test User, selected events, and selected log access path in the test evidence. The tester executes the selected events, opens the TOE log UI or supported log export, and observes the corresponding log entries. • **Test result:** The actual result is the observed log-entry content: each tested entry either displays timestamp, event type and action result matching the event, displays incomplete fields, or displays no corresponding entry. The result shall describe the visible field categories and whether the shown result matches the action outcome. • **Assessment (pass/fail):** PASS if each tested event has a corresponding log entry with timestamp, event type and correct action result. FAIL if any required field is missing, misleading, or inconsistent with the executed event.
TE.Log3.02	• **Test description with test expectation:** Log files shall not contain authentication credentials or sensitive authentication data. The expected result is that tested log entries show allowed event information only, without revealing credentials, authentication factors, tokens, secrets, or reusable authentication artefacts. • **Test preparation:** Prepare events involving User authentication or authentication-dependent wallet actions using test-only credentials/factors. Identify prohibited authentication-related content categories from VE.Log3.01 before executing the test.

Element	Content
	• **Testing steps:** The tester records the TOE version, platform, selected authentication-related events, and prohibited content categories in the test evidence, without recording actual secret values. The tester executes the selected events and observes the TOE log UI and any supported user-accessible log export. • **Test result:** The actual result is the observed log content: the log either omits prohibited authentication data, shows only allowed generic authentication/event information, or exposes prohibited authentication-related content. The result shall describe categories of observed content and shall not copy secret values. • **Assessment (pass/fail):** PASS if no authentication credentials, sensitive authentication data, or reusable authentication artefacts are visible in the tested log views. FAIL if such data is recorded or displayed in the log.
Test Verdict	PASS iff TE.Log3.01 and TE.Log3.02 both PASS. FAIL if any TE fails or if VE evidence is insufficient to identify required log fields, event taxonomy, result values, or prohibited authentication-related content.

Table — VE and TE of AS.Log4 (FITCEM PP: Logging)

Element	Content
Logging — AS.Log4	**The TOE shall generate audit and evidence records for security-relevant wallet operations, including activation, credential acquisition or generation, presentation, disclosure authorization, consent, signature initiation, attestation handling, backup or restore, trust or certificate validation, and relevant security failures. Each record shall contain sufficient tamper-evident context to later verify what was requested, authorized and performed, by which authenticated user or process, when, with what result, and under which transaction context, while avoiding unnecessary personal data.**
Required vendor information	
VE.Log4.01	The vendor shall identify security-relevant wallet operations covered by Log4 and define the audit/evidence record schema. The schema shall describe the context needed to verify request, authorization, performance, authenticated user or process, time, result, transaction context, and personal-data minimization.
VE.Log4.02	The vendor shall provide test instructions, test accounts, test endpoints or test backend support, and a conformance-level record-verification method. The method shall allow the tester to observe record generation and tamper-evident status without exposing secrets or unnecessary personal data.
Required test procedures	
TE.Log4.01	• **Test description with test expectation:** The TOE shall generate audit/evidence records for selected security-relevant wallet operations. The expected result is that each tested operation produces a record containing the required context categories defined in VE.Log4.01. • **Test preparation:** Prepare a TOE instance, authenticated test User, and test endpoint/backend according to VE.Log4.02. Select representative applicable operation classes from VE.Log4.01, including at least one operation involving User authorization or consent and one operation ending in a security-relevant failure. • **Testing steps:** The tester records the TOE version, platform, selected operation classes, test endpoint/backend identifier, and selected record-observation method in the test evidence. The tester executes the selected operations and observes the generated audit/evidence records through the vendor-defined observation method. • **Test result:** The actual result is the observed record content: each tested operation either has a corresponding record with the required context categories, has an incomplete record, or has no corresponding audit/evidence record. The result shall describe the observed context categories and visible operation outcome, not secret values. • **Assessment (pass/fail):** PASS if each tested security-relevant operation generates an audit/evidence record with sufficient context to verify the request, authorization, performance, actor/process, time, result and transaction context. FAIL if records are missing or do not contain the required context categories.
TE.Log4.02	• **Test description with test expectation:** The TOE or vendor-provided verification method shall indicate whether the tested audit/evidence record remains valid and context-consistent. The expected result is

Element	Content
	acceptance of the original record and safe rejection or invalid indication for the vendor-supported negative record scenario. • **Test preparation:** Use at least one audit/evidence record generated in TE.Log4.01 and prepare the vendor-supported negative verification scenario from VE.Log4.02. The negative scenario shall not require uncontrolled storage tampering or exposure of secret values. • **Testing steps:** The tester records the TOE version, platform, selected record, verification method and negative scenario in the test evidence. The tester verifies the original record, applies the supported negative scenario, and repeats verification through the same method. • **Test result:** The actual result is the observed verification behaviour: the original record is indicated as valid, while the negative record scenario is indicated as invalid, inconsistent, rejected, or otherwise safely handled. The result shall describe the verification status shown by the tool, UI, backend, or test endpoint. • **Assessment (pass/fail):** PASS if the original record verifies as valid and the negative scenario is not accepted as a valid unchanged record. FAIL if the negative scenario is accepted as valid or if no verification outcome can be observed.
TE.Log4.03	• **Test description with test expectation:** Audit/evidence records shall avoid unnecessary personal data while preserving the required verification context. The expected result is that records contain allowed contextual categories but do not contain personal data fields prohibited by VE.Log4.01. • **Test preparation:** Prepare one selected operation using known test personal data or test attributes. Identify before execution which personal-data elements are necessary for evidence context and which are prohibited or unnecessary according to VE.Log4.01. • **Testing steps:** The tester records the TOE version, platform, selected operation, personal-data categories used in the test, and prohibited record-content categories in the test evidence, without recording real personal data. The tester executes the operation and observes the generated audit/evidence record through the approved observation method. • **Test result:** The actual result is the observed record content: the record either contains only the allowed context categories, contains excessive personal-data categories, or lacks required context because of over-minimization. The result shall describe content categories rather than copying personal values. • **Assessment (pass/fail):** PASS if the record contains sufficient verification context and does not include prohibited unnecessary personal-data categories. FAIL if the record contains unnecessary personal data or omits required context needed for later verification.
Test Verdict	PASS iff TE.Log4.01, TE.Log4.02 and TE.Log4.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify security-relevant wallet operations, record schema, tamper-evident verification method, transaction context, or personal-data minimization rules.

Table — VE and TE of AS.Log5 (FITCEM PP: Logging)

Element	Content
Logging — AS.Log5	**The TOE shall detect logging failures, unauthorized local, network or remote connection indicators and security-relevant anomalies that may affect wallet security or accountability.**
Required vendor information	
VE.Log5.01	The vendor shall define the logging-failure, unauthorized-connection-indicator and security-relevant-anomaly cases detected by the TOE in the evaluated configuration. The vendor shall specify the observable detection outcome for each case.
VE.Log5.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints or test hooks allowing the tester to trigger the selected detection cases without exploit development. The method shall allow the tester to observe detection through UI, approved diagnostics, test endpoint or test backend.
Required test procedures	
TE.Log5.01	• **Test description with test expectation:** The TOE shall detect a selected logging failure that may affect wallet security or accountability. The expected result is a visible or otherwise observable detection state defined by the vendor.

Element	Content
	• **Test preparation:** Prepare a TOE instance with logging enabled and at least one security-relevant operation that normally generates a log or evidence record. Prepare the vendor-supported logging-failure scenario from VE.Log5.02. • **Testing steps:** The tester records the TOE version, platform, selected operation, selected logging-failure scenario and observation method in the test evidence. The tester triggers the logging-failure scenario, executes or attempts the selected operation, and observes the TOE UI, approved diagnostic output, test endpoint or test backend. • **Test result:** The actual result is the observed detection behaviour: the TOE either reports a logging failure, enters a documented degraded/safe state, or shows no detection. The result shall describe the visible status, message category, diagnostic state or endpoint/backend signal observed by the tester. • **Assessment (pass/fail):** PASS if the logging failure is detected and produces the vendor-defined observable outcome. FAIL if the logging failure occurs without any observable detection outcome.
TE.Log5.02	• **Test description with test expectation:** The TOE shall detect a selected unauthorized local, network or remote connection indicator. The expected result is that the TOE identifies the indicator before it is treated as a normal trusted interaction. • **Test preparation:** Prepare the TOE and the vendor-supported unauthorized-connection-indicator scenario from VE.Log5.02. Select one representative connection type applicable to the evaluated TOE interface scope. • **Testing steps:** The tester records the TOE version, platform, selected connection type, indicator scenario and observation method in the test evidence. The tester initiates the selected scenario through the approved test setup and observes the TOE UI, approved diagnostic output, test endpoint or test backend. • **Test result:** The actual result is the observed application or endpoint behaviour: the TOE either flags the unauthorized connection indicator, rejects or marks the interaction as unsafe, or treats it as normal. The result shall describe the observed detection status and any user-visible or tester-visible signal. • **Assessment (pass/fail):** PASS if the selected unauthorized connection indicator is detected and not treated as a normal trusted interaction. FAIL if the TOE accepts or ignores the indicator contrary to the vendor-defined detection behaviour.
TE.Log5.03	• **Test description with test expectation:** The TOE shall detect a selected security-relevant anomaly that may affect wallet security or accountability. The expected result is a vendor-defined observable anomaly-detection outcome. • **Test preparation:** Select one representative security-relevant anomaly from VE.Log5.01 and prepare the vendor-supported trigger method. Ensure that the selected anomaly is within conformance testing and does not require exploit development or uncontrolled tampering. • **Testing steps:** The tester records the TOE version, platform, selected anomaly, trigger method and observation method in the test evidence. The tester triggers the selected anomaly and observes the TOE UI, approved diagnostic output, test endpoint or test backend. • **Test result:** The actual result is the observed anomaly handling: the TOE either reports, flags or records the anomaly, enters a documented safe/degraded state, or shows no observable detection. The result shall describe the observed state or message category without copying sensitive log or diagnostic content. • **Assessment (pass/fail):** PASS if the selected security-relevant anomaly is detected and produces the specified observable outcome. FAIL if the anomaly is not detected or is handled as normal operation without documented reason.
Test Verdict	PASS iff TE.Log5.01, TE.Log5.02 and TE.Log5.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify detectable cases, trigger methods or observable detection outcomes.

Table — VE and TE of AS.Log6 (FITCEM PP: Logging)

Element	Content
Logging — AS.Log6	**The TOE shall alert the authorized user, preserve available evidence, block or degrade affected security-relevant operations where mandatory evidence cannot be generated, and support incident handling through defined escalation and recovery information.**

Element	Content
Required vendor information	
VE.Log6.01	The vendor shall define the alerting, evidence-preservation, blocking/degradation, escalation and recovery behaviour for Log5-relevant detected cases. The vendor shall identify which evidence is mandatory for each selected security-relevant operation.
VE.Log6.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints or test backend support allowing the tester to observe alerting, evidence preservation, affected-operation handling and recovery information. The method shall avoid exposing secrets or unnecessary personal data.
Required test procedures	
TE.Log6.01	• **Test description with test expectation:** After a selected Log5-relevant detected case, the TOE shall alert the authorized User and preserve available evidence. The expected result is a user-visible alert or equivalent authorized-user notification and an observable preserved evidence state. • **Test preparation:** Prepare a TOE instance, authorized test User, and one selected detected case from Log5 testing. Identify from VE.Log6.01 the expected alert content category and the evidence-preservation observation method. • **Testing steps:** The tester records the TOE version, platform, selected detected case, expected alert category and evidence-observation method in the test evidence. The tester triggers the selected case, authenticates or remains authenticated as the authorized User where required, and observes the TOE UI plus the approved evidence view, diagnostic output or test backend. • **Test result:** The actual result is the observed reaction: the authorized User either receives an alert and preserved evidence is visible through the approved method, receives only partial information, or receives no alert/evidence indication. The result shall describe the alert category and evidence state without copying secrets or unnecessary personal data. • **Assessment (pass/fail):** PASS if the authorized User is alerted and available evidence is preserved as specified. FAIL if no alert is provided to the authorized User or available evidence is lost/not observable contrary to the specification.
TE.Log6.02	• **Test description with test expectation:** Where mandatory evidence cannot be generated, the TOE shall block or degrade affected security-relevant operations according to the vendor-defined policy. The expected result is that the affected operation is not completed as a normal fully evidenced operation. • **Test preparation:** Select one security-relevant operation with mandatory evidence defined in VE.Log6.01. Prepare the vendor-supported scenario in which mandatory evidence cannot be generated for that operation. • **Testing steps:** The tester records the TOE version, platform, selected operation, mandatory-evidence type/category and evidence-generation-failure scenario in the test evidence. The tester triggers the scenario, attempts the affected operation, and observes the TOE UI, test endpoint or backend result. • **Test result:** The actual result is the observed operation handling: the operation is blocked, degraded with clear indication, safely redirected, or completed normally despite missing mandatory evidence. The result shall describe the visible status and endpoint/backend outcome. • **Assessment (pass/fail):** PASS if the TOE blocks or degrades the affected operation as specified when mandatory evidence cannot be generated. FAIL if the operation completes normally despite missing mandatory evidence.
TE.Log6.03	• **Test description with test expectation:** The TOE shall support incident handling by presenting or making available defined escalation and recovery information after the selected Log6-relevant case. The expected result is that the authorized User can identify the next safe action or recovery path from the TOE-supported information. • **Test preparation:** Use the TOE state after TE.Log6.01 or TE.Log6.02, or prepare an equivalent vendor-supported incident state. Identify the expected escalation and recovery information from VE.Log6.01 and VE.Log6.02. • **Testing steps:** The tester records the TOE version, platform, selected incident state and expected escalation/recovery information category in the test evidence. The tester observes the TOE UI, in-app help, approved support flow or supported recovery screen available to the authorized User. • **Test result:** The actual result is the observed incident-handling support: the TOE provides clear escalation/recovery information, provides incomplete information, or provides no relevant

Element	Content
	information. The result shall describe the information category and whether it matches the SUG/test instructions. • **Assessment (pass/fail):** PASS if the TOE provides the defined escalation and recovery information for the selected incident state. FAIL if the authorized User cannot obtain the defined escalation/recovery information through the TOE-supported path.
Test Verdict	PASS iff TE.Log6.01, TE.Log6.02 and TE.Log6.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify authorized-user alerting, evidence preservation, mandatory-evidence cases, blocking/degradation policy, or escalation/recovery information.

SRV — Services

Table — VE and TE of AS.Srv1 (FITCEM PP: Services)

Element	Content
Services — AS.Srv1	**If security mechanisms, e.g. application isolation/sandboxing, are available on the mobile device, these shall be correctly integrated and utilized by the TOE. Application Note: This also applies to residual data in shared resources, buffers, caches, clipboard, screenshots or IPC channels that shall be cleared, blocked or isolated before reuse.**
Required vendor information	
VE.Srv1.01	The vendor shall identify the mobile-platform security mechanisms available and used by the TOE in the evaluated configuration, including application isolation/sandboxing, shared resources, screenshots, clipboard, cache, buffers and IPC controls. The vendor shall define which controls are observable through UI/platform-supported means and which require documentation, configuration or source-code evidence.
VE.Srv1.02	The vendor shall provide SUG/test instructions, supported platform configuration, test scenarios and implementation evidence sufficient to verify the selected controls. Implementation evidence may include platform manifests, entitlements, configuration snippets, design documentation, API mapping, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Srv1.01	• **Test description with test expectation:** The TOE shall use the selected platform security mechanism identified by the vendor. The expected result is consistency between the vendor evidence and the observed or evidenced TOE behaviour. • **Test preparation:** Prepare the TOE in the evaluated configuration and select one platform security mechanism from VE.Srv1.01. Prepare the relevant observation method or implementation evidence from VE.Srv1.02. • **Testing steps:** The tester records the TOE version, platform, selected mechanism, observation method and evidence type. The tester verifies the vendor evidence and, where observable, performs one normal wallet operation while the mechanism is active. • **Test result:** The actual result is the observed or evidenced use of the selected mechanism: consistent, inconsistent, missing or not verifiable. The result shall identify the evidence type and observed TOE/platform behaviour. • **Assessment (pass/fail):** PASS if the selected mechanism is evidenced for the evaluated build and observable behaviour does not contradict the evidence. FAIL if the mechanism is claimed but not evidenced, disabled, inconsistent with the evaluated build or contradicted by TOE behaviour.
TE.Srv1.02	• **Test description with test expectation:** The TOE shall clear, block or isolate residual wallet data in the selected shared or reusable resource. The expected result is no exposure of sensitive wallet data through the selected supported observation path, or sufficient evidence for non-UI-observable cleanup. • **Test preparation:** Select one wallet flow using test sensitive data and one residual-data resource from VE.Srv1.01. Prepare the UI/platform observation method or corresponding implementation evidence. • **Testing steps:** The tester executes the selected wallet flow and then changes context as defined by the vendor. For non-observable resources, the tester reviews the documented or source-code-evidenced cleanup/blocking/isolation rule. • **Test result:** The actual result is the residual-data state: cleared, masked, blocked, isolated, evidenced only, or exposed. The result shall describe the content category without copying sensitive values. • **Assessment (pass/fail):** PASS if sensitive wallet data is not exposed and non-observable handling is supported by sufficient implementation evidence. FAIL if sensitive wallet data remains visible/reusable or if the claimed non-observable control lacks adequate evidence.
TE.Srv1.03	• **Test description with test expectation:** The TOE shall block or safely handle unsupported or unauthorized IPC/shared-resource interaction relevant to wallet data or wallet operations. The expected result is safe denial, no disclosure and no unauthorized operation execution. • **Test preparation:** Select one IPC or shared-resource scenario from VE.Srv1.01 and VE.Srv1.02. Use a vendor-provided test app, test endpoint, platform-supported method or implementation evidence.

Element	Content
	• **Testing steps:** The tester reviews the declared IPC/shared-resource restriction and initiates the selected unsupported or unauthorized interaction. The tester observes the TOE UI, platform response, diagnostic output or test endpoint result. • **Test result:** The actual result is the interaction handling: denied, ignored, safely prompted, safely redirected, exposed, executed, or not evidenced. The result shall state whether wallet data or a protected operation became available. • **Assessment (pass/fail):** PASS if the interaction is safely handled and the evidence is consistent with the declared restrictions. FAIL if wallet data is exposed, a protected operation is executed, or IPC/shared-resource restrictions are missing or inconsistent.
Test Verdict	PASS iff TE.Srv1.01, TE.Srv1.02 and TE.Srv1.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the selected platform mechanisms, resources, IPC/shared-resource scenarios, expected behaviour or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Srv2 (FITCEM PP: Services)

Element	Content
Services — AS.Srv2	**The TOE shall respond robustly to internally and externally generated data. The TOE shall not enter an undefined state at any time.**
Required vendor information	
VE.Srv2.01	The vendor shall identify the internally and externally generated data classes processed by the TOE in the evaluated configuration. The vendor shall define the relevant TOE states, the allowed transitions, and the observable safe outcomes for valid, invalid, incomplete or unsupported data.
VE.Srv2.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints, test data and conformance-level negative scenarios allowing the tester to observe TOE behaviour for selected internal and external data inputs. The method shall allow detection of crash, hang, uncontrolled restart, inconsistent UI state, corrupted operation state or documented safe recovery.
Required test procedures	
TE.Srv2.01	• **Test description with test expectation:** The TOE shall process selected valid internally and externally generated data without entering an undefined state. The expected result is normal completion, documented rejection, or another specified defined state. • **Test preparation:** Prepare the TOE, test User and required test endpoint according to VE.Srv2.02. Select one valid internal-data scenario and one valid external-data scenario from VE.Srv2.01. • **Testing steps:** The tester records the TOE version, platform, selected data classes, selected scenarios and expected defined outcomes in the test evidence. The tester executes each selected scenario and observes the TOE UI, test endpoint, diagnostic output or backend result for the resulting state. • **Test result:** The actual result is the observed TOE behaviour: the TOE completes the operation, rejects it in a documented way, or reaches another defined state. The result shall identify whether any crash, hang, uncontrolled restart, inconsistent UI state or corrupted operation state was observed. • **Assessment (pass/fail):** PASS if each selected valid scenario ends in the expected defined state without crash, hang or inconsistent operation state. FAIL if the TOE enters an undefined state or the expected defined outcome cannot be observed.
TE.Srv2.02	• **Test description with test expectation:** The TOE shall handle selected invalid, incomplete or unsupported internally generated data through a defined safe outcome. The expected result is rejection, safe recovery, reinitialisation of the affected operation, or another documented defined state. • **Test preparation:** Select one internal-data negative scenario from VE.Srv2.02. Prepare the TOE state required to trigger that scenario without exploit development, reverse engineering or uncontrolled storage tampering. • **Testing steps:** The tester records the TOE version, platform, selected internal-data scenario, trigger method and expected safe outcome in the test evidence. The tester triggers the scenario and observes the TOE state, UI, diagnostic output or test backend result after the internal-data condition is processed.

Element	Content
	• **Test result:** The actual result is the observed internal-data handling: the TOE rejects, recovers, reinitialises, safely reports the condition, or enters an uncontrolled or inconsistent state. The result shall describe the observed state category and any user-visible or tester-visible message category. • **Assessment (pass/fail):** PASS if the TOE handles the selected internal-data condition through the specified defined safe outcome. FAIL if the TOE crashes, hangs, corrupts the operation state, proceeds unpredictably, or leaves the tester unable to determine the TOE state.
TE.Srv2.03	• **Test description with test expectation:** The TOE shall handle selected invalid, incomplete or unsupported externally generated data through a defined safe outcome. The expected result is rejection, safe error handling, ignored input, protected retry, or another documented defined state. • **Test preparation:** Select one external interface and one external-data negative scenario from VE.Srv2.02. Prepare the test endpoint or test data source needed to send the selected data without using exploit development or broad fuzzing. • **Testing steps:** The tester records the TOE version, platform, selected external interface, selected data scenario and expected safe outcome in the test evidence. The tester submits the selected external data to the TOE and observes the TOE UI, endpoint result, diagnostic output or backend decision. • **Test result:** The actual result is the observed external-data handling: the TOE rejects, ignores, safely reports, requests retry, or enters an uncontrolled or inconsistent state. The result shall state whether the TOE remained in a defined state and whether normal operation could continue or recover as specified. • **Assessment (pass/fail):** PASS if the TOE handles the selected external-data condition through the specified defined safe outcome and remains recoverable or operational as specified. FAIL if the TOE crashes, hangs, restarts uncontrollably, accepts the data unpredictably, or enters an undefined state.
Test Verdict	PASS iff TE.Srv2.01, TE.Srv2.02 and TE.Srv2.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify data classes, defined states, expected safe outcomes or observation methods.

Table — VE and TE of AS.Srv3 (FITCEM PP: Services)

Element	Content
Services — AS.Srv3	**The TOE shall protect the integrity, authenticity, and confidentiality of the transmitted data in accordance with the protection requirements.Note: Implemented security functions can depend on the protection requirements (integrity, confidentiality, and authenticity) of the data.**
Required vendor information	
VE.Srv3.01	The vendor shall identify the TOE transmission flows in the evaluated configuration and assign protection requirements to the transmitted data. The assignment shall state whether integrity, authenticity, confidentiality, or a defined subset of these properties is required for each selected flow.
VE.Srv3.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints or test backend support allowing the tester to execute valid and invalid transmission scenarios. The method shall allow observation of transmission acceptance, rejection, and protected-channel status without exposing secret values or unnecessary personal data.
Required test procedures	
TE.Srv3.01	• **Test description with test expectation:** The TOE shall transmit the selected data using the protection properties assigned to that data flow. The expected result is successful transmission only when the required protection state for integrity, authenticity and/or confidentiality is established. • **Test preparation:** Prepare a TOE instance, test User, and test endpoint/backend according to VE.Srv3.02. Select one representative transmission flow and identify its required protection properties from VE.Srv3.01. • **Testing steps:** The tester records the TOE version, platform, selected transmission flow, data category, required protection properties and observation method in the test evidence. The tester executes the selected valid transmission flow and observes the TOE UI, protocol metadata, test endpoint or backend decision. • **Test result:** The actual result is the observed transmission behaviour: the data is transmitted only after the required protected state is established, is blocked, or is transmitted without the required

Element	Content
	protection. The result shall describe the observed protection state and endpoint/backend outcome without recording transmitted secret values. • **Assessment (pass/fail):** PASS if the selected transmission succeeds only with the required protection properties in force. FAIL if the TOE transmits the selected data without the required integrity, authenticity or confidentiality protection.
TE.Srv3.02	• **Test description with test expectation:** The TOE shall reject or safely handle transmitted data when the selected integrity or authenticity condition fails. The expected result is safe rejection, retry, or another documented safe outcome before the data is accepted or acted upon. • **Test preparation:** Select one flow from VE.Srv3.01 for which integrity or authenticity is required. Prepare the vendor-supported negative scenario, such as modified test data, invalid sender/receiver context, invalid signature status, invalid channel endpoint, or equivalent controlled failure. • **Testing steps:** The tester records the TOE version, platform, selected flow, selected failure condition and expected safe outcome in the test evidence. The tester executes the negative scenario through the approved test setup and observes the TOE UI, protocol metadata, test endpoint or backend decision. • **Test result:** The actual result is the observed handling of the failed integrity/authenticity condition: the TOE rejects the transmission, requests retry, safely reports the failure, or accepts/processes the data. The result shall describe the visible status and whether the transmitted data was accepted or acted upon. • **Assessment (pass/fail):** PASS if the TOE does not accept or act on transmitted data when the required integrity or authenticity condition fails. FAIL if the TOE accepts, processes, or completes the operation using data whose required integrity or authenticity check failed.
TE.Srv3.03	• **Test description with test expectation:** For a flow requiring confidentiality, the TOE shall not transmit protected data through an unprotected or confidentiality-failed transmission condition. The expected result is safe rejection, protected retry, or no disclosure of the selected data category through the approved observation method. • **Test preparation:** Select one flow from VE.Srv3.01 for which confidentiality is required. Prepare the vendor-supported confidentiality-failure scenario or test endpoint configuration, without requiring decryption attacks or exposure of real personal data. • **Testing steps:** The tester records the TOE version, platform, selected flow, selected confidentiality-failure condition and observation method in the test evidence. The tester initiates the selected flow under the failure condition and observes the TOE UI, endpoint/backend decision, and approved protocol/session metadata. • **Test result:** The actual result is the observed confidentiality handling: the TOE blocks the transmission, retries only after protection is restored, sends only non-sensitive placeholder/error data, or transmits protected data despite the failed confidentiality condition. The result shall describe the observed data category and shall not copy personal data or secrets. • **Assessment (pass/fail):** PASS if protected data is not transmitted or disclosed when required confidentiality is not established. FAIL if the TOE transmits or exposes protected data despite the confidentiality requirement not being met.
Test Verdict	PASS iff TE.Srv3.01, TE.Srv3.02 and TE.Srv3.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify transmitted data flows, assigned protection requirements, negative scenarios, or observation methods.

Table — VE and TE of AS.Srv4 (FITCEM PP: Services)

Element	Content
Services — AS.Srv4	**The TOE shall validate the input and output to prevent incorrect processing.Application Note: The input and output data of the TOE itself or external interfaces shall be checked for syntax, length, and content.**
Required vendor information	

Element	Content
VE.Srv4.01	The vendor shall identify the TOE input and output data classes in the evaluated configuration and define validation rules for syntax, length and content. The vendor shall also define the observable outcomes for accepted, rejected, sanitized or safely handled data.
VE.Srv4.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints and test data allowing the tester to exercise representative valid and invalid input/output cases. The method shall allow observation of whether data was processed, rejected, displayed, transmitted, stored or safely handled.
Required test procedures	
TE.Srv4.01	• **Test description with test expectation:** The TOE shall accept and correctly process selected valid input and output data that satisfy the defined syntax, length and content rules. The expected result is normal processing and a defined successful outcome. • **Test preparation:** Prepare the TOE, test User and test endpoint according to VE.Srv4.02. Select one representative data flow and identify the corresponding input and output validation rules from VE.Srv4.01. • **Testing steps:** The tester records the TOE version, platform, selected data flow, selected input/output data classes and expected validation outcome in the test evidence. The tester submits valid input data, observes the resulting TOE output or external-interface output, and checks whether both conform to the defined validation rules. • **Test result:** The actual result is the observed TOE behaviour: the valid data is accepted and processed, the expected output is produced or transmitted, or the flow is rejected despite valid data. The result shall describe the visible UI, endpoint/backend decision or diagnostic state without copying secrets or unnecessary personal data. • **Assessment (pass/fail):** PASS if the selected valid input and output satisfy the defined rules and the TOE reaches the expected successful outcome. FAIL if valid data is incorrectly rejected, produces invalid output, or leads to undefined or incorrect processing.
TE.Srv4.02	• **Test description with test expectation:** The TOE shall reject or safely handle selected invalid input data that violates syntax, length or content rules. The expected result is no normal processing of the invalid data and no undefined TOE state. • **Test preparation:** Select one input data class and one invalid-input scenario from VE.Srv4.02. The scenario shall be representative and shall not require exploit development, broad fuzzing or uncontrolled tampering. • **Testing steps:** The tester records the TOE version, platform, selected input data class, invalid-input scenario and expected safe outcome in the test evidence. The tester submits the invalid input through the approved UI, API, test endpoint or external-interface path and observes whether the TOE rejects, sanitizes, ignores or otherwise safely handles it. • **Test result:** The actual result is the observed input-validation behaviour: the TOE rejects or safely handles the invalid input, or it accepts and processes it as if it were valid. The result shall state whether the protected operation, state change, storage action or transmission occurred. • **Assessment (pass/fail):** PASS if the invalid input is not processed as valid and the TOE remains in a defined safe state. FAIL if the TOE accepts invalid input, performs incorrect processing, or enters an undefined or inconsistent state.
TE.Srv4.03	• **Test description with test expectation:** The TOE shall validate selected output data before it is displayed, stored, transmitted or used by another TOE/external-interface component. The expected result is conformant output or safe blocking/sanitization of output that violates the defined validation rules. • **Test preparation:** Select one output data class and one vendor-supported invalid-output scenario from VE.Srv4.02. Define whether the tested output is UI-visible, stored, transmitted to an endpoint, or passed to another component. • **Testing steps:** The tester records the TOE version, platform, selected output data class, output destination and expected validation outcome in the test evidence. The tester triggers the selected output-generation or output-handling scenario and observes the UI, storage-visible state, test endpoint, backend or component response through the approved observation method. • **Test result:** The actual result is the observed output-validation behaviour: the TOE produces only conformant output, blocks or sanitizes invalid output, or emits/uses invalid output. The result shall describe the observed output category and destination without copying secret, personal or attacker-useful values.

Element	Content
	• **Assessment (pass/fail):** PASS if invalid output is blocked, sanitized or otherwise safely handled before it can cause incorrect processing. FAIL if the TOE displays, stores, transmits or uses invalid output contrary to the defined syntax, length or content rules.
Test Verdict	PASS iff TE.Srv4.01, TE.Srv4.02 and TE.Srv4.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify input/output data classes, validation rules, invalid scenarios or observation methods.

Table — VE and TE of AS.Srv5 (FITCEM PP: Services)

Element	Content
Services — AS.Srv5	**The TOE shall not disclose any critical information in error messages.Application Note: Error messages may not contain any information helpful to potential attackers.**
Required vendor information	
VE.Srv5.01	The vendor shall identify the TOE error-message channels in the evaluated configuration and define prohibited critical-information categories. The vendor shall also define the allowed content level for representative user-facing, external-interface and diagnostic/support error messages.
VE.Srv5.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints and test data allowing the tester to trigger representative error conditions. The method shall allow observation of error messages without requiring exploit development, fuzzing or exposure of real secrets/personal data.
Required test procedures	
TE.Srv5.01	• **Test description with test expectation:** User-facing TOE error messages shall not disclose critical information or attacker-useful technical details. The expected result is a safe, user-comprehensible error message limited to the allowed content level defined by the vendor. • **Test preparation:** Prepare the TOE, test User and one representative user-facing error condition according to VE.Srv5.02. Identify the prohibited critical-information categories from VE.Srv5.01 before executing the test. • **Testing steps:** The tester records the TOE version, platform, selected UI flow, selected error condition and prohibited content categories in the test evidence. The tester triggers the selected error condition through the normal TOE UI and observes the error message shown to the User. • **Test result:** The actual result is the observed UI error message: it either contains only allowed safe information, contains excessive technical detail, or discloses critical information. The result shall describe the observed content categories and shall not copy secret or personal values into the report. • **Assessment (pass/fail):** PASS if the user-facing error message does not disclose critical information or attacker-useful details. FAIL if the message exposes prohibited critical information or implementation details contrary to VE.Srv5.01.
TE.Srv5.02	• **Test description with test expectation:** Error messages or error responses observable through selected external-interface flows shall not disclose critical information. The expected result is a generic or otherwise safe error response consistent with the allowed content level. • **Test preparation:** Select one external-interface error condition from VE.Srv5.02 and identify the observation method, such as test endpoint, backend response, protocol metadata or TOE-visible status. Use only test data and approved test infrastructure. • **Testing steps:** The tester records the TOE version, platform, selected external interface, selected error condition and expected safe response category in the test evidence. The tester triggers the error condition through the approved test setup and observes the TOE-side and external-interface error response. • **Test result:** The actual result is the observed interface error behaviour: the TOE/external flow returns a safe error, returns excessive diagnostic detail, or exposes critical information. The result shall describe the visible status/message category and whether any prohibited information category was present. • **Assessment (pass/fail):** PASS if the observed external-interface error response does not disclose critical information or attacker-useful details. FAIL if the response exposes prohibited implementation, security, credential, session, key, topology or configuration information.

Element	Content
TE.Srv5.03	• **Test description with test expectation:** Supported diagnostic, support or detailed-error flows shall not disclose critical information beyond the vendor-defined allowed diagnostic content. The expected result is redaction, omission, generic categorisation or another documented safe disclosure rule. • **Test preparation:** Prepare one TOE error condition for which a diagnostic/support/detailed-error path is available, if such a path exists in the evaluated configuration. Identify the allowed and prohibited diagnostic content categories from VE.Srv5.01. • **Testing steps:** The tester records the TOE version, platform, selected error condition, selected diagnostic/support path and prohibited content categories in the test evidence. The tester triggers the error condition, opens or exports the supported diagnostic/detailed-error view, and observes its content using the approved method. • **Test result:** The actual result is the observed diagnostic/error-detail content: it either contains only allowed redacted categories, contains excessive diagnostic details, or discloses critical information. If no such path exists, the result shall record the documented non-applicability for the evaluated configuration. • **Assessment (pass/fail):** PASS if the supported diagnostic/detailed-error path does not disclose prohibited critical information, or if the path is documented as not applicable. FAIL if the supported path exposes critical information or attacker-useful details contrary to VE.Srv5.01.
Test Verdict	PASS iff TE.Srv5.01, TE.Srv5.02 and TE.Srv5.03 all PASS or are documented as not applicable where the corresponding channel does not exist. FAIL if any applicable TE fails or if VE evidence is insufficient to identify error-message channels, prohibited critical-information categories, or allowed error-message content.

Table — VE and TE of AS.Srv6 (FITCEM PP: Services)

Element	Content
Services — AS.Srv6	**In its default configuration, the TOE shall enable only those functions which are strictly necessary for its intended operation. All non-essential functions shall be disabled by default or removed. If the TOE includes optional functions, it shall provide an administrative interface to enable or disable them without compromising the Secure State of the TOE.**
Required vendor information	
VE.Srv6.01	The vendor shall identify the TOE functions enabled in the default evaluated configuration and justify why each enabled function is strictly necessary for the intended operation. The vendor shall also identify non-essential, removed, disabled-by-default and optional functions.
VE.Srv6.02	The vendor shall provide SUG/test instructions, test accounts and configuration information allowing the tester to install the TOE in default configuration, observe enabled functions, and, where applicable, enable/disable optional functions through the administrative interface. The vendor shall define observable Secure State criteria before and after optional-function changes.
Required test procedures	
TE.Srv6.01	• **Test description with test expectation:** In default configuration, the TOE shall enable only functions identified as strictly necessary for intended operation. The expected result is that non-essential or optional functions are not available for normal use before explicit enablement. • **Test preparation:** Prepare a fresh TOE installation according to VE.Srv6.02 and identify the default-enabled, non-essential and optional function categories from VE.Srv6.01. Select representative functions that should be enabled by default and, where present, functions that should be disabled or removed by default. • **Testing steps:** The tester records the TOE version, platform, installation source, default configuration state, selected enabled functions and selected disabled/removed/optional functions in the test evidence. The tester opens the TOE after fresh installation and observes the available UI entries, supported service paths or test-accessible functions relevant to the selected sample. • **Test result:** The actual result is the observed default-configuration behaviour: only strictly necessary functions are available, or non-essential/optional functions are also visible or executable. The result shall identify the observed availability category for each selected function without relying on source-code inspection.

Element	Content
	• **Assessment (pass/fail):** PASS if the selected default-enabled functions are available and the selected non-essential or optional functions are disabled, unavailable or removed by default. FAIL if any selected non-essential or optional function is enabled by default contrary to VE.Srv6.01.
TE.Srv6.02	• **Test description with test expectation:** If the TOE includes optional functions, the TOE shall provide an administrative interface to enable and disable them. The expected result is that the selected optional function changes state only through the defined administrative path. • **Test preparation:** Select one optional function from VE.Srv6.01 and prepare the administrative interface or configuration path defined in VE.Srv6.02. If the TOE has no optional functions, this TE is documented as not applicable with vendor justification. • **Testing steps:** The tester records the TOE version, platform, selected optional function, administrative interface path and expected enabled/disabled states in the test evidence. The tester attempts to use the selected optional function before enablement, enables it through the administrative interface, verifies availability, disables it again, and verifies unavailability. • **Test result:** The actual result is the observed optional-function state transition: the function is unavailable before enablement, available after administrative enablement, and unavailable again after administrative disablement, or it behaves inconsistently. The result shall describe the UI/configuration state and functional availability observed by the tester. • **Assessment (pass/fail):** PASS if the selected optional function can be enabled and disabled only through the defined administrative interface and its availability follows the configured state. FAIL if the optional function cannot be controlled as specified or remains available when disabled.
TE.Srv6.03	• **Test description with test expectation:** Enabling or disabling an optional function shall not compromise the Secure State of the TOE. The expected result is that vendor-defined Secure State criteria remain satisfied after the optional-function state change. • **Test preparation:** Use the optional function tested in TE.Srv6.02 and identify the Secure State criteria from VE.Srv6.02. Select one representative security-relevant wallet operation or configuration check that evidences the Secure State after the optional-function change. • **Testing steps:** The tester records the TOE version, platform, selected optional function, selected Secure State criteria and selected security-relevant check in the test evidence. The tester enables and disables the optional function through the administrative interface and, after each state change, performs the selected Secure State check using the approved observation method. • **Test result:** The actual result is the observed Secure State behaviour: the TOE remains in the defined Secure State after optional-function changes, enters a documented safe restricted state, or violates the defined Secure State criteria. The result shall describe the visible access decision, configuration status, UI state, endpoint/backend decision or diagnostic state used as evidence. • **Assessment (pass/fail):** PASS if the TOE satisfies the defined Secure State criteria after each tested optional-function change. FAIL if enabling or disabling the optional function disables, bypasses or contradicts the defined Secure State criteria.
Test Verdict	PASS iff TE.Srv6.01 PASS and TE.Srv6.02/TE.Srv6.03 PASS where optional functions exist. FAIL if any applicable TE fails or if VE evidence is insufficient to identify default-enabled functions, non-essential functions, optional functions, administrative interface behaviour or Secure State criteria.

Table — VE and TE of AS.Srv7 (FITCEM PP: Services)

Element	Content
Services — AS.Srv7	**Only services required to provide the TOE's functions may be present on the TOE. Any information related to the development shall be removed.Application Note: This requirement does not apply to the debug version of the TOE.**
Required vendor information	
VE.Srv7.01	The vendor shall provide the service inventory for the evaluated TOE build and identify which services are required to provide the TOE's functions. The vendor shall also identify any excluded, debug-only, development-only, test, staging or support services that are not part of the evaluated production TOE.
VE.Srv7.02	The vendor shall provide SUG/test instructions and observation methods allowing the tester to inspect the evaluated build, supported service paths, service availability and visible development-related

Element	Content
	information. The vendor shall define how the tester distinguishes the evaluated TOE build from any debug version.
Required test procedures	
TE.Srv7.01	• **Test description with test expectation:** The evaluated TOE build shall expose or enable only services required to provide the TOE's functions. The expected result is that every observed service corresponds to a required TOE function identified in VE.Srv7.01. • **Test preparation:** Prepare the evaluated TOE build in the certified configuration and obtain the vendor-declared service inventory. Select the supported observation method for service availability, such as UI, configuration view, approved diagnostic output, test endpoint, or platform-supported service listing. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected observation method, declared required services and selected service sample in the test evidence. The tester observes available services through the approved method and compares them with the vendor-declared service inventory. • **Test result:** The actual result is the observed service availability: only declared required services are present, or additional undeclared/non-required services are visible or executable. The result shall identify observed service categories without relying on source-code inspection. • **Assessment (pass/fail):** PASS if each observed service is declared and required for TOE functions. FAIL if any undeclared, non-required, test, staging, support or development service is present in the evaluated TOE build.
TE.Srv7.02	• **Test description with test expectation:** The evaluated TOE build shall not expose development-related information through supported user-facing, service-facing or diagnostic paths. The expected result is that development, debug, test, staging, source-path, internal-build or developer-only information is absent or redacted. • **Test preparation:** Identify the prohibited development-information categories from VE.Srv7.01 and VE.Srv7.02. Prepare representative TOE paths where such information could normally appear, such as about/version screens, diagnostics, support bundles, error detail views, configuration views or supported exports. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected TOE paths and prohibited development-information categories in the test evidence. The tester opens or triggers the selected paths using only supported interactions and observes whether development-related information is displayed or exported. • **Test result:** The actual result is the observed information exposure: the TOE shows only production-appropriate information, shows redacted/generic information, or exposes development-related information. The result shall describe categories of exposed information and shall not copy sensitive internal values if exposure occurs. • **Assessment (pass/fail):** PASS if no prohibited development-related information is exposed through the tested supported paths. FAIL if the evaluated TOE build exposes development, debug, test, staging, source, internal endpoint, internal credential, or developer-only information contrary to VE.Srv7.01.
TE.Srv7.03	• **Test description with test expectation:** The build used for Srv7 conformance testing shall be identifiable as the evaluated non-debug TOE build. The expected result is that debug-only services or development information are not present in the evaluated build, and any debug version is clearly outside the evaluated configuration. • **Test preparation:** Obtain the vendor-defined build identification method from VE.Srv7.02. If the vendor uses separate debug and production builds, obtain the criteria that distinguish them without requiring reverse engineering. • **Testing steps:** The tester records the TOE version, platform, build identifier and build-identification evidence in the test report. The tester verifies through the approved method that the installed TOE corresponds to the evaluated build and observes whether debug-only indicators, controls or services are present. • **Test result:** The actual result is the observed build status: the TOE is identifiable as the evaluated non-debug build, or it appears to be a debug/ambiguous build. The result shall describe whether debug-only services, debug indicators or development controls were visible.

Element	Content
	• **Assessment (pass/fail):** PASS if the tested TOE is the evaluated non-debug build and no debug-only service or development control is present. FAIL if the tested build is debug, ambiguous, or contains debug-only services within the evaluated configuration.
Test Verdict	PASS iff TE.Srv7.01, TE.Srv7.02 and TE.Srv7.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify required services, prohibited development information, or the evaluated non-debug TOE build.

Table — VE and TE of AS.Srv8 (FITCEM PP: Services)

Element	Content
Services — AS.Srv8	**The TOE shall regularly check for available updates from trustworthy sources. If the update is available, the TOE shall prompt users to install it.**
Required vendor information	
VE.Srv8.01	The vendor shall specify the TOE update-check mechanism, including the update source, trust basis, check frequency, trigger conditions and observable outcomes for “no update”, “update available” and “source not trusted”. The vendor shall also define whether the check is automatic, user-triggered, backend-triggered, platform-store-triggered, or a combination of these.
VE.Srv8.02	The vendor shall provide SUG/test instructions, test accounts, test backend/store configuration or equivalent test support allowing the tester to observe update checking and user prompting. The vendor shall define how the tester can distinguish a trusted update source from an untrusted or invalid update source without requiring reverse engineering or attack tooling.
Required test procedures	
TE.Srv8.01	• **Test description with test expectation:** The TOE shall perform the vendor-defined update availability check against the configured trustworthy update source. The expected result is a defined “no update available” or equivalent safe status when no newer update is available. • **Test preparation:** Prepare the TOE in the evaluated configuration and configure the vendor-supported update source or test update source according to VE.Srv8.02. Ensure that the test source is in a state where no newer update is available for the tested TOE build. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected update source, check trigger and expected “no update” outcome in the test evidence. The tester triggers or waits for the vendor-defined update check and observes the TOE UI, platform-store status, test backend response or approved diagnostic output. • **Test result:** The actual result is the observed update-check behaviour: the TOE reports no update available, remains in a documented current state, fails to perform the check, or reaches an ambiguous state. The result shall describe the visible status or backend/store response category observed by the tester. • **Assessment (pass/fail):** PASS if the TOE performs the update check and reaches the documented “no update/current version” outcome. FAIL if the TOE does not perform the check, cannot report a defined status, or checks a source not identified as trustworthy.
TE.Srv8.02	• **Test description with test expectation:** If an update is available from a trustworthy source, the TOE shall prompt the User to install it. The expected result is a clear user-facing prompt or equivalent platform-supported update notification linked to the trusted installation path. • **Test preparation:** Prepare the TOE and vendor-supported update source in a state where a newer update is available for the tested build. Ensure that the update source is the source identified as trustworthy in VE.Srv8.01. • **Testing steps:** The tester records the TOE version, platform, build identifier, trusted update source, available update identifier/category and expected prompt behaviour in the test evidence. The tester performs the update check and observes the TOE UI, platform-store prompt, in-app notification or approved update-status screen. • **Test result:** The actual result is the observed user-prompt behaviour: the TOE prompts the User to install the available update, provides only a passive status, provides no prompt, or directs the User to an untrusted installation path. The result shall describe the prompt category, user action offered and whether the path corresponds to the trusted source.

Element	Content
	• **Assessment (pass/fail):** PASS if the TOE informs the User that an update is available and provides a trusted installation path or platform-supported installation action. FAIL if no prompt is shown, the prompt is unclear, or the prompt directs the User to an untrusted or undocumented source.
TE.Srv8.03	• **Test description with test expectation:** The TOE shall not prompt the User to install an update from a source that is not recognized as trustworthy for the evaluated configuration. The expected result is safe rejection, no update prompt, or a documented warning/error state. • **Test preparation:** Prepare the vendor-supported untrusted-source or invalid-source scenario from VE.Srv8.02. The scenario shall not require exploit development, network attack tooling or reverse engineering of the update mechanism. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected untrusted/invalid source scenario and expected safe outcome in the test evidence. The tester performs the update check under that scenario and observes the TOE UI, platform-store status, test backend response or approved diagnostic output. • **Test result:** The actual result is the observed source-trust behaviour: the TOE rejects the source, shows a safe warning/error, remains on the trusted source, or prompts the User to install from the untrusted source. The result shall describe the visible status and whether any installation prompt was offered. • **Assessment (pass/fail):** PASS if the TOE does not prompt installation from the untrusted or invalid source and reaches the documented safe outcome. FAIL if the TOE treats the untrusted source as valid or prompts the User to install from it.
Test Verdict	PASS iff TE.Srv8.01, TE.Srv8.02 and TE.Srv8.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the update source, trust basis, check frequency, trigger conditions, user prompt behaviour or negative source scenario.

Table — VE and TE of AS.Srv9 (FITCEM PP: Services)

Element	Content
Services — AS.Srv9	**The TOE shall implement state-of-the-art rooting and jailbreaking detection and, upon detection of rooted or jailbroken operating system TOE shall prevent operation.**
Required vendor information	
VE.Srv9.01	The vendor shall provide the rooting/jailbreaking detection specification for each evaluated platform, including detected signal categories, platform APIs or integrity services used, detection states, blocked operations, recovery conditions and expected observable outcomes. The vendor shall identify which parts are observable through UI/platform-supported means and which require documentation, configuration or source-code evidence.
VE.Srv9.02	The vendor shall provide SUG/test instructions, supported devices or test configurations, one non-rooted/non-jailbroken baseline scenario and one rooted/jailbroken or vendor-supported equivalent detection scenario. Implementation evidence may include design documentation, API mapping, detection-rule description, library/version information, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Srv9.01	• **Test description with test expectation:** On a supported non-rooted/non-jailbroken platform, the TOE shall remain operational and shall not enter the rooted/jailbroken blocking state. The expected result is normal availability of the selected wallet operation. • **Test preparation:** Prepare the TOE in the evaluated configuration on a supported non-rooted/non-jailbroken platform. Select one representative wallet operation and prepare the relevant observation method or implementation evidence from VE.Srv9.02. • **Testing steps:** The tester records the TOE version, platform, OS version, device/test configuration, selected operation, observation method and evidence type. The tester reviews the evidence for the evaluated build and performs the selected wallet operation. • **Test result:** The actual result is the observed baseline behaviour: normal operation, no rooted/jailbroken blocking state, unexpected block, or evidence inconsistency. The result shall describe the visible UI/status, diagnostic state or endpoint/backend outcome.

Element	Content
	• **Assessment (pass/fail):** PASS if the TOE remains operational in the supported non-compromised configuration and the evidence does not contradict the observed behaviour. FAIL if the TOE blocks normal operation without documented reason, cannot reach the expected operational state, or the evaluated build lacks evidence of the claimed detection mechanism.
TE.Srv9.02	• **Test description with test expectation:** When the selected rooted/jailbroken or approved equivalent detection condition is present, the TOE shall detect it and prevent operation. The expected result is a blocked, terminated, or safely restricted state before wallet functions become available. • **Test preparation:** Prepare one rooted/jailbroken or vendor-supported equivalent scenario for an applicable platform. Identify the expected detection state, selected blocked wallet operation and corresponding evidence from VE.Srv9.01 and VE.Srv9.02. • **Testing steps:** The tester records the TOE version, platform, selected condition, trigger method, selected wallet operation and expected blocked state. The tester reviews the detection evidence and starts or resumes the TOE under the selected condition. • **Test result:** The actual result is the observed detection behaviour: detected and blocked, safely restricted, inconclusive, not detected, or inconsistent with the evidence. The result shall state whether the selected wallet operation was available or completed. • **Assessment (pass/fail):** PASS if the TOE detects the selected condition and prevents the selected wallet operation from proceeding, with evidence consistent with the evaluated build. FAIL if the TOE treats the condition as normal, permits wallet operation, or the claimed detection path lacks adequate documentation or implementation evidence.
TE.Srv9.03	• **Test description with test expectation:** After rooting/jailbreaking has been detected, the TOE shall continue to prevent security-relevant wallet operation until the vendor-defined recovery condition is met. The expected result is that only documented safe actions remain available, if any. • **Test preparation:** Use the detected/blocked state obtained in TE.Srv9.02. Identify blocked operations, permitted safe actions, recovery condition and supporting evidence from VE.Srv9.01. • **Testing steps:** The tester records the selected blocked operation, permitted safe actions, recovery condition and evidence type. While the rooted/jailbroken condition remains present, the tester attempts one selected security-relevant wallet operation and observes the TOE response. • **Test result:** The actual result is the post-detection state: operation remains blocked, only documented safe actions are available, operation resumes, or blocking/recovery behaviour is not evidenced. The result shall state whether any security-relevant wallet operation completed. • **Assessment (pass/fail):** PASS if security-relevant wallet operation remains prevented while the condition is detected and the permitted safe actions match the evidence. FAIL if protected wallet functions continue or resume contrary to VE.Srv9.01, or if the blocking/recovery logic is claimed without adequate evidence.
Test Verdict	PASS iff TE.Srv9.01, TE.Srv9.02 and TE.Srv9.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the detection mechanism, test trigger method, observable detection state, blocked operations, safe actions, recovery condition or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Srv10 (FITCEM PP: Services)

Element	Content
Services — AS.Srv10	**The TOE shall enforce that the User operates the Wallet Instance under a standard user account on the mobile device with no administrative (root/superuser) privileges. The application shall run exclusively within the standard sandboxed execution context provided by the operating system (Android or iOS).**
Required vendor information	
VE.Srv10.01	The vendor shall specify the supported standard-user execution context for each platform, including how the TOE determines or relies on the absence of administrative, root or superuser privileges. The vendor shall define the observable outcomes for standard-user operation, elevated/admin/root operation, and unsupported or inconclusive execution context.
VE.Srv10.02	The vendor shall provide SUG/test instructions, supported devices or platform test configurations, and conformance-level observation methods allowing the tester to verify standard-account operation and sandboxed execution. The vendor shall also provide a supported negative scenario for

Element	Content
	elevated/root/superuser or non-standard execution context without requiring exploit development or reverse engineering.
Required test procedures	
TE.Srv10.01	• **Test description with test expectation:** The TOE shall operate under the standard mobile-device user account without requiring administrative, root or superuser privileges. The expected result is normal operation in the supported standard-user context. • **Test preparation:** Prepare the TOE in the evaluated secure configuration on a supported Android or iOS device without administrative/root/superuser privileges. Select one representative wallet operation that demonstrates normal TOE operation in the standard-user context. • **Testing steps:** The tester records the TOE version, platform, OS version, device/test configuration, selected user-account context and selected wallet operation in the test evidence. The tester starts the TOE, verifies the standard-user context using the vendor-supported observation method, and performs the selected wallet operation. • **Test result:** The actual result is the observed application behaviour: the TOE operates normally in the standard-user context, requires elevated privileges, or fails in a way inconsistent with the evaluated configuration. The result shall describe the visible UI/status, platform setting, diagnostic state or endpoint/backend outcome observed by the tester. • **Assessment (pass/fail):** PASS if the TOE operates normally under the standard user account and does not require administrative/root/superuser privileges. FAIL if the TOE requires or requests such privileges for normal operation.
TE.Srv10.02	• **Test description with test expectation:** The TOE shall not permit normal wallet operation when the User or TOE execution context has administrative, root or superuser privileges. The expected result is safe blocking, restricted mode, termination, or another vendor-defined safe outcome. • **Test preparation:** Prepare the vendor-supported elevated/admin/root/superuser or equivalent non-standard execution-context scenario from VE.Srv10.02. Identify one security-relevant wallet operation that must not proceed in that context. • **Testing steps:** The tester records the TOE version, platform, selected non-standard execution-context scenario, trigger method, selected wallet operation and expected safe outcome in the test evidence. The tester starts or resumes the TOE under the selected scenario and observes the TOE UI, diagnostic state, platform-integrity signal, test endpoint or backend decision. • **Test result:** The actual result is the observed enforcement behaviour: the TOE blocks or restricts operation, reports an unsupported/inconclusive context, or continues normal wallet operation. The result shall describe the visible state and whether the selected wallet operation was available or completed. • **Assessment (pass/fail):** PASS if the TOE prevents normal wallet operation in the selected administrative/root/superuser or non-standard execution context. FAIL if the TOE permits protected wallet functions to proceed as normal in that context.
TE.Srv10.03	• **Test description with test expectation:** The TOE application shall run exclusively within the standard sandboxed execution context provided by Android or iOS. The expected result is that the TOE is observed as a standard sandboxed application and does not use a non-standard privileged execution context for wallet operation. • **Test preparation:** Prepare the TOE in the evaluated configuration and identify the vendor-supported observation method for sandboxed execution context. Select one wallet operation that exercises normal TOE processing while the execution context is observed. • **Testing steps:** The tester records the TOE version, platform, OS version, build identifier, selected sandbox-observation method and selected wallet operation in the test evidence. The tester observes the TOE execution context using the approved method, performs the selected operation, and checks whether any non-standard privileged helper or execution context is required or exposed. • **Test result:** The actual result is the observed execution-context behaviour: the TOE remains within the standard sandboxed context, or wallet operation depends on a non-standard privileged context. The result shall describe the observed sandbox/application context and any visible privileged helper, permission state or diagnostic indication. • **Assessment (pass/fail):** PASS if the TOE is observed to run in the standard sandboxed execution context and the selected wallet operation does not require a privileged or non-standard execution

Element	Content
	context. FAIL if the TOE operates outside the standard sandbox or depends on an undocumented privileged execution context.
Test Verdict	PASS iff TE.Srv10.01, TE.Srv10.02 and TE.Srv10.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the standard-user context, non-standard/elevated context scenario, sandbox-observation method or expected safe outcome.

Table — VE and TE of AS.Srv11 (FITCEM PP: Services)

Element	Content
Services — AS.Srv11	**TOE shall provide each application user with only one role. The users have access to the same application functionality and operate with the same privileges.**
Required vendor information	
VE.Srv11.01	The vendor shall specify the application user model for the evaluated TOE, including the single user role, the available application functionality and the privilege model applied to application users. The vendor shall identify any non-user, administrative, support, testing, backend or platform roles that are outside the evaluated application-user role.
VE.Srv11.02	The vendor shall provide SUG/test instructions, test accounts or test personas, and supported observation methods allowing the tester to compare application functionality and privileges for different application users. The vendor shall define the expected observable behaviour for user onboarding, normal operation, role/privilege visibility and denied role-elevation attempts, where such attempts are supported by the UI or test configuration.
Required test procedures	
TE.Srv11.01	• **Test description with test expectation:** The TOE shall expose only one application-user role in the evaluated configuration. The expected result is that onboarding, account/profile views and supported settings do not allow selection, assignment or change to another application-user role. • **Test preparation:** Prepare the TOE in the evaluated configuration and obtain the declared application-user role model from VE.Srv11.01. Prepare one fresh test User and identify the supported UI/configuration paths where role or account type information could be visible. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected test User, declared role model and selected observation paths in the test evidence. The tester onboards or activates the test User, opens the selected account/profile/settings paths, and observes whether any additional application-user role can be selected, assigned or displayed. • **Test result:** The actual result is the observed role behaviour: the TOE shows or implies a single application-user role, or it exposes multiple application-user roles or role-selection controls. The result shall describe the visible role/account-type categories without relying on source-code inspection. • **Assessment (pass/fail):** PASS if only one application-user role is visible, assigned and usable in the evaluated configuration. FAIL if the TOE allows or exposes multiple application-user roles for application users.
TE.Srv11.02	• **Test description with test expectation:** Different application users shall have access to the same application functionality under the same evaluated configuration. The expected result is that any observed differences are attributable to user data/state or lifecycle status, not to different application-user roles. • **Test preparation:** Prepare two application users or two equivalent test personas according to VE.Srv11.02. Bring both users to the same relevant lifecycle state where possible, or document any necessary state difference before testing. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected users/personas, selected lifecycle state and selected functionality sample in the test evidence. The tester accesses the same representative application functions for each user and observes the available UI entries, action controls and operation outcomes. • **Test result:** The actual result is the observed functionality comparison: both users see and can use the same functional categories, or one user has additional or restricted application functionality due to a role/privilege distinction. The result shall distinguish functional availability from differences caused by user-specific wallet content or lifecycle state.

Element	Content
	• **Assessment (pass/fail):** PASS if tested application users have the same application functionality in equivalent state. FAIL if users in equivalent state have different functionality because of different application-user roles or privileges.
TE.Srv11.03	• **Test description with test expectation:** Application users shall operate with the same privileges and shall not be able to elevate or switch application-user privileges through supported TOE paths. The expected result is absence of role-elevation controls or safe denial of any vendor-defined role/privilege-change attempt. • **Test preparation:** Use one or both test users from TE.Srv11.02 and identify supported paths that could affect privileges, such as profile, settings, support, recovery or account-management flows. If the TOE has no such path, document non-applicability for the role-elevation attempt and still verify absence of privilege controls. • **Testing steps:** The tester records the TOE version, platform, selected user, selected privilege-related paths and expected safe outcome in the test evidence. The tester attempts any supported privilege or account-type change path, or verifies that no such control exists, and then observes whether additional application functions or privileged operations become available. • **Test result:** The actual result is the observed privilege behaviour: no elevation path exists, the attempted change is safely denied, or additional privileges/functions become available. The result shall describe the visible UI/status and whether any privilege-dependent operation became available. • **Assessment (pass/fail):** PASS if application users remain under the same privilege model and no supported path grants a different application-user privilege level. FAIL if the TOE permits role switching, privilege elevation or privilege differentiation for application users contrary to VE.Srv11.01.
Test Verdict	PASS iff TE.Srv11.01, TE.Srv11.02 and TE.Srv11.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the application-user role model, functionality sample, privilege model or supported observation paths.

Table — VE and TE of AS.Srv12 (FITCEM PP: Services)

Element	Content
Services — AS.Srv12	**The TOE shall associate security and privacy attributes with wallet requests, credentials, attestations, presentations, confirmations, logs and communication sessions, including at least purpose, recipient or requester identity, requested and released attribute set, consent state, trust or authorization status, assurance level, session or transaction identifiers and relevant channel context. The TOE shall preserve the integrity of those attributes and enforce access-control, disclosure, validation and logging decisions based on them.**
Required vendor information	
VE.Srv12.01	The vendor shall provide the security/privacy-attribute model used by the TOE for wallet requests, credentials, attestations, presentations, confirmations, logs and communication sessions. The model shall define mandatory attributes, dependent TOE decisions, observation methods, and which associations or decisions require documentation, configuration or source-code evidence.
VE.Srv12.02	The vendor shall provide SUG/test instructions, test accounts, test credentials or attestations, test endpoints and conformance-level negative scenarios allowing the tester to verify attribute association, decision enforcement and integrity preservation. Implementation evidence may include data model excerpts, decision tables, state/sequence diagrams, API mapping, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Srv12.01	• **Test description with test expectation:** The TOE shall associate the required security/privacy attributes with the selected wallet flow and related artefacts. The expected result is that the association is observable or sufficiently evidenced for the evaluated build. • **Test preparation:** Prepare the TOE, authenticated test User, test credential/attestation and test endpoint according to VE.Srv12.02. Select one wallet request, presentation or confirmation flow and identify required attributes from VE.Srv12.01.

Element	Content
	• **Testing steps:** The tester records the TOE version, platform, selected flow, selected artefacts, required attribute categories, observation method and evidence type. The tester executes the flow and reviews UI, log, test endpoint, backend response, diagnostic output or implementation evidence. • **Test result:** The actual result is the attribute-association state: present and linked, partially present, absent, ambiguous, or evidenced only. The result shall describe attribute categories and linkage, not personal values, tokens or secrets. • **Assessment (pass/fail):** PASS if the required attributes are associated with the selected flow and are observable or supported by sufficient implementation evidence. FAIL if required attributes are missing, ambiguous, not linked to the relevant operation/artefact, or claimed without adequate evidence.
TE.Srv12.02	• **Test description with test expectation:** The TOE shall enforce access-control, disclosure, validation and logging decisions based on associated security/privacy attributes. The expected result is that a valid attribute state permits the selected operation, while an invalid or disallowed state is rejected or safely handled. • **Test preparation:** Select one attribute-dependent decision from VE.Srv12.01, such as disclosure authorization, requester trust validation, consent enforcement or logging decision. Prepare one valid scenario and one vendor-supported invalid/disallowed scenario. • **Testing steps:** The tester records the selected decision, attribute condition, valid scenario, invalid/disallowed scenario and evidence type. The tester executes both scenarios and reviews the TOE UI, log entry, test endpoint, backend decision or decision-rule evidence. • **Test result:** The actual result is the decision behaviour: permitted, rejected, blocked, re-prompted, disclosure-limited, logged as specified, or inconsistent. The result shall state the decision category and visible/evidenced outcome. • **Assessment (pass/fail):** PASS if the TOE decision changes consistently with the associated attribute state and the invalid/disallowed state is not processed as valid. FAIL if the TOE ignores required attributes, permits the operation contrary to the specified rule, or the claimed rule lacks adequate evidence.
TE.Srv12.03	• **Test description with test expectation:** The TOE shall preserve the integrity of security/privacy attributes across the selected flow. The expected result is safe rejection, revalidation, re-prompting or another documented safe outcome if the attribute context becomes mismatched, altered or stale. • **Test preparation:** Select one flow and one vendor-supported integrity negative scenario from VE.Srv12.02, such as mismatched purpose, requester identity, released attribute set, consent state, transaction identifier or channel context. Prepare the corresponding observation method or implementation evidence. • **Testing steps:** The tester records the selected flow, integrity scenario, expected safe outcome and evidence type. The tester executes the approved mismatch/tamper/stale-context scenario and reviews TOE UI, test endpoint, backend decision, resulting log/evidence state or implementation evidence. • **Test result:** The actual result is the integrity handling: rejected, revalidated, re-prompted, safely logged, completed under altered context, or not evidenced. The result shall state whether any operation completed under the altered context. • **Assessment (pass/fail):** PASS if the TOE does not complete the operation under mismatched, altered or stale attributes unless it first revalidates and obtains the required decision again. FAIL if altered attributes are silently accepted, cause access/disclosure/validation/logging under the wrong context, or the claimed integrity handling lacks adequate evidence.
Test Verdict	PASS iff TE.Srv12.01, TE.Srv12.02 and TE.Srv12.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify required attributes, dependent decisions, integrity scenarios, observation methods or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Srv13 (FITCEM PP: Services)

Element	Content
Services — AS.Srv13	**The TOE shall provide security-relevant events or status information needed for monitoring processes performed by the Wallet Provider.Note: Such information includes security alerts, advisories and directives relevant to the TOE, its dependencies, cryptographic mechanisms, supported platforms and backend interfaces.**

Element	Content
Required vendor information	
VE.Srv13.01	The vendor shall define the security-relevant events and status information provided by the TOE for Wallet Provider monitoring. The definition shall identify the event/status categories, trigger conditions, destination or access method, expected format, timing, and observable delivery outcome.
VE.Srv13.02	The vendor shall provide SUG/test instructions, test accounts, monitoring endpoint or test backend support, and conformance-level trigger methods allowing the tester to observe selected events/status information. The method shall avoid exposure of secrets, personal data, reusable tokens or internal monitoring credentials.
Required test procedures	
TE.Srv13.01	• **Test description with test expectation:** The TOE shall provide selected security-relevant status information to the Wallet Provider monitoring process or approved test equivalent. The expected result is that the selected status categories are available through the vendor-defined monitoring path. • **Test preparation:** Prepare the TOE, test User where needed, and Wallet Provider monitoring endpoint or test backend according to VE.Srv13.02. Select representative status categories from VE.Srv13.01, covering the TOE state and at least one relevant dependency, platform, cryptographic-mechanism or backend-interface status where applicable. • **Testing steps:** The tester records the TOE version, platform, selected monitoring path, selected status categories and expected delivery outcome in the test evidence. The tester starts the TOE in the evaluated configuration, triggers or waits for the vendor-defined status publication/check, and observes the monitoring endpoint, test backend, approved diagnostic output or administrative monitoring view. • **Test result:** The actual result is the observed monitoring-status behaviour: the selected status categories are provided, partially provided, delayed beyond the defined condition, or not provided. The result shall describe the visible status categories and delivery state without copying secrets, tokens or personal data. • **Assessment (pass/fail):** PASS if the selected security-relevant status information is provided through the defined monitoring path as specified. FAIL if required selected status information is missing, ambiguous, delivered to the wrong path, or not observable through the approved method.
TE.Srv13.02	• **Test description with test expectation:** The TOE shall provide a monitoring event or updated status when a selected security-relevant condition occurs. The expected result is an observable event/status update corresponding to the triggered condition. • **Test preparation:** Select one security-relevant event or status-change condition from VE.Srv13.01 and prepare the vendor-supported trigger method. Ensure that the trigger does not require exploit development, uncontrolled tampering, or disclosure of monitoring secrets. • **Testing steps:** The tester records the TOE version, platform, selected event/status condition, trigger method, monitoring path and expected event/status outcome in the test evidence. The tester triggers the selected condition and observes the monitoring endpoint, test backend, administrative view or approved diagnostic output for the corresponding update. • **Test result:** The actual result is the observed event-reporting behaviour: the TOE provides the expected event/status update, provides an incomplete or generic update, or provides no observable update. The result shall describe the event/status category, severity or state category where visible, without copying sensitive operational details unnecessarily. • **Assessment (pass/fail):** PASS if the selected security-relevant condition results in the specified monitoring event or status update. FAIL if the condition is not reported, is reported under an incorrect category, or cannot be correlated to the triggered condition through the approved observation method.
TE.Srv13.03	• **Test description with test expectation:** The TOE or its monitoring integration shall support provision of selected alert/advisory/directive information relevant to the TOE scope. The expected result is that the selected information category is available to the Wallet Provider monitoring process and is associated with the relevant TOE, dependency, platform, cryptographic mechanism or backend-interface category. • **Test preparation:** Select one alert, advisory or directive category from VE.Srv13.01 and prepare the vendor-supported test message, simulated advisory or equivalent test condition. Define before execution which category and relevance mapping shall be observable.

Element	Content
	• **Testing steps:** The tester records the TOE version, platform, selected alert/advisory/directive category, relevance mapping and observation method in the test evidence. The tester introduces the selected test condition through the approved monitoring setup and observes whether the information is made available with the expected category and relevance mapping. • **Test result:** The actual result is the observed alert/advisory/directive handling: the information is available with the expected category and relevance mapping, is available without sufficient mapping, or is not available. The result shall describe the information category and mapping outcome without copying sensitive advisory details that are not needed for evidence. • **Assessment (pass/fail):** PASS if the selected alert/advisory/directive category is provided or reflected in the monitoring information as specified. FAIL if the TOE or monitoring integration omits the selected category, misclassifies it, or fails to associate it with the relevant TOE/dependency/platform/crypto/backend-interface scope.
Test Verdict	PASS iff TE.Srv13.01, TE.Srv13.02 and TE.Srv13.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify monitoring information categories, trigger conditions, delivery paths, observation methods or expected outcomes.

STRG — Local storage

Table — VE and TE of AS.Strg1 (FITCEM PP: Local storage)

Element	Content
Local storage — AS.Strg1	**The TOE shall implement measures against unauthorised access to the stored data.**
Required vendor information	
VE.Strg1.01	The vendor shall identify stored-data classes in the evaluated configuration and the authorisation condition required to access each class. The vendor shall define which access-control measures are observable through UI/platform-supported means and which require documentation, configuration or source-code evidence.
VE.Strg1.02	The vendor shall provide SUG/test instructions, test accounts, test stored data, authorised and unauthorised states, supported access paths and local/platform paths relevant to stored data. Implementation evidence may include data-flow descriptions, access-control mapping, storage configuration, API mapping, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Strg1.01	• **Test description with test expectation:** The TOE shall allow authorised access to the selected stored-data class only through the specified TOE-supported path. The expected result is successful authorised use consistent with the vendor evidence. • **Test preparation:** Prepare the TOE, test User and one selected stored-data class from VE.Strg1.01. Prepare the authorised state and the relevant observation method or implementation evidence from VE.Strg1.02. • **Testing steps:** The tester records the TOE version, platform, selected stored-data class, authorised state, access path and evidence type. The tester creates or retrieves the selected stored data and accesses it through the authorised TOE path. • **Test result:** The actual result is the authorised-access behaviour: data available, data unavailable, access gated differently than specified, or evidence inconsistent. The result shall describe content category and operation outcome, not personal values or reusable artefacts. • **Assessment (pass/fail):** PASS if the selected stored data is available only through the specified authorised path and the evidence supports the access-control mapping. FAIL if authorised access cannot be demonstrated, is inconsistent with the evidence, or data is accessible without the required authorisation.
TE.Strg1.02	• **Test description with test expectation:** The TOE shall deny unauthorised access to the selected stored-data class through TOE-supported access paths. The expected result is authentication prompt, safe denial, hidden data or another documented safe response before stored data is disclosed or used. • **Test preparation:** Use the stored data from TE.Strg1.01 and put the TOE in an unauthenticated, locked or otherwise unauthorised state from VE.Strg1.02. Identify the TOE-supported access path and expected denial outcome. • **Testing steps:** The tester records the TOE state, selected access path, expected denial outcome and evidence type. The tester attempts to access or use the selected stored data without satisfying the required authorisation condition. • **Test result:** The actual result is the unauthorised-access behaviour: prompt, denial, hidden data, safe redirect, disclosure or use. The result shall state whether the stored data became visible, exportable, usable or otherwise available. • **Assessment (pass/fail):** PASS if unauthorised access is blocked or safely gated before stored data is disclosed or used. FAIL if the selected stored data is visible, exportable, usable or processed as accessible in the unauthorised state.
TE.Strg1.03	• **Test description with test expectation:** The TOE shall prevent unauthorised access to selected stored data through supported export, sharing, diagnostic or equivalent local/platform paths. The expected result is denial, redaction, omission of stored data or another documented safe outcome. • **Test preparation:** Select one supported local/platform path from VE.Strg1.02 that could expose or transfer stored data. Use test data only and identify prohibited stored-data categories before execution.

Element	Content
	• **Testing steps:** The tester records the selected stored-data class, local/platform path, authorisation state, prohibited content categories and evidence type. The tester invokes the selected path in an unauthorised state or without the required authorisation. • **Test result:** The actual result is the local/platform-path behaviour: blocked, authorisation required, redacted, omitted, exposed or not evidenced. The result shall describe content categories and access decision, not personal data, secrets or reusable artefacts. • **Assessment (pass/fail):** PASS if the selected local/platform path does not disclose or transfer stored data without the required authorisation and the evidence is consistent. FAIL if stored data is exposed, exported, shared or diagnosed in reusable form contrary to VE.Strg1.01.
Test Verdict	PASS iff TE.Strg1.01, TE.Strg1.02 and TE.Strg1.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify stored-data classes, authorisation conditions, supported access paths, unauthorised states, expected access-control outcomes or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Strg2 (FITCEM PP: Local storage)

Element	Content
Local storage — AS.Strg2	**The TOE shall implement measures to maintain the confidentiality and integrity of the stored data, in particular personal data.**
Required vendor information	
VE.Strg2.01	The vendor shall identify stored-data classes in the evaluated configuration, including personal-data classes, and specify the confidentiality and integrity measures applied to each class. The vendor shall define which measures are observable through UI/platform-supported means and which require documentation, configuration or source-code evidence.
VE.Strg2.02	The vendor shall provide SUG/test instructions, test accounts, test data, supported observation methods and one conformance-level integrity-negative scenario. Implementation evidence may include storage design, data-flow mapping, protected-storage configuration, keystore/API mapping, integrity-check description, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Strg2.01	• **Test description with test expectation:** The TOE shall store the selected data class under the vendor-defined confidentiality and integrity measures. The expected result is an observable or sufficiently evidenced protected-storage state for the evaluated build. • **Test preparation:** Prepare the TOE, authenticated test User and one selected stored personal-data class from VE.Strg2.01. Prepare the relevant observation method or implementation evidence from VE.Strg2.02. • **Testing steps:** The tester records the TOE version, platform, selected data class, claimed protection measures, observation method and evidence type. The tester creates the selected data through a normal TOE flow and verifies the protected-storage state using the approved method or evidence. • **Test result:** The actual result is the protected-storage state: present, absent, ambiguous, inconsistent or evidenced only. The result shall describe protection categories and operation outcome without copying personal data, keys or protected object contents. • **Assessment (pass/fail):** PASS if the selected data class is stored under the defined protection measures and the evidence is consistent with the evaluated build. FAIL if the protected-storage state is missing, inconsistent, outside the evaluated configuration or not sufficiently evidenced.
TE.Strg2.02	• **Test description with test expectation:** The TOE shall not expose selected stored personal data in clear or directly reusable form through the selected supported observation path. The expected result is absence, redaction, protected representation or another documented safe representation. • **Test preparation:** Use the stored personal-data class from TE.Strg2.01 and select one supported confidentiality observation path from VE.Strg2.02. Define prohibited content categories before execution and use only test data. • **Testing steps:** The tester records the selected data class, observation path, prohibited content categories and evidence type. The tester invokes the approved path, such as supported export, diagnostic view, test storage inspector or equivalent vendor-approved method.

Element	Content
	• **Test result:** The actual result is the confidentiality behaviour: absent, redacted, protected, metadata-only, exposed, reusable, or evidenced only. The result shall describe representation type, not personal values, secrets or reusable artefacts. • **Assessment (pass/fail):** PASS if stored personal data is not exposed in clear or directly reusable form and non-observable protection is supported by sufficient evidence. FAIL if personal data or a reusable equivalent is visible, exportable, recoverable through the supported path, or claimed protection lacks adequate evidence.
TE.Strg2.03	• **Test description with test expectation:** The TOE shall detect or safely handle the selected stored-data integrity failure. The expected result is rejection, recovery, reinitialisation, re-issuance requirement, user notification or another documented safe state before affected data is used. • **Test preparation:** Select one stored-data class and one supported integrity-negative scenario from VE.Strg2.02. Prepare the expected safe outcome and corresponding observation method or implementation evidence. • **Testing steps:** The tester records the selected data class, integrity-negative scenario, expected safe outcome and evidence type. The tester applies the vendor-supported scenario and attempts to use the affected stored data through the normal TOE path. • **Test result:** The actual result is the integrity handling: rejected, recovered, reinitialised, re-issued, error-reported, used as valid, or not evidenced. The result shall describe the safe state or operation outcome without copying personal data or integrity values. • **Assessment (pass/fail):** PASS if the TOE does not use integrity-invalid or inconsistent stored data as valid and follows the documented safe handling path. FAIL if the TOE silently accepts, displays, presents, exports or otherwise relies on affected data as intact, or if claimed integrity handling lacks adequate evidence.
Test Verdict	PASS iff TE.Strg2.01, TE.Strg2.02 and TE.Strg2.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify stored-data classes, personal-data classes, confidentiality measures, integrity measures, supported observation methods, integrity-negative scenarios or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Strg3 (FITCEM PP: Local storage)

Element	Content
Local storage — AS.Strg3	**The TOE shall prevent users from backing up stored app data. It should also be impossible to transfer application data directly between phones.**
Required vendor information	
VE.Strg3.01	The vendor shall identify stored app data in the evaluated configuration and specify which backup, restore, export, migration and device-transfer mechanisms are blocked, disabled, excluded or safely handled by the TOE. The vendor shall define observable outcomes for backup attempt, restore attempt and direct phone-to-phone transfer attempt.
VE.Strg3.02	The vendor shall provide SUG/test instructions, test accounts, test data, supported devices and platform configurations allowing the tester to attempt backup and direct phone-to-phone transfer through supported Android/iOS mechanisms. The method shall avoid rooting/jailbreaking, storage forensics, reverse engineering or uncontrolled extraction of application files.
Required test procedures	
TE.Strg3.01	• **Test description with test expectation:** The TOE shall prevent backup of selected stored app data through the supported backup path for the evaluated configuration. The expected result is that backup is unavailable, excludes TOE app data, or produces a state that cannot restore stored app data as usable wallet data. • **Test preparation:** Prepare the TOE with one authenticated test User and selected stored app data created through a normal wallet flow. Identify the platform/application backup path and expected blocked or excluded outcome from VE.Strg3.01. • **Testing steps:** The tester records the TOE version, platform, OS version, selected stored app data class, selected backup path and expected backup-prevention outcome in the test evidence. The tester

Element	Content
	attempts the supported backup path and observes the TOE UI, platform backup status, backup configuration, test-backup result or approved diagnostic output. • **Test result:** The actual result is the observed backup behaviour: the backup path is blocked, disabled or excludes TOE app data, or it backs up stored app data contrary to the specified protection. The result shall describe the observed status and content category without copying personal data, secrets or reusable artefacts. • **Assessment (pass/fail):** PASS if the selected stored app data is not backed up in a usable or restorable form through the tested supported backup path. FAIL if the tested backup path stores TOE app data so that it can later be restored as usable wallet application data.
TE.Strg3.02	• **Test description with test expectation:** The TOE shall prevent direct transfer of application data from one phone to another through the supported device-transfer scenario. The expected result is that the second device does not receive usable stored app data and must follow the vendor-defined activation, registration or recovery flow. • **Test preparation:** Prepare Device A with an activated TOE instance and selected stored app data. Prepare Device B as a fresh supported device and identify the direct phone-to-phone transfer method from VE.Strg3.02. • **Testing steps:** The tester records the TOE version, platforms, OS versions, Device A/Device B test identifiers, selected stored app data class, transfer method and expected transfer-prevention outcome in the test evidence. The tester performs the supported direct transfer scenario and then opens the TOE on Device B to observe whether the transferred data is present or usable. • **Test result:** The actual result is the observed transfer behaviour: Device B starts as fresh/unbound, requires activation or recovery, contains no usable transferred wallet state, or contains usable application data from Device A. The result shall describe the visible lifecycle state and data-availability category, not personal data or secrets. • **Assessment (pass/fail):** PASS if Device B cannot use application data transferred directly from Device A and must complete the specified secure setup/recovery path. FAIL if Device B receives and uses stored app data directly transferred from Device A without the required secure lifecycle steps.
TE.Strg3.03	• **Test description with test expectation:** After a backup/restore or device-transfer attempt, the TOE shall not accept restored or transferred stored app data as an active valid wallet state unless the vendor-defined secure recovery or activation flow is completed. The expected result is safe reset, reactivation requirement, re-binding requirement, or another documented safe state. • **Test preparation:** Use the backup or direct-transfer scenario from TE.Strg3.01 or TE.Strg3.02. Identify one wallet operation that would prove whether restored/transferred stored app data is usable. • **Testing steps:** The tester records the TOE version, platform, restore/transfer scenario, selected post-restore wallet operation and expected safe state in the test evidence. The tester attempts to restore or use the resulting application state and then attempts the selected wallet operation before completing any required secure recovery or activation flow. • **Test result:** The actual result is the observed post-restore/post-transfer behaviour: the TOE remains fresh, locked, unbound, deactivated or safely recoverable, or it accepts the restored/transferred state as valid. The result shall describe the visible lifecycle state and whether the selected wallet operation completed. • **Assessment (pass/fail):** PASS if restored or transferred app data is not accepted as a valid active wallet state before the required secure recovery or activation flow. FAIL if the TOE permits wallet operation using restored or directly transferred application data contrary to VE.Strg3.01.
Test Verdict	PASS iff TE.Strg3.01, TE.Strg3.02 and TE.Strg3.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify stored app data, supported backup/transfer paths, expected safe outcomes or observation methods.

Table — VE and TE of AS.Strg4 (FITCEM PP: Local storage)

Element	Content
Local storage — AS.Strg4	**The TOE shall ensure that PID credentials and retrieved attestations of attributes (EEAs) containing personal data are encrypted.**
Required vendor information	

Element	Content
VE.Strg4.01	The vendor shall identify all PID credentials and retrieved attestations of attributes containing personal data that can be stored by the TOE in the evaluated configuration. The vendor shall specify the encryption mechanism, encrypted storage location/category, key source, authorised use path, and which evidence is UI/platform-observable or requires documentation, configuration or source-code evidence.
VE.Strg4.02	The vendor shall provide SUG/test instructions, test PID credentials, test attestations, test accounts, supported observation methods and one conformance-level negative encryption-context scenario. Implementation evidence may include storage design, Crypto-Annex reference, encryption/API mapping, keystore or WSCA-interface mapping, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Strg4.01	• **Test description with test expectation:** The TOE shall store the selected PID credential containing personal data in encrypted form. The expected result is an observable or sufficiently evidenced encrypted storage state for the evaluated build. • **Test preparation:** Prepare the TOE, authenticated test User, test PID Provider or issuance endpoint, and one test PID credential containing non-real personal data. Prepare the approved observation method or implementation evidence from VE.Strg4.01 and VE.Strg4.02. • **Testing steps:** The tester records the TOE version, platform, selected PID credential type, encryption mechanism category, observation method and evidence type. The tester obtains or imports the test PID credential, verifies the encrypted storage state, and uses the PID through an authorised TOE path. • **Test result:** The actual result is the PID-storage behaviour: encrypted/evidenced, ambiguous, inconsistent, clear/reusable, or unusable through the authorised path. The result shall describe representation categories without copying personal data, keys or reusable artefacts. • **Assessment (pass/fail):** PASS if the selected PID credential is stored encrypted and remains usable only through the authorised TOE path. FAIL if it is stored or observable in clear/reusable form, or if the encrypted storage state lacks adequate evidence.
TE.Strg4.02	• **Test description with test expectation:** The TOE shall store the selected retrieved attestation containing personal data in encrypted form. The expected result is an observable or sufficiently evidenced encrypted storage state for the evaluated build. • **Test preparation:** Prepare the TOE, authenticated test User, test Attestation Provider or retrieval endpoint, and one test attestation containing non-real personal data. Identify the applicable observation method and authorised attestation-use flow. • **Testing steps:** The tester records the TOE version, platform, selected attestation type, encryption mechanism category, observation method and evidence type. The tester retrieves the attestation, verifies the encrypted storage state, and uses it through an authorised TOE function. • **Test result:** The actual result is the attestation-storage behaviour: encrypted/evidenced, ambiguous, inconsistent, clear/reusable, or unusable through the authorised path. The result shall describe representation categories without copying personal data or reusable credential material. • **Assessment (pass/fail):** PASS if the selected retrieved attestation is stored encrypted and remains usable only through the authorised TOE path. FAIL if it is stored or observable in clear/reusable form, or if the encrypted storage state lacks adequate evidence.
TE.Strg4.03	• **Test description with test expectation:** The TOE shall not expose or use PID credentials or retrieved attestations as clear usable data when the required encryption/decryption context is unavailable or invalid. The expected result is safe rejection, re-authentication, recovery, re-issuance requirement or another documented safe state. • **Test preparation:** Select one encrypted object from TE.Strg4.01 or TE.Strg4.02 and one vendor-supported negative encryption-context scenario from VE.Strg4.02. Prepare the expected safe outcome and observation method or implementation evidence. • **Testing steps:** The tester records the selected object type, negative scenario, expected safe outcome and evidence type. The tester applies the vendor-supported scenario and attempts to view, present, export or otherwise use the affected object through the normal TOE path. • **Test result:** The actual result is the encryption-context handling: rejected, re-authenticated, recovered, re-issued, safe state, clear exposure, or use as valid data. The result shall describe the safe-state or operation outcome without copying protected object contents.

Element	Content
	• **Assessment (pass/fail):** PASS if the TOE does not expose or use the affected object as clear valid data and follows the documented safe handling path. FAIL if the TOE falls back to clear storage, discloses contents, or processes the affected object as valid despite the invalid encryption context.
Test Verdict	PASS iff TE.Strg4.01, TE.Strg4.02 and TE.Strg4.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify applicable PID credentials, retrieved attestations, encryption mechanisms, key source, encrypted storage location/category, observation methods, authorised use paths or negative encryption-context scenarios.

NET — Network & Interfaces

Table — VE and TE of AS.Net1 (FITCEM PP: Network & Interfaces)

Element	Content
Network & Interfaces — AS.Net1	**Wallet Instance shall verify the authenticity, trustworthiness and authorisation of its communication partners.**
Required vendor information	
VE.Net1.01	The vendor shall identify all communication partner classes in the evaluated configuration and specify how the Wallet Instance verifies authenticity, trustworthiness and authorisation for each class. The vendor shall define which verification outcomes are observable through UI/protocol/test-endpoint means and which require documentation, configuration or source-code evidence.
VE.Net1.02	The vendor shall provide SUG/test instructions, test endpoints, test partner identities, trust-status configuration, authorisation policy and conformance-level negative scenarios. Implementation evidence may include trust-policy mapping, partner registry metadata, certificate/trust-source configuration, decision tables, API mapping, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Net1.01	• **Test description with test expectation:** The Wallet Instance shall accept the selected valid communication partner only after authenticity, trustworthiness and authorisation are satisfied. The expected result is successful completion of the selected protected flow with the valid partner. • **Test preparation:** Prepare the TOE, test User, one communication partner class and one valid partner scenario from VE.Net1.01 and VE.Net1.02. Identify the expected verification evidence and protected wallet operation. • **Testing steps:** The tester records the TOE version, platform, partner class, selected operation, expected verification conditions and evidence type. The tester reviews the evidence and initiates the selected flow with the valid partner. • **Test result:** The actual result is the valid-partner behaviour: accepted, rejected, partially accepted, or inconsistent with evidence. The result shall state whether the selected wallet operation completed. • **Assessment (pass/fail):** PASS if the TOE completes the selected flow only for the partner satisfying the specified authenticity, trustworthiness and authorisation conditions. FAIL if the valid partner cannot be accepted without documented reason or if the evidence does not support the verified state.
TE.Net1.02	• **Test description with test expectation:** The Wallet Instance shall reject or safely handle a selected unauthentic or untrusted communication partner. The expected result is safe rejection, trusted-source error, reconfiguration requirement or another documented safe outcome before protected wallet data or operations are made available. • **Test preparation:** Select one partner class and prepare one unauthentic or untrusted partner scenario from VE.Net1.02. Define the protected flow that must not proceed. • **Testing steps:** The tester records the selected partner class, negative partner condition, expected safe outcome and evidence type. The tester initiates the selected flow against the negative partner and observes UI, protocol/session metadata, test endpoint result or backend decision. • **Test result:** The actual result is the rejection behaviour: rejected, warned and stopped, safely retried, accepted, or not evidenced. The result shall state whether protected wallet data was transmitted, received, displayed or acted upon. • **Assessment (pass/fail):** PASS if the TOE does not continue the protected communication or operation with the unauthentic or untrusted partner. FAIL if the TOE accepts the partner or completes the selected flow despite failed authenticity or trustworthiness verification.
TE.Net1.03	• **Test description with test expectation:** The Wallet Instance shall reject or safely handle a communication partner that is not authorised for the selected operation, role, purpose or data request. The expected result is denial, limited disclosure, re-prompt with corrected context or another documented safe outcome. • **Test preparation:** Select one partner class and one operation for which authorisation is relevant. Prepare the vendor-supported unauthorised-partner or wrong-purpose scenario from VE.Net1.02. • **Testing steps:** The tester records the selected partner class, operation, authorisation condition, expected safe outcome and evidence type. The tester initiates the selected operation under the

Element	Content
	unauthorised or wrong-purpose condition and observes UI, test endpoint, backend decision or log/evidence outcome. • **Test result:** The actual result is the authorisation behaviour: denied, disclosure-limited, re-prompted, accepted, or not evidenced. The result shall state whether any wallet data or protected operation became available. • **Assessment (pass/fail):** PASS if the TOE does not complete the selected operation for the unauthorised condition unless authorisation is first re-established. FAIL if the TOE ignores partner authorisation and completes the operation contrary to VE.Net1.01.
Test Verdict	PASS iff TE.Net1.01, TE.Net1.02 and TE.Net1.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify partner classes, authenticity criteria, trustworthiness criteria, authorisation criteria, observation methods or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Net2 (FITCEM PP: Network & Interfaces)

Element	Content
Network & Interfaces — AS.Net2	**Wallet Instance shall ensure the integrity of data and processes.**
Required vendor information	
VE.Net2.01	The vendor shall identify the data objects and process flows for which integrity is enforced by the Wallet Instance in the evaluated configuration. The vendor shall define valid, invalid, stale, mismatched and interrupted integrity states, their expected outcomes, and which controls are observable or require documentation, configuration or source-code evidence.
VE.Net2.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints, test data and conformance-level negative scenarios for selected data and process flows. Implementation evidence may include integrity model, state/sequence diagrams, data-binding description, decision tables, API mapping, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Net2.01	• **Test description with test expectation:** The Wallet Instance shall complete the selected data/process flow when the relevant integrity conditions are valid. The expected result is normal completion of the selected operation and a defined integrity-valid outcome. • **Test preparation:** Prepare the TOE, test User and test endpoint according to VE.Net2.02. Select one representative flow involving integrity-protected data and one process sequence for which integrity is relevant. • **Testing steps:** The tester records the TOE version, platform, selected flow, selected data object/process sequence, integrity condition, observation method and evidence type. The tester executes the selected flow using valid data and the expected process order. • **Test result:** The actual result is the integrity-valid behaviour: completed, rejected, ambiguous or inconsistent with evidence. The result shall describe operation status and integrity-state category without copying secrets, personal data or internal integrity values. • **Assessment (pass/fail):** PASS if the selected flow completes in the defined integrity-valid state and the evidence supports the integrity model. FAIL if valid integrity conditions cannot be processed without documented reason or the resulting state is ambiguous or inconsistent.
TE.Net2.02	• **Test description with test expectation:** The Wallet Instance shall reject or safely handle selected data whose integrity is invalid, stale or mismatched. The expected result is safe rejection, revalidation, re-fetch, re-prompting or another documented safe outcome before the data is used. • **Test preparation:** Select one integrity-protected data object from VE.Net2.01 and one vendor-supported data-integrity negative scenario from VE.Net2.02. The scenario may use modified test data, stale data, mismatched transaction context, invalid status or equivalent controlled failure. • **Testing steps:** The tester records the selected data object, negative integrity scenario, expected safe outcome, observation method and evidence type. The tester executes the scenario through the approved test setup and observes UI, endpoint/backend decision, log/evidence entry or diagnostic output.

Element	Content
	• **Test result:** The actual result is the data-integrity handling: rejected, revalidated, retried, safe failure, accepted as valid, or not evidenced. The result shall state whether the affected data was displayed, stored, transmitted, presented or otherwise acted upon. • **Assessment (pass/fail):** PASS if integrity-invalid, stale or mismatched data is not processed as valid and the TOE follows the documented safe outcome. FAIL if the TOE silently accepts, displays, transmits, presents or otherwise relies on data whose integrity condition failed.
TE.Net2.03	• **Test description with test expectation:** The Wallet Instance shall reject or safely handle a selected process flow whose integrity is invalid because the expected sequence, context or state binding is broken. The expected result is no completion of the security-relevant process under the invalid sequence or context. • **Test preparation:** Select one process flow from VE.Net2.01 and one vendor-supported process-integrity negative scenario from VE.Net2.02. The scenario shall not require exploit development, uncontrolled protocol tampering, reverse engineering or broad fuzzing. • **Testing steps:** The tester records the selected process flow, sequence/context failure, expected safe outcome and evidence type. The tester triggers the process-integrity negative scenario and observes UI, endpoint/backend decision, transaction state, log/evidence entry or diagnostic output. • **Test result:** The actual result is the process-integrity handling: blocked, reinitialised, re-prompted, safe failure, completed despite invalid context, or not evidenced. The result shall describe the visible process state and whether the security-relevant operation completed. • **Assessment (pass/fail):** PASS if the TOE does not complete the selected process under an invalid sequence, broken context or inconsistent state. FAIL if the TOE completes or authorizes the operation despite the process-integrity failure.
Test Verdict	PASS iff TE.Net2.01, TE.Net2.02 and TE.Net2.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify integrity-protected data, integrity-relevant processes, valid/invalid integrity states, expected safe outcomes, observation methods or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Net3 (FITCEM PP: Network & Interfaces)

Element	Content
Network & Interfaces — AS.Net3	**The Wallet Instance shall communicate with the WSCA via an end-to-end secure channel.**
Required vendor information	
VE.Net3.01	The vendor shall specify the WI–WSCA secure-channel model, including the channel endpoints, protected data categories, channel establishment conditions and observable secure-channel states. The specification shall define what “end-to-end” means when any Wallet Provider backend, relay, gateway or intermediary is present.
VE.Net3.02	The vendor shall provide SUG/test instructions, test WSCA or WSCA test environment, test accounts and conformance-level positive and negative channel scenarios. The method shall allow the tester to observe channel establishment, channel failure and WSCA operation outcome without exposing cryptographic secrets or private key material.
Required test procedures	
TE.Net3.01	• **Test description with test expectation:** The Wallet Instance shall establish the vendor-defined end-to-end secure channel before communicating with the WSCA for the selected WSCA-dependent operation. The expected result is successful operation only after the secure-channel state is established. • **Test preparation:** Prepare the TOE, authenticated test User where required, and WSCA test environment according to VE.Net3.02. Select one representative WSCA-dependent operation and identify the expected secure-channel state from VE.Net3.01. • **Testing steps:** The tester records the TOE version, platform, OS version, selected WSCA-dependent operation, WSCA test environment, secure-channel model and expected secure-channel state in the test evidence. The tester initiates the selected operation and observes the TOE UI, protocol/session metadata, WSCA test endpoint result or approved diagnostic output.

Element	Content
	• **Test result:** The actual result is the observed channel-establishment behaviour: the secure channel is established and the WSCA-dependent operation completes, the channel is not established and the operation is blocked, or the operation completes without observable secure-channel establishment. The result shall describe the visible channel state and operation outcome without recording secrets, private key material or protected payload values. • **Assessment (pass/fail):** PASS if the selected WSCA-dependent operation completes only after the end-to-end secure channel is established as specified. FAIL if the operation proceeds without the required secure-channel state or if the secure-channel state cannot be determined by the approved method.
TE.Net3.02	• **Test description with test expectation:** The Wallet Instance shall not communicate with the WSCA for the selected protected operation when the end-to-end secure channel cannot be established. The expected result is safe rejection, retry, user-visible error, or another documented safe outcome before WSCA communication is used for the operation. • **Test preparation:** Select the same WSCA-dependent operation as in TE.Net3.01 and prepare the vendor-supported secure-channel-failure scenario from VE.Net3.02. Define the expected safe outcome before execution. • **Testing steps:** The tester records the TOE version, platform, selected operation, selected channel-failure scenario and expected safe outcome in the test evidence. The tester triggers the channel-failure scenario and observes the TOE UI, WSCA test endpoint result, protocol/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed failure handling: the TOE blocks WSCA communication, retries safely, reports a channel error, or continues the WSCA-dependent operation despite failed channel establishment. The result shall state whether any protected WSCA request or operation was accepted or completed. • **Assessment (pass/fail):** PASS if the TOE does not complete the selected WSCA-dependent operation when the end-to-end secure channel is not established. FAIL if the TOE communicates with WSCA or completes the operation despite the failed secure-channel condition.
TE.Net3.03	• **Test description with test expectation:** Where any intermediary is present, the WI–WSCA communication shall remain bound to the WSCA-side endpoint of the secure channel. The expected result is that the intermediary cannot be accepted as the final WSCA channel endpoint for the selected WSCA-dependent operation. • **Test preparation:** Identify from VE.Net3.01 whether the evaluated architecture uses a backend, relay, gateway or equivalent intermediary between WI and WSCA. Prepare the vendor-supported endpoint-substitution or relay-only scenario, or document non-applicability if no intermediary exists. • **Testing steps:** The tester records the TOE version, platform, selected architecture path, intermediary role if applicable, selected endpoint-substitution/relay-only scenario and expected safe outcome in the test evidence. The tester executes the scenario through the approved test setup and observes the TOE UI, WSCA test endpoint, intermediary/test backend and approved diagnostic output. • **Test result:** The actual result is the observed end-to-end handling: the TOE communicates only when the WSCA-side channel endpoint is valid, rejects relay-only or substituted endpoint handling, or accepts the intermediary as if it were the WSCA endpoint. The result shall describe endpoint-role categories and operation outcome without recording protected payload or channel secrets. • **Assessment (pass/fail):** PASS if the TOE does not treat an intermediary or substituted endpoint as the final WSCA secure-channel endpoint for the tested operation. FAIL if WSCA-dependent communication completes through a channel that is not end-to-end to the WSCA endpoint as specified.
Test Verdict	PASS iff TE.Net3.01, TE.Net3.02 and TE.Net3.03 all PASS, with TE.Net3.03 documented as not applicable only where no intermediary or endpoint-substitution scenario exists. FAIL if any applicable TE fails or if VE evidence is insufficient to identify the WI–WSCA channel endpoints, secure-channel states, negative scenarios or observation methods.

Table — VE and TE of AS.Net4 (FITCEM PP: Network & Interfaces)

Element	Content
Network & Interfaces — AS.Net4	**The channel between the Wallet Instance and the WSCA shall ensure session binding and freshness.**

Element	Content
Required vendor information	
VE.Net4.01	The vendor shall specify the WI–WSCA session model, including session creation, session termination, binding to the relevant Wallet Instance operation, and freshness conditions. The vendor shall define which session-binding and freshness outcomes are observable through UI/test-endpoint/protocol metadata and which require documentation, configuration or source-code evidence.
VE.Net4.02	The vendor shall provide SUG/test instructions, WSCA test environment or simulator, test accounts, test data and conformance-level mismatch, stale and replay scenarios. Implementation evidence may include protocol/session model, state/sequence diagrams, nonce/challenge/timestamp binding description, API mapping, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Net4.01	• **Test description with test expectation:** The WI–WSCA channel shall establish a fresh session bound to the selected WSCA-dependent operation. The expected result is completion of the operation only under the valid current session context. • **Test preparation:** Prepare the TOE, authenticated test User where required, and WSCA test environment according to VE.Net4.02. Select one WSCA-dependent operation and identify the expected binding and freshness evidence. • **Testing steps:** The tester records the TOE version, platform, selected operation, WSCA test environment, expected binding state, freshness condition and evidence type. The tester executes the selected operation and observes UI, test endpoint result, protocol/session metadata, diagnostic output or implementation evidence. • **Test result:** The actual result is the session-valid behaviour: completed in fresh bound session, rejected, ambiguous, or inconsistent with evidence. The result shall not record session secrets, nonces, tokens or protected payload values. • **Assessment (pass/fail):** PASS if the selected operation completes only in the valid fresh bound session state and the evidence supports that state. FAIL if the operation completes without adequate binding/freshness evidence or if the valid state cannot be determined by the approved method.
TE.Net4.02	• **Test description with test expectation:** The TOE shall reject or safely handle WI–WSCA session context that is bound to a different operation, transaction or session. The expected result is safe rejection, re-binding, re-authentication or new-session creation before the operation proceeds. • **Test preparation:** Select one WSCA-dependent operation and one vendor-supported mismatched session/operation scenario from VE.Net4.02. Prepare the expected safe outcome and the corresponding observation method or evidence. • **Testing steps:** The tester records the selected operation, mismatch condition, expected safe outcome and evidence type. The tester executes the approved mismatch scenario and observes UI, WSCA test endpoint result, session metadata, diagnostic output or implementation evidence. • **Test result:** The actual result is the binding-mismatch handling: rejected, re-bound, re-authenticated, new session created, accepted, or not evidenced. The result shall state whether any WSCA-dependent operation completed under the mismatched context. • **Assessment (pass/fail):** PASS if the TOE does not complete the selected operation under a mismatched session, operation or transaction context. FAIL if the TOE accepts session artefacts or decisions from one context to complete another context.
TE.Net4.03	• **Test description with test expectation:** The TOE shall reject or safely handle stale, expired or replayed WI–WSCA session context. The expected result is safe rejection, fresh-context creation, retry with fresh session or another documented safe outcome. • **Test preparation:** Select one freshness-protected WSCA-dependent operation and one vendor-supported stale/replay scenario from VE.Net4.02. Define the freshness window, expiry condition or replay condition according to VE.Net4.01. • **Testing steps:** The tester records the selected operation, stale/replay scenario, freshness condition, expected safe outcome and evidence type. The tester obtains a valid context through the approved setup, then attempts the vendor-supported stale/replay use and observes the TOE and WSCA test endpoint result.

Element	Content
	• **Test result:** The actual result is the freshness handling: rejected, fresh context requested, safe retry, accepted as fresh, or not evidenced. The result shall state whether the WSCA-dependent operation completed using stale, expired or replayed context. • **Assessment (pass/fail):** PASS if stale, expired or replayed context is not accepted as fresh and the TOE follows the documented safe handling path. FAIL if the TOE completes the selected operation using stale, expired or replayed session context.
Test Verdict	PASS iff TE.Net4.01, TE.Net4.02 and TE.Net4.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the WI–WSCA session model, binding context, freshness condition, mismatch scenario, stale/replay scenario, observation method or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Net5 (FITCEM PP: Network & Interfaces)

Element	Content
Network & Interfaces — AS.Net5	**The WSCA interface (SCI) shall be secured with TLSv1.3. The TLS connections shall utilize strong cipher suites.**
Required vendor information	
VE.Net5.01	The vendor shall specify the SCI TLS profile used between the Wallet Instance and WSCA, including endpoint roles, TLS version, allowed cipher suites, certificate/trust configuration and observable connection outcomes. The vendor shall define which cipher suites are considered strong for the evaluated configuration.
VE.Net5.02	The vendor shall provide SUG/test instructions, WSCA test environment or simulator, test accounts and conformance-level TLS scenarios allowing the tester to observe accepted and rejected SCI connections. The method shall allow observation of TLS version and negotiated cipher-suite category without exposing private keys, session secrets or protected payloads.
Required test procedures	
TE.Net5.01	• **Test description with test expectation:** The SCI connection between the Wallet Instance and WSCA shall be established using TLSv1.3 and an allowed strong cipher suite. The expected result is successful SCI communication only after the required TLS profile is negotiated. • **Test preparation:** Prepare the TOE, authenticated test User where required, and WSCA test endpoint according to VE.Net5.02. Select one representative SCI operation and identify the expected TLSv1.3/cipher-suite outcome from VE.Net5.01. • **Testing steps:** The tester records the TOE version, platform, OS version, selected SCI operation, WSCA endpoint, expected TLS version, expected cipher-suite category and approved observation method in the test evidence. The tester initiates the selected SCI operation and observes the TOE UI, SCI endpoint result, TLS/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed TLS negotiation and SCI behaviour: the SCI connection is established with TLSv1.3 and an allowed strong cipher suite, or the operation is blocked, or the operation completes without observable TLSv1.3/strong-cipher evidence. The result shall describe the TLS version category, cipher-suite category and SCI operation outcome without recording session secrets or protected payloads. • **Assessment (pass/fail):** PASS if the selected SCI operation completes only over TLSv1.3 with an allowed strong cipher suite. FAIL if the operation completes without TLSv1.3, with a non-allowed cipher suite, or if the required TLS profile cannot be observed by the approved method.
TE.Net5.02	• **Test description with test expectation:** The SCI shall not establish normal WSCA communication using TLS versions other than TLSv1.3. The expected result is connection failure, safe rejection, retry with TLSv1.3, or another documented safe outcome before the SCI operation is completed. • **Test preparation:** Configure the WSCA test endpoint or simulator to offer only a non-TLSv1.3 protocol version, according to VE.Net5.02. Use the same or equivalent SCI operation as in TE.Net5.01. • **Testing steps:** The tester records the TOE version, platform, selected SCI operation, non-TLSv1.3 scenario and expected safe outcome in the test evidence. The tester initiates the SCI operation against the negative endpoint and observes the TOE UI, SCI endpoint result, TLS/session metadata or approved diagnostic output.

Element	Content
	• **Test result:** The actual result is the observed protocol-version handling: the TOE rejects the connection, retries only with TLSv1.3, reports a safe channel error, or completes SCI communication using the non-TLSv1.3 version. The result shall state whether any WSCA-dependent operation was accepted or completed. • **Assessment (pass/fail):** PASS if the TOE does not complete SCI communication using a TLS version other than TLSv1.3. FAIL if the TOE accepts, negotiates or completes the selected SCI operation over a non-TLSv1.3 channel.
TE.Net5.03	• **Test description with test expectation:** The SCI shall not complete WSCA communication when no allowed strong TLS cipher suite can be negotiated. The expected result is connection failure, safe rejection, or another documented safe outcome before protected SCI data or operation is exchanged. • **Test preparation:** Configure the WSCA test endpoint or simulator to offer no cipher suite allowed by VE.Net5.01. Ensure that the test does not require exposure of private keys, TLS session secrets or protected payload values. • **Testing steps:** The tester records the TOE version, platform, selected SCI operation, disallowed-cipher scenario, expected safe outcome and observation method in the test evidence. The tester initiates the SCI operation and observes the TOE UI, SCI endpoint result, TLS/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed cipher-suite handling: the TOE rejects the connection, reports a safe channel error, or completes SCI communication despite absence of an allowed strong cipher suite. The result shall describe the negotiated or rejected cipher-suite category and whether any SCI operation completed. • **Assessment (pass/fail):** PASS if the TOE does not complete SCI communication unless an allowed strong cipher suite is negotiated. FAIL if the TOE completes the selected SCI operation with a disallowed or weak cipher-suite category.
Test Verdict	PASS iff TE.Net5.01, TE.Net5.02 and TE.Net5.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the SCI TLS profile, allowed strong cipher suites, negative TLS scenarios or observation methods.

Table — VE and TE of AS.Net6 (FITCEM PP: Network & Interfaces)

Element	Content
Network & Interfaces — AS.Net6	**The TOE shall establish and maintain secure communication channels using TLSv1.3 with all Wallet Provider backend services.**
Required vendor information	
VE.Net6.01	The vendor shall identify all Wallet Provider backend services contacted by the TOE in the evaluated configuration and specify the TLSv1.3 communication profile for each service. The specification shall define endpoint roles, connection triggers, allowed backend endpoints, certificate/trust configuration and observable connection outcomes.
VE.Net6.02	The vendor shall provide SUG/test instructions, test backend services or backend simulators, test accounts and conformance-level positive and negative TLS scenarios. The method shall allow observation of TLS version, connection establishment, channel interruption and operation outcome without exposing private keys, session secrets, tokens or protected payloads.
Required test procedures	
TE.Net6.01	• **Test description with test expectation:** The TOE shall establish secure communication channels using TLSv1.3 with the selected Wallet Provider backend services. The expected result is successful backend-dependent operation only after a TLSv1.3 channel is established. • **Test preparation:** Prepare the TOE, test User where required, and Wallet Provider test backend services according to VE.Net6.02. Select backend services from the complete service inventory in VE.Net6.01, covering each distinct backend-service class in the evaluated configuration. • **Testing steps:** The tester records the TOE version, platform, OS version, selected backend services, selected backend-dependent operations, expected TLS version and approved observation method in the test evidence. The tester initiates each selected operation and observes the TOE UI, backend endpoint result, TLS/session metadata or approved diagnostic output.

Element	Content
	• **Test result:** The actual result is the observed backend-channel behaviour: the TOE establishes TLSv1.3 and completes the backend-dependent operation, blocks the operation, or completes without observable TLSv1.3 evidence. The result shall describe the TLS version category and operation outcome without recording secrets, tokens or protected payload values. • **Assessment (pass/fail):** PASS if each tested backend-dependent operation completes only over TLSv1.3 with the corresponding Wallet Provider backend service. FAIL if any tested backend service is contacted or used without TLSv1.3, or if the TLSv1.3 state cannot be observed by the approved method.
TE.Net6.02	• **Test description with test expectation:** The TOE shall not complete communication with a Wallet Provider backend service using a TLS version other than TLSv1.3. The expected result is connection failure, safe rejection, retry with TLSv1.3, or another documented safe outcome before the backend-dependent operation is completed. • **Test preparation:** Select one backend service and one backend-dependent operation from TE.Net6.01. Configure the corresponding test backend or simulator to offer only a non-TLSv1.3 protocol version according to VE.Net6.02. • **Testing steps:** The tester records the TOE version, platform, selected backend service, selected operation, non-TLSv1.3 scenario and expected safe outcome in the test evidence. The tester initiates the selected operation against the negative backend configuration and observes the TOE UI, backend endpoint result, TLS/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed protocol-version handling: the TOE rejects the connection, retries only with TLSv1.3, reports a safe channel error, or completes backend communication using the non-TLSv1.3 version. The result shall state whether the backend-dependent operation was accepted or completed. • **Assessment (pass/fail):** PASS if the TOE does not complete backend communication using a TLS version other than TLSv1.3. FAIL if the TOE accepts, negotiates or completes the selected backend-dependent operation over a non-TLSv1.3 channel.
TE.Net6.03	• **Test description with test expectation:** The TOE shall maintain backend communication only while the TLSv1.3 channel remains valid, or re-establish TLSv1.3 before continuing after channel loss. The expected result is safe interruption, retry, re-establishment or documented failure, not continuation over an invalid or non-secure channel. • **Test preparation:** Select one backend service and operation requiring a maintained backend channel. Prepare the vendor-supported channel-interruption or reconnect scenario from VE.Net6.02 and define the expected safe outcome. • **Testing steps:** The tester records the TOE version, platform, selected backend service, selected operation, channel-interruption scenario and expected safe outcome in the test evidence. The tester starts the operation, triggers the approved channel interruption or reconnect condition, and observes the TOE UI, backend result, TLS/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed channel-maintenance behaviour: the TOE blocks or pauses the operation, re-establishes TLSv1.3 before continuing, reports a safe error, or continues backend communication without a valid TLSv1.3 channel. The result shall describe the visible state and whether the operation completed after a valid TLSv1.3 channel was restored. • **Assessment (pass/fail):** PASS if the TOE does not continue backend communication after channel loss unless TLSv1.3 is re-established or the operation is safely terminated. FAIL if the TOE continues or completes the backend-dependent operation without a valid TLSv1.3 channel.
Test Verdict	PASS iff TE.Net6.01, TE.Net6.02 and TE.Net6.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify all Wallet Provider backend services, TLSv1.3 profile, negative TLS scenario, channel-maintenance scenario or observation method.

Table — VE and TE of AS.Net7 (FITCEM PP: Network & Interfaces)

Element	Content
Network & Interfaces — AS.Net7	**The PID Provider interface shall be secured with TLSv1.3.**
Required vendor information	

Element	Content
VE.Net7.01	The vendor shall identify the PID Provider interface used by the Wallet Instance in the evaluated configuration and specify the TLSv1.3 profile applied to this interface. The specification shall define endpoint roles, connection triggers, allowed PID Provider endpoints, certificate/trust configuration and observable connection outcomes.
VE.Net7.02	The vendor shall provide SUG/test instructions, PID Provider test endpoint or simulator, test accounts and conformance-level TLS scenarios allowing the tester to observe accepted and rejected PID Provider interface connections. The method shall allow observation of TLS version and issuance-flow outcome without exposing private keys, session secrets, tokens, PID contents or protected payload values.
Required test procedures	
TE.Net7.01	• **Test description with test expectation:** The PID Provider interface shall be established using TLSv1.3 before the Wallet Instance performs the selected PID issuance or retrieval operation. The expected result is successful PID Provider interface communication only after a TLSv1.3 channel is established. • **Test preparation:** Prepare the TOE, authenticated test User where required, and PID Provider test endpoint according to VE.Net7.02. Select one representative PID issuance or retrieval operation and identify the expected TLSv1.3 outcome from VE.Net7.01. • **Testing steps:** The tester records the TOE version, platform, OS version, selected PID Provider endpoint, selected PID operation, expected TLS version and approved observation method in the test evidence. The tester initiates the selected PID operation and observes the TOE UI, PID Provider test endpoint result, TLS/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed PID-interface behaviour: the TOE establishes TLSv1.3 and completes the PID operation, blocks the operation, or completes without observable TLSv1.3 evidence. The result shall describe the TLS version category and PID operation outcome without recording PID contents, secrets, tokens or protected payload values. • **Assessment (pass/fail):** PASS if the selected PID Provider operation completes only over TLSv1.3. FAIL if the TOE communicates with the PID Provider interface without TLSv1.3, or if TLSv1.3 use cannot be observed by the approved method.
TE.Net7.02	• **Test description with test expectation:** The PID Provider interface shall not complete communication using a TLS version other than TLSv1.3. The expected result is connection failure, safe rejection, retry with TLSv1.3, or another documented safe outcome before any PID issuance/retrieval operation is completed. • **Test preparation:** Configure the PID Provider test endpoint or simulator to offer only a non-TLSv1.3 protocol version according to VE.Net7.02. Use the same or equivalent PID operation as in TE.Net7.01. • **Testing steps:** The tester records the TOE version, platform, selected PID Provider endpoint, selected PID operation, non-TLSv1.3 scenario and expected safe outcome in the test evidence. The tester initiates the PID operation against the negative endpoint and observes the TOE UI, PID Provider endpoint result, TLS/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed protocol-version handling: the TOE rejects the connection, retries only with TLSv1.3, reports a safe channel error, or completes PID Provider communication using the non-TLSv1.3 version. The result shall state whether any PID issuance, retrieval or PID-related protected data exchange was accepted or completed. • **Assessment (pass/fail):** PASS if the TOE does not complete PID Provider interface communication using a TLS version other than TLSv1.3. FAIL if the TOE accepts, negotiates or completes the selected PID operation over a non-TLSv1.3 channel.
Test Verdict	PASS iff TE.Net7.01 and TE.Net7.02 both PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the PID Provider interface, TLSv1.3 profile, negative TLS scenario or observation method.

Table — VE and TE of AS.Net8 (FITCEM PP: Network & Interfaces)

Element	Content
Network & Interfaces — AS.Net8	**The TOE shall establish and maintain secure communication channels using TLSv1.2 or TLSv1.3 with services exposed by:• EAA (Electronic Attestations of Attributes (EAA) Provider,• Relying Party (RP),• Trusted List or LoTE Providers.The TOE shall ensure that the TLS certificate used to establish a connection with EAA, RP, TL/LoTE services are issued to these entities**

Element	Content
Required vendor information	
VE.Net8.01	The vendor shall identify the EAA Provider, Relying Party and Trusted List/LoTE service interfaces used by the TOE in the evaluated configuration. For each applicable service class, the vendor shall specify the TLS profile, allowed TLS versions, endpoint identity rules and observable connection outcomes.
VE.Net8.02	The vendor shall provide SUG/test instructions, test endpoints or simulators, test accounts and conformance-level positive and negative TLS/certificate scenarios. The method shall allow the tester to observe TLS version, endpoint identity verification and operation outcome without exposing private keys, session secrets, personal data or protected payloads.
Required test procedures	
TE.Net8.01	• **Test description with test expectation:** The TOE shall establish and maintain a TLSv1.2 or TLSv1.3 channel with each selected applicable service class: EAA Provider, RP, and Trusted List/LoTE Provider. The expected result is successful service-dependent operation only after an allowed TLS channel is established. • **Test preparation:** Prepare the TOE, test User where required, and test endpoints for applicable EAA, RP and Trusted List/LoTE service classes according to VE.Net8.02. Select one representative operation for each applicable service class and identify the expected TLS version outcome from VE.Net8.01. • **Testing steps:** The tester records the TOE version, platform, OS version, selected service class, selected test endpoint, selected operation, expected TLS version and approved observation method in the test evidence. The tester initiates each selected operation and observes the TOE UI, test endpoint result, TLS/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed channel behaviour: the TOE establishes TLSv1.2 or TLSv1.3 and completes the selected operation, blocks the operation, or completes without observable allowed-TLS evidence. The result shall describe the TLS version category and operation outcome without recording secrets, session material, personal data or protected payload values. • **Assessment (pass/fail):** PASS if each tested applicable service-class operation completes only over TLSv1.2 or TLSv1.3. FAIL if the TOE communicates with the selected EAA, RP or TL/LoTE service without an allowed TLS version, or if the allowed TLS state cannot be observed by the approved method.
TE.Net8.02	• **Test description with test expectation:** The TOE shall not complete communication with the selected EAA, RP or Trusted List/LoTE service if an allowed TLSv1.2/TLSv1.3 channel cannot be established or maintained. The expected result is connection failure, safe rejection, retry with an allowed TLS channel, or another documented safe outcome before the service-dependent operation is completed. • **Test preparation:** Select one applicable service class and operation from TE.Net8.01. Configure the corresponding test endpoint or scenario so that an allowed TLSv1.2/TLSv1.3 channel is unavailable, interrupted or replaced by a non-allowed protocol condition. • **Testing steps:** The tester records the TOE version, platform, selected service class, selected operation, negative TLS/channel scenario and expected safe outcome in the test evidence. The tester initiates the operation under the negative condition and observes the TOE UI, endpoint result, TLS/session metadata or approved diagnostic output. • **Test result:** The actual result is the observed channel-failure behaviour: the TOE rejects the connection, safely retries, reports a channel error, safely terminates the operation, or completes the operation without an allowed maintained TLS channel. The result shall state whether the service-dependent operation was accepted or completed. • **Assessment (pass/fail):** PASS if the TOE does not complete the selected operation unless an allowed TLSv1.2/TLSv1.3 channel is established or restored. FAIL if the TOE completes communication over a non-allowed, absent, interrupted or otherwise invalid channel.
TE.Net8.03	• **Test description with test expectation:** The TOE shall verify that the TLS certificate used for the selected service connection is issued to the expected EAA, RP or Trusted List/LoTE entity. The expected result is acceptance of the correctly issued certificate and rejection or safe handling of a certificate issued to a different or unauthorized entity. • **Test preparation:** Select one applicable service class and operation from VE.Net8.01. Prepare one valid certificate scenario and one vendor-supported wrong-entity certificate scenario for the same service class, using test certificates or approved test infrastructure. • **Testing steps:** The tester records the TOE version, platform, selected service class, selected entity identity rule, valid certificate scenario, wrong-entity certificate scenario and expected safe outcome in

Element	Content
	the test evidence. The tester executes the operation with the valid certificate scenario and then repeats it with the wrong-entity certificate scenario, observing TOE UI, endpoint result, TLS/certificate metadata or approved diagnostic output. • **Test result:** The actual result is the observed certificate-identity behaviour: the TOE accepts the certificate issued to the expected entity and rejects, warns, or safely stops when the certificate is issued to a different entity; or it accepts the wrong-entity certificate. The result shall describe identity-match categories and operation outcome without copying private keys, certificate secrets or protected payloads. • **Assessment (pass/fail):** PASS if the TOE accepts only certificates issued to the expected service entity for the selected connection and safely rejects the wrong-entity scenario. FAIL if the TOE completes the selected operation using a TLS certificate issued to a different or unauthorized entity.
Test Verdict	PASS iff TE.Net8.01, TE.Net8.02 and TE.Net8.03 all PASS for each applicable service class selected in the test scope. FAIL if any applicable TE fails or if VE evidence is insufficient to identify service classes, endpoint identity rules, allowed TLS versions, negative TLS/channel scenarios or certificate-identity scenarios.

SUG — User guidance documents

Table — VE and TE of AS.SUG1 (FITCEM PP: User guidance documents)

Element	Content
User guidance documents — AS.SUG1	**The TOE shall be accompanied by recommendations for users on the secure use of the application and its configuration. This includes, at a minimum:• Restrictions on the use of the TOE,• Guidelines and recommendations for the secure configuration, installation, setup, operation, and maintenance of the TOE for users,• Secure handling of authentication factors (e.g., PIN).**
Required vendor information	
VE.SUG1.01	The vendor shall provide the Secure User Guide or equivalent user recommendations accompanying the evaluated TOE. The guidance shall identify restrictions on TOE use, secure configuration, installation, setup, operation, maintenance and secure handling of authentication factors.
VE.SUG1.02	The vendor shall provide the evaluated TOE build, access to the user guidance delivery channel, test accounts where needed, and instructions for locating the guidance from the user perspective. The vendor shall identify the expected user audience, supported language(s), and the TOE configuration to which the guidance applies.
Required test procedures	
TE.SUG1.01	• **Test description with test expectation:** The TOE shall be accompanied by user recommendations for secure use and configuration. The expected result is that the tester can locate guidance corresponding to the evaluated TOE build and configuration through the vendor-defined user-accessible channel. • **Test preparation:** Prepare the evaluated TOE build and the guidance delivery channel according to VE.SUG1.02. Identify the expected audience, supported language, TOE version/build and evaluated configuration before testing. • **Testing steps:** The tester records the TOE version, platform, build identifier, guidance channel, language and evaluated configuration in the test evidence. The tester installs or opens the TOE as an expected user, locates the guidance through the defined channel, and checks that the guidance is identifiable as applying to the evaluated TOE/configuration. • **Test result:** The actual result is the observed guidance availability: the guidance is available and identifiable for the evaluated TOE, is available but ambiguous, or is not accessible to the expected user. The result shall describe the visible guidance channel, version/configuration indication and language availability. • **Assessment (pass/fail):** PASS if the secure-use recommendations accompany the evaluated TOE and are accessible to the expected user in the specified language/channel. FAIL if the guidance is missing, inaccessible, not identifiable as applying to the evaluated TOE, or only available as internal vendor material.
TE.SUG1.02	• **Test description with test expectation:** The user recommendations shall cover all minimum content areas required by SUG1. The expected result is that the guidance contains applicable restrictions on TOE use, secure configuration/installation/setup/operation/maintenance instructions, and secure handling of authentication factors. • **Test preparation:** Use the guidance located in TE.SUG1.01 and the evaluated TOE configuration from VE.SUG1.02. Prepare a checklist containing the minimum SUG1 content categories and any vendor-declared non-applicability rationale. • **Testing steps:** The tester records the TOE version, guidance version/channel, evaluated configuration and SUG1 content checklist in the test evidence. The tester reviews the guidance and maps each required content category to the corresponding guidance section, screen, link or in-app instruction. • **Test result:** The actual result is the observed content coverage: all required categories are covered, some categories are incomplete or ambiguous, or one or more required categories are absent. The result shall identify covered and missing categories without rewriting the full guidance text. • **Assessment (pass/fail):** PASS if all SUG1 minimum content categories are present and applicable non-applicability claims are justified. FAIL if restrictions, secure-use lifecycle guidance, or authentication-factor handling guidance is missing or materially incomplete.
TE.SUG1.03	• **Test description with test expectation:** The selected SUG1 recommendations shall be consistent with the evaluated TOE UI and behaviour. The expected result is that the tester can follow selected guidance steps and reach the described secure configuration or secure-use outcome.

Element	Content
	• **Test preparation:** Select one recommendation related to secure configuration/setup and one recommendation related to secure handling of an authentication factor. Prepare the TOE in the evaluated configuration and ensure that test authentication factors are used, not real user secrets. • **Testing steps:** The tester records the TOE version, platform, selected guidance sections and selected TOE flows in the test evidence. The tester follows the selected guidance steps in the evaluated TOE and observes whether the UI labels, sequence, required user actions and resulting TOE state match the guidance. • **Test result:** The actual result is the observed guidance-to-TOE consistency: the selected recommendations can be followed and produce the described secure-use outcome, or the guidance is outdated, ambiguous, inconsistent with the UI, or leads to a different state. The result shall describe the observed mismatch category if any, without recording authentication-factor values. • **Assessment (pass/fail):** PASS if the selected recommendations are executable on the evaluated TOE and lead to the described secure-use or configuration outcome. FAIL if the guidance cannot be followed, contradicts the TOE UI/behaviour, or causes the tester to deviate from the evaluated secure configuration.
Test Verdict	PASS iff TE.SUG1.01, TE.SUG1.02 and TE.SUG1.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the evaluated guidance, intended audience, language, TOE configuration or guidance delivery channel.

Table — VE and TE of AS.SUG2 (FITCEM PP: User guidance documents)

Element	Content
User guidance documents — AS.SUG2	**The TOE shall be accompanied by contact information for technical support and a procedure for reporting security issues.**
Required vendor information	
VE.SUG2.01	The vendor shall provide the user-accessible contact information for technical support and the procedure for reporting security issues. The evidence shall identify the applicable channels, expected audience, supported language(s), and the TOE version or configuration to which the information applies.
VE.SUG2.02	The vendor shall provide SUG/test instructions and, where applicable, a non-production support or security-reporting path allowing the tester to verify availability and usability of the contact/reporting information. The vendor shall identify any security-reporting path intended for end users, security researchers, or both.
Required test procedures	
TE.SUG2.01	• **Test description with test expectation:** The TOE shall be accompanied by user-accessible technical-support contact information and security-issue reporting information. The expected result is that the tester can locate both categories of information through the vendor-defined guidance or TOE-supported channel. • **Test preparation:** Prepare the evaluated TOE build and the guidance delivery channel according to VE.SUG2.01 and VE.SUG2.02. Identify the expected user audience, supported language, TOE version/build and evaluated configuration before testing. • **Testing steps:** The tester records the TOE version, platform, build identifier, guidance channel, language and evaluated configuration in the test evidence. The tester accesses the TOE or accompanying guidance as an expected user and locates the technical-support contact information and the security-issue reporting information. • **Test result:** The actual result is the observed availability of contact/reporting information: both categories are present and identifiable, only one category is present, the information is ambiguous, or the information is not accessible to the expected user. The result shall describe the visible channel and information categories, not personal contact data beyond what is necessary for evidence. • **Assessment (pass/fail):** PASS if both technical-support contact information and security-issue reporting information are accessible and identifiable for the evaluated TOE. FAIL if either category is missing, inaccessible, not attributable to the evaluated TOE, or only available as internal vendor information.

Element	Content
TE.SUG2.02	• **Test description with test expectation:** The security-issue reporting procedure shall provide clear steps for reporting security issues related to the TOE. The expected result is that the procedure identifies the reporting channel, reporting steps, required or recommended information, and any safety instruction for not disclosing secrets or unnecessary personal data. • **Test preparation:** Use the guidance or reporting procedure located in TE.SUG2.01. Prepare a checklist covering technical support contact, security-issue reporting channel, reporting steps, issue categories and safe-reporting instructions. • **Testing steps:** The tester records the TOE version, guidance version/channel, reporting-procedure location and checklist categories in the test evidence. The tester reviews the procedure and maps each required category to the corresponding guidance section, screen, form, link or in-app instruction. • **Test result:** The actual result is the observed procedure content: the procedure is complete and distinguishes security reporting from general support, is incomplete or ambiguous, or is absent. The result shall identify covered and missing categories without reproducing the full procedure text. • **Assessment (pass/fail):** PASS if the procedure gives sufficient user-facing steps to report security issues safely and distinguishes the security-reporting path from general technical support where applicable. FAIL if the procedure is missing, materially incomplete, or likely to cause users to disclose secrets/personal data unnecessarily.
TE.SUG2.03	• **Test description with test expectation:** The selected technical-support or security-reporting path shall be usable according to the procedure. The expected result is that the tester reaches the documented support/reporting endpoint, form, email instruction, in-app flow or equivalent channel without contradiction to the guidance. • **Test preparation:** Select one technical-support contact path and one security-issue reporting path from VE.SUG2.01. Prepare a harmless test inquiry or test security-reporting scenario that does not contain real vulnerabilities, secrets or personal data. • **Testing steps:** The tester records the TOE version, platform, selected support path, selected security-reporting path and test scenario in the test evidence. The tester follows the documented steps up to the point of submission or, where allowed, submits a harmless test message through the non-production or approved production-safe path. • **Test result:** The actual result is the observed usability of the path: the channel is reachable and consistent with the procedure, the channel is reachable but inconsistent or ambiguous, or the channel is unavailable. The result shall describe the observed endpoint/form/in-app state and any acknowledgement category, without including secrets or real vulnerability details. • **Assessment (pass/fail):** PASS if the selected support and security-reporting paths are reachable and consistent with the documented procedure. FAIL if a documented path is broken, unavailable, misleading, or cannot be used as described.
Test Verdict	PASS iff TE.SUG2.01, TE.SUG2.02 and TE.SUG2.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the applicable contact information, reporting procedure, guidance channel, supported language or test-safe reporting path.

Table — VE and TE of AS.SUG3 (FITCEM PP: User guidance documents)

Element	Content
User guidance documents — AS.SUG3	**The conditions and deadlines for revoking the TOE and the availability of the revocation service shall be communicated in clear language.**
Required vendor information	
VE.SUG3.01	The vendor shall provide the user-accessible guidance describing the conditions and deadlines for revoking the TOE and the availability of the revocation service. The guidance shall identify the applicable TOE version/configuration, intended audience, supported language(s), and user-accessible delivery channel.
VE.SUG3.02	The vendor shall provide SUG/test instructions, the evaluated TOE build, and, where applicable, access to the revocation-information channel or test revocation-service information page. The vendor shall define how the tester can verify that the revocation information is accessible, understandable and consistent with the TOE-supported user path.

Element	Content
Required test procedures	
TE.SUG3.01	• **Test description with test expectation:** The TOE shall be accompanied by user-accessible information on TOE revocation conditions, deadlines and revocation-service availability. The expected result is that the tester can locate this information through the vendor-defined SUG or TOE-supported user channel. • **Test preparation:** Prepare the evaluated TOE build and guidance channel according to VE.SUG3.01 and VE.SUG3.02. Identify the TOE version/build, evaluated configuration, expected user audience, language and revocation-information channel before testing. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected guidance/revocation-information channel, language and evaluated configuration in the test evidence. The tester accesses the TOE or accompanying guidance as an expected user and locates the section or user path describing TOE revocation conditions, deadlines and service availability. • **Test result:** The actual result is the observed availability of revocation information: the information is present and identifiable, partially present, ambiguous, or not accessible to the expected user. The result shall describe the visible channel and information categories without copying unnecessary legal or operational text. • **Assessment (pass/fail):** PASS if the revocation conditions, deadlines and revocation-service availability information are accessible and identifiable for the evaluated TOE. FAIL if any required category is missing, inaccessible, or not attributable to the evaluated TOE/configuration.
TE.SUG3.02	• **Test description with test expectation:** The revocation information shall be communicated in clear language suitable for the expected user audience. The expected result is that the guidance states the revocation conditions, deadlines and service-availability information without relying on unexplained internal, legal or technical terminology. • **Test preparation:** Use the revocation information located in TE.SUG3.01. Prepare a checklist covering revocation conditions, deadlines, service availability, user consequence, and user action or information path. • **Testing steps:** The tester records the TOE version, guidance version/channel, selected language, intended audience and SUG3 checklist in the test evidence. The tester reviews the located guidance and maps each required category to the corresponding text, screen, link, FAQ entry or in-app instruction. • **Test result:** The actual result is the observed clarity and coverage: the information is clear and complete for the expected user, incomplete, ambiguous, or dependent on unexplained terminology. The result shall identify missing or unclear categories rather than rewriting the full guidance text. • **Assessment (pass/fail):** PASS if the guidance clearly communicates all required SUG3 categories in language suitable for the expected user audience. FAIL if the conditions, deadlines or service availability are unclear, materially incomplete, or likely to mislead the expected user.
TE.SUG3.03	• **Test description with test expectation:** The revocation-service availability information shall be consistent with the TOE-supported or vendor-defined user path for obtaining that information. The expected result is that the tester can follow the guidance to the stated revocation-service status or information channel. • **Test preparation:** Select one revocation-service availability path from VE.SUG3.02, such as in-app status, help page, public service-status page, support flow or equivalent channel. Prepare a harmless test observation that does not trigger real TOE revocation unless a dedicated test environment is provided. • **Testing steps:** The tester records the TOE version, platform, selected revocation-service information path and expected availability/status outcome in the test evidence. The tester follows the guidance up to the stated availability/status information point and observes whether the path, labels and resulting information match the guidance. • **Test result:** The actual result is the observed path consistency: the stated path leads to the revocation-service availability information, leads to incomplete or inconsistent information, or is unavailable. The result shall describe the observed UI/page/status category and any mismatch with the guidance. • **Assessment (pass/fail):** PASS if the stated path is reachable and provides revocation-service availability information consistent with the guidance. FAIL if the path is broken, unavailable, misleading, or inconsistent with the revocation information communicated to users.

Element	Content
Test Verdict	PASS iff TE.SUG3.01, TE.SUG3.02 and TE.SUG3.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the applicable revocation information, intended audience, language, TOE configuration or revocation-service information path.

Table — VE and TE of AS.SUG4 (FITCEM PP: User guidance documents)

Element	Content
User guidance documents — AS.SUG4	**The user shall have access to information regarding mobile device models that are compatible with the TOE and the minimum version of the operating system required by the TOE.**
Required vendor information	
VE.SUG4.01	The vendor shall provide user-accessible compatibility information identifying mobile device models compatible with the TOE and the minimum operating-system version required by the TOE. The information shall identify the applicable TOE version/build, supported platforms, and the evaluated configuration to which the compatibility statement applies.
VE.SUG4.02	The vendor shall provide SUG/test instructions, the evaluated TOE build, supported and unsupported test-device information, and the user-accessible channel where compatibility information is published. The vendor shall define how the tester can verify that the compatibility information is reachable and consistent with the evaluated TOE.
Required test procedures	
TE.SUG4.01	• **Test description with test expectation:** The user shall have access to compatibility information for the TOE. The expected result is that the tester can locate information on compatible mobile device models and minimum OS version through the vendor-defined user-accessible channel. • **Test preparation:** Prepare the evaluated TOE build and guidance or compatibility-information channel according to VE.SUG4.01 and VE.SUG4.02. Identify the TOE version/build, supported platform, expected user audience, language and evaluated configuration before testing. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected compatibility-information channel, language and evaluated configuration in the test evidence. The tester accesses the TOE or accompanying guidance as an expected user and locates the compatibility information for device models and minimum OS version. • **Test result:** The actual result is the observed availability of compatibility information: the information is present and identifiable, partially present, ambiguous, or not accessible to the expected user. The result shall describe the visible channel, version/configuration indication and compatibility-information categories. • **Assessment (pass/fail):** PASS if compatibility information for mobile device models and minimum OS version is accessible and identifiable for the evaluated TOE. FAIL if the information is missing, inaccessible, not attributable to the evaluated TOE/configuration, or only available as internal vendor information.
TE.SUG4.02	• **Test description with test expectation:** The compatibility information shall clearly identify compatible mobile device models or model families and the minimum OS version required by the TOE. The expected result is that the information enables the expected user to determine whether a device/OS combination is supported. • **Test preparation:** Use the compatibility information located in TE.SUG4.01. Prepare a checklist covering supported platform, device model/model-family information, minimum OS version, TOE version/build applicability and any exclusions or limitations. • **Testing steps:** The tester records the TOE version, guidance/compatibility-information version, selected language, intended audience and SUG4 checklist in the test evidence. The tester reviews the information and maps each required category to the corresponding guidance section, compatibility table, support page, in-app screen or linked document. • **Test result:** The actual result is the observed content coverage: the compatibility information clearly covers device models/model families and minimum OS version, covers them incompletely, or leaves compatibility ambiguous. The result shall identify covered, missing or unclear categories without reproducing the full compatibility list.

Element	Content
	• **Assessment (pass/fail):** PASS if the compatibility information clearly states the compatible device scope and minimum OS version for the evaluated TOE. FAIL if the user cannot determine whether a device/OS combination is compatible with the TOE.
TE.SUG4.03	• **Test description with test expectation:** The selected compatibility information shall be consistent with the evaluated TOE behaviour on supported and below-minimum or unsupported configurations. The expected result is normal operation on the supported test configuration and safe blocking, warning, or documented non-support on the unsupported/below-minimum configuration where such testing is feasible. • **Test preparation:** Select one supported device/OS configuration and one unsupported or below-minimum configuration from VE.SUG4.02, where available in the test environment. If the unsupported configuration cannot be practically tested, prepare the vendor-defined non-applicability rationale and supported observation alternative. • **Testing steps:** The tester records the TOE version, platform, selected supported configuration, selected unsupported/below-minimum configuration, and expected TOE behaviour in the test evidence. The tester installs or opens the TOE on the selected configurations and observes installation, launch, compatibility warning, blocking behaviour, or normal operation through the approved method. • **Test result:** The actual result is the observed consistency between guidance and TOE behaviour: the supported configuration operates as described and the unsupported/below-minimum configuration is safely handled, or the behaviour contradicts the compatibility information. The result shall describe the visible UI/status, store/platform message, compatibility warning or operation outcome. • **Assessment (pass/fail):** PASS if observed TOE behaviour is consistent with the compatibility information for the tested configurations. FAIL if the TOE contradicts its stated compatibility information or allows normal use below the stated minimum OS without documented safe handling.
Test Verdict	PASS iff TE.SUG4.01, TE.SUG4.02 and TE.SUG4.03 all PASS, with TE.SUG4.03 partially documented as not applicable only where unsupported/below-minimum test configurations are not practically available and an accepted alternative observation is provided. FAIL if any applicable TE fails or if VE evidence is insufficient to identify compatibility information, supported platforms, minimum OS version, evaluated TOE build or user-accessible channel.

Table — VE and TE of AS.SUG5 (FITCEM PP: User guidance documents)

Element	Content
User guidance documents — AS.SUG5	**The user shall be informed of any suspension or revocation of the TOE, including the reasons and consequences of the suspension, in clear and transparent language no later than 24 hours after the event.**
Required vendor information	
VE.SUG5.01	The vendor shall provide the user-notification specification for TOE suspension and revocation events. The specification shall define notification triggers, delivery channels, deadline calculation, message content, language, reasons, consequences, and any permitted redaction for security-sensitive information.
VE.SUG5.02	The vendor shall provide SUG/test instructions, test accounts, evaluated TOE build, and a non-production or simulated suspension/revocation event mechanism allowing the tester to observe user notification without causing real certificate suspension or revocation. The vendor shall provide a method for observing event time, notification time, delivery status, and user-visible message content.
Required test procedures	
TE.SUG5.01	• **Test description with test expectation:** The TOE or its accompanying user-notification mechanism shall inform the User of a TOE suspension event no later than 24 hours after the event. The expected result is a user-visible notification containing the suspension status, reasons and consequences in clear and transparent language. • **Test preparation:** Prepare the evaluated TOE build, one test User and the vendor-defined suspension test event from VE.SUG5.02. Identify the expected notification channel, event timestamp source, deadline rule and required message-content categories from VE.SUG5.01.

Element	Content
	• **Testing steps:** The tester records the TOE version, platform, build identifier, test User, suspension event timestamp, selected notification channel and expected content categories in the test evidence. The tester triggers the vendor-supported suspension test event and observes the TOE UI, in-app inbox, push notification, email, support portal or other defined user channel until the notification is delivered or the test deadline is reached. • **Test result:** The actual result is the observed suspension-notification behaviour: the User receives a clear suspension notification within the defined deadline, receives an incomplete or late notification, or receives no notification. The result shall describe the observed channel, timing category, status category, reason category and consequence category without copying security-sensitive details unnecessarily. • **Assessment (pass/fail):** PASS if the User is informed of the suspension within 24 hours and the message clearly states the suspension, reasons and consequences, subject only to justified security-sensitive redaction. FAIL if the notification is missing, late, unclear, does not identify the suspension, or omits required reason/consequence categories without documented justification.
TE.SUG5.02	• **Test description with test expectation:** The TOE or its accompanying user-notification mechanism shall inform the User of a TOE revocation event no later than 24 hours after the event. The expected result is a user-visible notification containing the revocation status, reasons and consequences in clear and transparent language. • **Test preparation:** Prepare the evaluated TOE build, one test User and the vendor-defined revocation test event from VE.SUG5.02. Identify the expected notification channel, event timestamp source, deadline rule and required revocation-message categories from VE.SUG5.01. • **Testing steps:** The tester records the TOE version, platform, build identifier, test User, revocation event timestamp, selected notification channel and expected content categories in the test evidence. The tester triggers the vendor-supported revocation test event and observes the TOE UI, in-app inbox, push notification, email, support portal or other defined user channel until the notification is delivered or the test deadline is reached. • **Test result:** The actual result is the observed revocation-notification behaviour: the User receives a clear revocation notification within the defined deadline, receives an incomplete or late notification, or receives no notification. The result shall describe the observed channel, timing category, status category, reason category and consequence category without copying security-sensitive details unnecessarily. • **Assessment (pass/fail):** PASS if the User is informed of the revocation within 24 hours and the message clearly states the revocation, reasons and consequences, subject only to justified security-sensitive redaction. FAIL if the notification is missing, late, unclear, does not identify the revocation, or omits required reason/consequence categories without documented justification.
Test Verdict	PASS iff TE.SUG5.01 and TE.SUG5.02 both PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the notification trigger, delivery channel, event timestamp, deadline rule, required content categories or user-observation method.

Table — VE and TE of AS.SUG6 (FITCEM PP: User guidance documents)

Element	Content
User guidance documents — AS.SUG6	**The TOE shall present clear security and privacy notices before granting access to or executing any wallet function that may disclose wallet data, change wallet state, use a device feature, initiate a presentation, signature or credential operation, or otherwise affect the user's security or privacy. The notice shall identify the purpose, requester or recipient where applicable, data categories, retention or logging consequences, and required user action, and shall remain available until the user acknowledges or cancels the operation.**
Required vendor information	
VE.SUG6.01	The vendor shall identify wallet functions requiring security/privacy notices in the evaluated configuration and define the notice content required for each function class. The definition shall cover purpose, requester/recipient where applicable, data categories, retention/logging consequences and required User action.

Element	Content
VE.SUG6.02	The vendor shall provide SUG/test instructions, test accounts, test data and test endpoints allowing the tester to trigger representative notice-requiring wallet functions. The vendor shall define expected behaviour for acknowledge, cancel and no-action states.
Required test procedures	
TE.SUG6.01	• **Test description with test expectation:** The TOE shall present a clear security/privacy notice before executing the selected notice-requiring wallet function. The expected result is that the notice appears before execution and contains the required content categories applicable to that function. • **Test preparation:** Prepare the evaluated TOE, one test User, test data and test endpoint according to VE.SUG6.02. Select at least one representative function that may disclose wallet data or initiate a presentation/credential/signature operation. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected function, expected notice-content categories and observation method in the test evidence. The tester initiates the selected function and observes the notice before taking any acknowledge or cancel action. • **Test result:** The actual result is the observed notice behaviour: the notice is shown before execution with the required categories, is shown but incomplete or ambiguous, or is not shown before the function proceeds. The result shall describe visible content categories without copying real personal data or secrets. • **Assessment (pass/fail):** PASS if the notice is presented before execution and identifies the applicable purpose, requester/recipient, data categories, retention/logging consequences and required User action. FAIL if the function proceeds before the notice or if required notice categories are missing or materially unclear.
TE.SUG6.02	• **Test description with test expectation:** The TOE shall keep the notice available until the User acknowledges or cancels the operation. The expected result is no execution while the notice is pending, and safe non-execution after cancellation. • **Test preparation:** Use the same TOE, User and selected function as in TE.SUG6.01. Prepare the vendor-defined pending/no-action and cancel paths for the selected notice. • **Testing steps:** The tester records the TOE version, platform, selected function, pending/cancel scenario and expected non-execution outcome in the test evidence. The tester initiates the function, leaves the notice pending where feasible, then cancels the operation and observes the TOE UI, test endpoint or backend result. • **Test result:** The actual result is the observed control behaviour: the notice remains available while pending, cancellation stops the operation, or the function executes despite no acknowledgement or cancellation. The result shall state whether any data disclosure, state change, device-feature use or operation initiation occurred. • **Assessment (pass/fail):** PASS if the function is not executed before acknowledgement and is not completed after cancellation. FAIL if the function proceeds while the notice is pending or after the User cancels.
TE.SUG6.03	• **Test description with test expectation:** The TOE shall also present a clear notice before a selected state-changing, device-feature or otherwise security/privacy-affecting wallet function. The expected result is that the notice explains the purpose, consequence and required User action before the function proceeds. • **Test preparation:** Select one representative non-disclosure function from VE.SUG6.01, such as wallet-state change, device-feature use or another security/privacy-affecting operation. Prepare the TOE state and test data required to trigger that function safely. • **Testing steps:** The tester records the TOE version, platform, selected function class, expected notice-content categories and expected execution outcome in the test evidence. The tester initiates the selected function, observes the notice, acknowledges it, and then observes whether the function proceeds as described. • **Test result:** The actual result is the observed notice and execution behaviour: the notice is shown before the function and the acknowledged operation proceeds as described, or the notice is missing/incomplete, or the operation outcome differs from the notice. The result shall describe the visible notice category and operation outcome. • **Assessment (pass/fail):** PASS if the notice is shown before the selected non-disclosure security/privacy-affecting function and the operation proceeds only after acknowledgement. FAIL if

Element	Content
	the function proceeds without notice, the notice omits material consequence information, or the operation outcome contradicts the notice.
Test Verdict	PASS iff TE.SUG6.01, TE.SUG6.02 and TE.SUG6.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify notice-requiring functions, required notice content, acknowledge/cancel behaviour or observation methods.

SECDOC — Secure use of the TOE

Table — VE and TE of AS.SecDoc1 (FITCEM PP: Secure use of the TOE)

Element	Content
Secure use of the TOE — AS.SecDoc1	**The Wallet Provider shall document the following functions available in the TOE:• TOE interfaces,• Users,• Services and associated functions,• Activation and deactivation status for the services.**
Required vendor information	
VE.SecDoc1.01	The vendor shall provide the documentation identifying TOE interfaces, Users, services, associated functions, and activation/deactivation status for services in the evaluated configuration. The documentation may be provided in ST, functional specification, TOE design, SUG or another clearly identified evaluation artefact.
VE.SecDoc1.02	The vendor shall provide the evaluated TOE build, test configuration, test accounts and instructions allowing the tester to observe representative interfaces, users, services/functions and service activation/deactivation states. The vendor shall define how documented functions map to observable TOE behaviour.
Required test procedures	
TE.SecDoc1.01	• **Test description with test expectation:** The vendor documentation shall contain all SecDoc1 categories for the evaluated TOE configuration. The expected result is that TOE interfaces, Users, services, associated functions, and activation/deactivation status for services are explicitly documented. • **Test preparation:** Prepare the submitted documentation set and identify which artefact is intended to satisfy SecDoc1. Prepare a checklist containing the four required SecDoc1 categories. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, documentation version and selected documentation artefacts in the test evidence. The tester reviews the submitted documentation and maps each SecDoc1 category to the corresponding section, table, diagram or inventory item. • **Test result:** The actual result is the observed documentation coverage: all required categories are documented, some categories are incomplete or ambiguous, or one or more categories are absent. The result shall identify missing or ambiguous categories without reproducing the full documentation. • **Assessment (pass/fail):** PASS if all four SecDoc1 categories are present and attributable to the evaluated TOE configuration. FAIL if any required category is missing, materially incomplete, or not clearly linked to the evaluated TOE.
TE.SecDoc1.02	• **Test description with test expectation:** The documented TOE interfaces, Users, services and associated functions shall be consistent with the observable evaluated TOE. The expected result is that representative documented items can be observed or exercised as described, and no documented item is materially inconsistent with the tested TOE configuration. • **Test preparation:** Select a representative sample of documented TOE interfaces, User types or personas, and services/functions from VE.SecDoc1.01. Prepare the TOE and test accounts according to VE.SecDoc1.02. • **Testing steps:** The tester records the TOE version, platform, build identifier, selected documented items and observation method in the test evidence. The tester observes or exercises the selected items through the TOE UI, supported interfaces, test endpoint, configuration screen or approved diagnostic output. • **Test result:** The actual result is the observed documentation-to-TOE consistency: selected items behave as documented, are absent or inaccessible despite being documented as available, or behave in a materially different way. The result shall describe the mismatch category where applicable. • **Assessment (pass/fail):** PASS if the selected documented items are consistent with observable TOE behaviour in the evaluated configuration. FAIL if documented interfaces, Users, services or functions are materially inconsistent with the TOE build under test.
TE.SecDoc1.03	• **Test description with test expectation:** The activation and deactivation status of selected services shall be documented and consistent with observable TOE configuration. The expected result is that an active service is available as documented, and a deactivated or unavailable service is not usable except through the documented activation path, if such a path exists.

Element	Content
	• **Test preparation:** Select at least one documented active service and, where present, one documented deactivated or optional service from VE.SecDoc1.01. Identify the expected status and observation path for each selected service. • **Testing steps:** The tester records the TOE version, platform, selected services, documented activation/deactivation status and expected observable state in the test evidence. The tester checks the service status through the supported UI, configuration path, test endpoint or approved diagnostic method and attempts only the normal use paths described by the vendor. • **Test result:** The actual result is the observed status behaviour: each selected service is available, unavailable, activatable or deactivated consistently with the documentation, or the observed state contradicts the documented status. The result shall identify the service-status category and any inconsistency. • **Assessment (pass/fail):** PASS if selected service activation/deactivation states are documented and match observable TOE behaviour. FAIL if service status is undocumented, ambiguous, or inconsistent with the evaluated TOE configuration.
Test Verdict	PASS iff TE.SecDoc1.01, TE.SecDoc1.02 and TE.SecDoc1.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the documented TOE interfaces, Users, services/functions or service activation/deactivation states.

Table — VE and TE of AS.SecDoc2 (FITCEM PP: Secure use of the TOE)

Element	Content
Secure use of the TOE — AS.SecDoc2	**The Wallet Provider shall provide a Hardware Specification for the TOE. This specification shall describe all hardware components and features that are necessary for the Target of Evaluation (TOE) to perform its security functions correctly and securely within the defined hardware environment (e.g. Camera, NFC module).**
Required vendor information	
VE.SecDoc2.01	The vendor shall provide the Hardware Specification for the evaluated TOE configuration. The specification shall identify all hardware components and hardware features required for correct and secure operation of TOE security functions within the defined hardware environment.
VE.SecDoc2.02	The vendor shall provide the evaluated TOE build, supported device information, test accounts, and instructions allowing the tester to observe representative hardware-dependent TOE security functions. The vendor shall define how each selected hardware component or feature maps to the TOE security function that depends on it.
Required test procedures	
TE.SecDoc2.01	• **Test description with test expectation:** The Hardware Specification shall identify all hardware components and features necessary for TOE security functions in the evaluated configuration. The expected result is that required hardware components/features and their dependent TOE security functions are explicitly documented. • **Test preparation:** Prepare the submitted Hardware Specification and the evaluated TOE configuration. Prepare a checklist covering hardware components/features, dependent TOE security functions, supported device environment, exclusions and assumptions. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, Hardware Specification version and checklist categories in the test evidence. The tester reviews the Hardware Specification and maps each required category to the corresponding section, table, diagram or inventory item. • **Test result:** The actual result is the observed documentation coverage: the Hardware Specification covers the required hardware components/features and their security-function relevance, covers them incompletely, or omits one or more necessary items. The result shall identify missing or ambiguous categories without reproducing the full specification. • **Assessment (pass/fail):** PASS if the Hardware Specification explicitly covers the hardware components/features necessary for TOE security functions in the evaluated hardware environment. FAIL if required hardware components/features are missing, materially incomplete, or not linked to the TOE security functions they support.

Element	Content
TE.SecDoc2.02	• **Test description with test expectation:** Documented hardware-dependent TOE security functions shall be consistent with observable TOE behaviour on a supported hardware environment. The expected result is that the selected TOE function uses or requires the documented hardware component/feature as specified. • **Test preparation:** Select one or more representative hardware-dependent TOE security functions from VE.SecDoc2.01. Prepare the TOE on a supported device or test configuration that contains the documented hardware component/feature. • **Testing steps:** The tester records the TOE version, platform, device model, OS version, selected hardware component/feature, selected TOE security function and expected behaviour in the test evidence. The tester executes the selected TOE flow and observes the TOE UI, platform permission/status, test endpoint, backend result or approved diagnostic output. • **Test result:** The actual result is the observed hardware-function consistency: the selected TOE function behaves consistently with the documented hardware dependency, behaves without the documented dependency, or cannot be exercised as documented. The result shall describe the visible UI/status, permission state, operation outcome or diagnostic category. • **Assessment (pass/fail):** PASS if the selected hardware-dependent TOE function is consistent with the Hardware Specification on the supported hardware environment. FAIL if the TOE behaviour contradicts the documented dependency or the documented function cannot be exercised without justified limitation.
TE.SecDoc2.03	• **Test description with test expectation:** Where a required hardware component/feature is absent, disabled or outside the defined hardware environment, the TOE shall follow the documented unsupported-hardware or degraded-state behaviour. The expected result is safe blocking, clear non-support indication, disabled function, or another documented safe outcome. • **Test preparation:** Select one required hardware component/feature from VE.SecDoc2.01 and prepare a vendor-supported unsupported, absent, disabled or permission-denied scenario. If such a scenario is not feasible in the laboratory, prepare the vendor's documented non-applicability rationale and alternative observation method. • **Testing steps:** The tester records the TOE version, platform, device/test configuration, selected hardware component/feature, unsupported-hardware scenario and expected safe outcome in the test evidence. The tester attempts the related TOE security function under the unsupported/disabled condition and observes the TOE UI, platform status, test endpoint, backend decision or approved diagnostic output. • **Test result:** The actual result is the observed unsupported-hardware handling: the TOE blocks or disables the affected function, shows a clear unsupported/degraded state, safely redirects the User, or proceeds as if the required hardware were available. The result shall describe the visible state and whether the security-relevant function completed. • **Assessment (pass/fail):** PASS if the TOE handles the unsupported/disabled hardware condition according to the Hardware Specification and does not execute the affected security function insecurely. FAIL if the TOE proceeds contrary to the documented hardware requirement or leaves the tester unable to determine the hardware/security state.
Test Verdict	PASS iff TE.SecDoc2.01, TE.SecDoc2.02 and TE.SecDoc2.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify the Hardware Specification, required hardware components/features, dependent TOE security functions, supported hardware environment or unsupported-hardware behaviour.

Table — VE and TE of AS.SecDoc3 (FITCEM PP: Secure use of the TOE)

Element	Content
Secure use of the TOE — AS.SecDoc3	**The Wallet Provider shall provide a Compatibility Description of the supported end devices. This description shall specify the hardware and software configurations for which the TOE is intended to operate and shall serve as the basis for assessing whether the TOE can maintain its security properties under the defined system environment conditions.Alternatively, the Wallet Provider shall specify the list of supported mobile devices.**
Required vendor information	

Element	Content
VE.SecDoc3.01	The vendor shall provide the Compatibility Description for supported end devices, or the alternative list of supported mobile devices. The evidence shall identify the applicable TOE version/build, supported platforms, hardware configuration, software configuration, OS versions and system-environment assumptions relevant to TOE security properties.
VE.SecDoc3.02	The vendor shall provide the evaluated TOE build, test-device information, supported and unsupported configuration examples, and instructions allowing the tester to observe TOE behaviour on selected end-device configurations. The vendor shall define how compatibility status is determined and how unsupported or out-of-scope configurations are handled.
Required test procedures	
TE.SecDoc3.01	• **Test description with test expectation:** The Compatibility Description or supported-device list shall identify the end-device configurations for which the TOE is intended to operate. The expected result is that supported hardware/software configurations, or supported mobile devices, are explicitly documented and linked to the evaluated TOE version/build. • **Test preparation:** Prepare the submitted Compatibility Description or supported-device list and the evaluated TOE configuration. Prepare a checklist covering TOE version/build, supported platforms, hardware configuration, software/OS configuration, supported devices or device families, exclusions and system-environment assumptions. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, documentation version and checklist categories in the test evidence. The tester reviews the Compatibility Description or supported-device list and maps each required category to the corresponding section, table, compatibility matrix or inventory item. • **Test result:** The actual result is the observed documentation coverage: the Compatibility Description/list clearly identifies supported end-device configurations, identifies them incompletely or ambiguously, or omits necessary compatibility information. The result shall identify missing or unclear categories without reproducing the full compatibility matrix. • **Assessment (pass/fail):** PASS if the submitted evidence clearly defines the supported end-device scope for the evaluated TOE. FAIL if the supported hardware/software configurations or supported mobile devices are missing, ambiguous, not versioned, or not attributable to the evaluated TOE configuration.
TE.SecDoc3.02	• **Test description with test expectation:** A selected supported end-device configuration shall allow the TOE to operate in the evaluated secure configuration and to exercise representative security-relevant TOE functions. The expected result is that the TOE behaves consistently with the Compatibility Description on the selected supported configuration. • **Test preparation:** Select one supported device or supported hardware/software configuration from VE.SecDoc3.01. Prepare the TOE, test User and one representative security-relevant TOE operation that depends on the defined system environment. • **Testing steps:** The tester records the TOE version, platform, device model, OS version, selected compatibility entry, selected security-relevant operation and expected behaviour in the test evidence. The tester installs or opens the TOE on the selected supported configuration and exercises the selected operation through the normal TOE path. • **Test result:** The actual result is the observed supported-configuration behaviour: the TOE operates as described and maintains the selected security-relevant function, fails despite the configuration being documented as supported, or shows a compatibility/security-state ambiguity. The result shall describe the visible UI/status, platform state, operation outcome or approved diagnostic category. • **Assessment (pass/fail):** PASS if the selected supported configuration is consistent with the documentation and the selected security-relevant operation behaves as expected. FAIL if the TOE cannot operate securely on a configuration documented as supported, unless a documented limitation or non-applicability condition applies.
TE.SecDoc3.03	• **Test description with test expectation:** Where the TOE is used on an unsupported or out-of-scope end-device configuration, it shall follow the vendor-defined compatibility handling. The expected result is safe blocking, clear unsupported-configuration indication, restricted operation, or another documented safe outcome. • **Test preparation:** Select one unsupported, below-minimum or out-of-scope device/software configuration from VE.SecDoc3.02. If such configuration is not practically available, prepare the

Element	Content
	vendor-defined non-applicability rationale and an alternative observation method, such as store eligibility, installation gating or compatibility-status evidence. • **Testing steps:** The tester records the TOE version, platform, selected unsupported/out-of-scope configuration, expected handling and observation method in the test evidence. The tester attempts installation, launch or the selected security-relevant operation under that configuration and observes the TOE UI, platform/store response, compatibility warning, blocking state or approved diagnostic output. • **Test result:** The actual result is the observed unsupported-configuration handling: the TOE blocks, warns, restricts, documents non-support, or proceeds as if the configuration were supported. The result shall describe the visible compatibility status and whether any security-relevant operation completed. • **Assessment (pass/fail):** PASS if the TOE handles the unsupported or out-of-scope configuration consistently with the Compatibility Description. FAIL if the TOE permits normal security-relevant operation on an unsupported configuration without documented safe handling.
Test Verdict	PASS iff TE.SecDoc3.01, TE.SecDoc3.02 and TE.SecDoc3.03 all PASS, with TE.SecDoc3.03 documented as partially not applicable only where an unsupported/out-of-scope test configuration is not practically available and an accepted alternative observation is provided. FAIL if any applicable TE fails or if VE evidence is insufficient to identify supported end devices, hardware/software configurations, system-environment assumptions or compatibility-handling behaviour.

Table — VE and TE of AS.SecDoc4 (FITCEM PP: Secure use of the TOE)

Element	Content
Secure use of the TOE — AS.SecDoc4	**The Wallet Provider shall provide a Software Specification that clearly defines the supported operating systems for the mobile devices, including their minimum required versions. Furthermore, it shall specify any additional software components or libraries required for the TOE's execution due to dependencies. The specifications shall include the minimum versions, and where applicable, the maximum versions of all supported operating systems and additional software.**
Required vendor information	
VE.SecDoc4.01	The vendor shall provide the Software Specification for the evaluated TOE configuration. The specification shall identify supported operating systems, minimum required OS versions, and, where applicable, maximum supported OS versions.
VE.SecDoc4.02	The vendor shall identify additional software components or libraries required for TOE execution due to dependencies. The evidence shall state minimum versions and, where applicable, maximum versions, and shall map each dependency to the TOE function or execution condition that requires it.
Required test procedures	
TE.SecDoc4.01	• **Test description with test expectation:** The Software Specification shall clearly define supported mobile operating systems and their minimum required versions, and maximum versions where applicable. The expected result is explicit, versioned and TOE-build-specific software-environment information. • **Test preparation:** Prepare the submitted Software Specification and the evaluated TOE configuration. Prepare a checklist covering supported OS families, minimum OS versions, maximum OS versions where applicable, TOE version/build applicability and exclusions. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, Software Specification version and checklist categories in the test evidence. The tester reviews the Software Specification and maps each required OS/version category to the corresponding section, table, compatibility matrix or inventory item. • **Test result:** The actual result is the observed documentation coverage: the Software Specification clearly identifies supported OS families and version bounds, identifies them incompletely or ambiguously, or omits required OS/version information. The result shall identify missing or unclear categories without reproducing the full specification.

Element	Content
	• **Assessment (pass/fail):** PASS if the Software Specification explicitly defines supported operating systems and required version bounds for the evaluated TOE. FAIL if supported OS scope, minimum versions, applicable maximum versions, or TOE-build applicability are missing or ambiguous.
TE.SecDoc4.02	• **Test description with test expectation:** The Software Specification shall identify additional software components or libraries required for TOE execution and shall specify their minimum and, where applicable, maximum versions. The expected result is a clear dependency list linked to the evaluated TOE configuration. • **Test preparation:** Use the Software Specification from TE.SecDoc4.01 and the vendor dependency evidence from VE.SecDoc4.02. Prepare a checklist covering additional components/libraries, required versions, version bounds, dependency purpose and non-applicability claims. • **Testing steps:** The tester records the TOE version, build identifier, Software Specification version, selected dependency categories and checklist categories in the test evidence. The tester reviews the dependency information and maps each required software component or library to its version requirement and stated TOE execution dependency. • **Test result:** The actual result is the observed dependency-documentation coverage: required additional software is listed with version bounds and purpose, listed incompletely, or not identified. The result shall identify dependency categories and missing version information, not proprietary implementation details beyond what is needed for evidence. • **Assessment (pass/fail):** PASS if required additional software components/libraries are identified with minimum versions and applicable maximum versions. FAIL if required dependencies are missing, version bounds are absent, or the dependency relationship to TOE execution is unclear.
TE.SecDoc4.03	• **Test description with test expectation:** The Software Specification shall be consistent with observable TOE behaviour on selected supported and unsupported software configurations. The expected result is normal operation on a supported configuration and safe blocking, warning, restricted operation or documented non-support on an out-of-scope configuration. • **Test preparation:** Select one supported OS/software configuration and, where feasible, one below-minimum, above-maximum or missing-dependency configuration from VE.SecDoc4.01 and VE.SecDoc4.02. Prepare a representative TOE operation that demonstrates normal execution under the software environment. • **Testing steps:** The tester records the TOE version, platform, OS version, selected additional software/dependency state, selected supported/out-of-scope configuration and expected behaviour in the test evidence. The tester installs or opens the TOE and performs the selected operation on the selected configuration, then observes TOE UI, platform/store response, compatibility warning, diagnostic output or backend/test endpoint result. • **Test result:** The actual result is the observed software-configuration behaviour: the supported configuration operates as described, the unsupported configuration is blocked/warned/restricted as documented, or TOE behaviour contradicts the Software Specification. The result shall describe visible status, compatibility message, operation outcome or diagnostic category. • **Assessment (pass/fail):** PASS if observed TOE behaviour is consistent with the Software Specification for the tested configurations. FAIL if the TOE cannot operate on a documented supported configuration, or permits normal security-relevant operation on an out-of-scope software configuration without documented safe handling.
Test Verdict	PASS iff TE.SecDoc4.01, TE.SecDoc4.02 and TE.SecDoc4.03 all PASS, with the unsupported-configuration part of TE.SecDoc4.03 documented as not applicable only where such a configuration is not practically available and an accepted alternative observation is provided. FAIL if any applicable TE fails or if VE evidence is insufficient to identify supported OS versions, software dependencies, version bounds or configuration-handling behaviour.

Table — VE and TE of AS.SecDoc5 (FITCEM PP: Secure use of the TOE)

Element	Content
Secure use of the TOE — AS.SecDoc5	**Dependencies: The TOE's dependencies on the system environment in the form of software packages, interfaces, libraries, database connections, APIs, or other components shall be specified.**
Required vendor information	

Element	Content
VE.SecDoc5.01	The vendor shall provide a dependency specification for the evaluated TOE configuration. The specification shall identify software packages, interfaces, libraries, database connections, APIs and other system-environment components required by the TOE.
VE.SecDoc5.02	The vendor shall provide the evaluated TOE build, test configuration, test accounts, test endpoints and observation methods allowing the tester to compare selected dependencies with observable TOE behaviour. The vendor shall define expected behaviour when a selected required dependency is available, unavailable, incompatible or out of scope.
Required test procedures	
TE.SecDoc5.01	• **Test description with test expectation:** The dependency specification shall identify TOE dependencies on the system environment. The expected result is that software packages, interfaces, libraries, database connections, APIs and other required components are specified where applicable to the evaluated TOE. • **Test preparation:** Prepare the submitted dependency specification and the evaluated TOE configuration. Prepare a checklist covering dependency name/category, purpose, required version or version range, provider/source, TOE function using the dependency, and any non-applicability rationale. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, dependency specification version and checklist categories in the test evidence. The tester reviews the dependency specification and maps each required dependency category to the corresponding section, table, inventory item or justified non-applicability statement. • **Test result:** The actual result is the observed documentation coverage: the dependency specification clearly identifies applicable dependency categories, identifies them incompletely or ambiguously, or omits required dependency information. The result shall identify missing or unclear categories without reproducing the full dependency inventory. • **Assessment (pass/fail):** PASS if applicable TOE dependencies on the system environment are specified with sufficient identification and purpose for the evaluated configuration. FAIL if required dependencies are missing, ambiguous, not versioned where version matters, or not attributable to the evaluated TOE.
TE.SecDoc5.02	• **Test description with test expectation:** Selected documented dependencies shall be consistent with observable TOE behaviour in the evaluated configuration. The expected result is that the selected TOE function uses or requires the documented dependency as specified. • **Test preparation:** Select representative dependencies from VE.SecDoc5.01, including at least one external interface/API or service dependency and, where applicable, one local software/library/platform dependency. Prepare the TOE and test environment according to VE.SecDoc5.02. • **Testing steps:** The tester records the TOE version, platform, OS version, build identifier, selected dependencies, selected TOE functions and observation method in the test evidence. The tester exercises the selected TOE functions and observes the TOE UI, configuration status, test endpoint, diagnostic output, backend response or platform status relevant to the selected dependencies. • **Test result:** The actual result is the observed dependency-to-TOE consistency: the selected dependencies are present and used as documented, are not required despite being documented as required, or the TOE requires an undocumented dependency for the selected function. The result shall describe the consistency or mismatch category without exposing secrets, tokens or proprietary implementation details beyond what is needed for evidence. • **Assessment (pass/fail):** PASS if the selected dependencies are consistent with observable TOE behaviour and the selected TOE functions operate as documented. FAIL if a selected documented dependency is materially inconsistent with TOE behaviour, or if the TOE visibly depends on an undocumented component for the tested function.
TE.SecDoc5.03	• **Test description with test expectation:** Where a selected required dependency is unavailable, incompatible or out of scope, the TOE shall follow the vendor-defined handling behaviour. The expected result is safe blocking, restricted operation, clear unsupported-dependency indication, fallback explicitly specified as safe, or another documented defined outcome. • **Test preparation:** Select one dependency from VE.SecDoc5.01 for which a vendor-supported unavailable, incompatible or out-of-scope scenario can be created. If such a scenario cannot be

Element	Content
	practically created, prepare the vendor's documented non-applicability rationale and alternative observation method. • **Testing steps:** The tester records the TOE version, platform, selected dependency, selected unavailable/incompatible scenario, expected handling and observation method in the test evidence. The tester attempts to execute the affected TOE function under the selected dependency condition and observes the TOE UI, platform status, test endpoint, backend decision or approved diagnostic output. • **Test result:** The actual result is the observed dependency-handling behaviour: the TOE blocks, warns, restricts, follows documented safe fallback, or proceeds as if the dependency were valid and available. The result shall describe the visible state and whether the affected security-relevant function completed. • **Assessment (pass/fail):** PASS if the TOE handles the unavailable, incompatible or out-of-scope dependency consistently with the dependency specification. FAIL if the TOE proceeds in an undocumented dependency state, enters an undefined state, or executes a security-relevant function contrary to the specified dependency conditions.
Test Verdict	PASS iff TE.SecDoc5.01, TE.SecDoc5.02 and TE.SecDoc5.03 all PASS, with TE.SecDoc5.03 documented as partially not applicable only where no supported negative dependency scenario is practically available and an accepted alternative observation is provided. FAIL if any applicable TE fails or if VE evidence is insufficient to identify TOE dependencies, dependency purposes, version constraints, affected TOE functions or unsupported-dependency handling.

HRDWSEC — Additional security measures

Table — VE and TE of AS.HrdwSec1 (FITCEM PP: Additional security measures)

Element	Content
Additional security measures — AS.HrdwSec1	**The generation, storage, and processing of cryptographic secrets by the TOE shall be protected by a hardware-based key store (Android StrongBox/TEE, iOS Secure Enclave) on the mobile device using appropriate mechanisms. These mechanisms are required for the Wallet Instance to be installed and operated properly.**
Required vendor information	
VE.HrdwSec1.01	The vendor shall identify cryptographic secrets generated, stored or processed by the TOE in the evaluated configuration and specify the hardware-based keystore mechanism protecting each secret class. The vendor shall distinguish TOE-managed secrets from secrets managed by WSCA/WSCD, platform components or backend services, and define which evidence is observable or requires documentation, configuration or source-code evidence.
VE.HrdwSec1.02	The vendor shall provide SUG/test instructions, supported device information, test accounts, supported observation methods and one conformance-level negative keystore-availability scenario. Implementation evidence may include keystore/API mapping, key-generation parameters, attestation/capability evidence, non-exportability rules, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.HrdwSec1.01	• **Test description with test expectation:** The TOE shall generate, store and use the selected TOE-managed cryptographic secret under the specified hardware-based keystore mechanism. The expected result is a protected keystore state and successful use only through the authorised TOE operation. • **Test preparation:** Prepare the TOE on a supported device with the required hardware-based keystore capability. Select one TOE-managed secret class and one operation proving generation and later use of that secret. • **Testing steps:** The tester records the TOE version, platform, OS version, device model, selected secret class, keystore mechanism, observation method and evidence type. The tester triggers secret generation, verifies the hardware-keystore evidence, and performs the selected operation. • **Test result:** The actual result is the keystore-use behaviour: hardware-keystore protected, ambiguous, inconsistent, software-only, or not evidenced. The result shall describe evidence category and operation outcome without recording secret values or reusable artefacts. • **Assessment (pass/fail):** PASS if the selected secret is generated/stored/used under the specified hardware-based keystore mechanism and the operation succeeds through that mechanism. FAIL if the secret uses a software-only or undocumented path, or if hardware-keystore protection lacks adequate evidence.
TE.HrdwSec1.02	• **Test description with test expectation:** The TOE shall not complete normal installation, activation or protected operation when the required hardware-based keystore mechanism is unavailable or unsuitable. The expected result is safe blocking, unsupported-device indication, restricted operation or another documented safe outcome. • **Test preparation:** Prepare one vendor-supported negative device/configuration scenario from VE.HrdwSec1.02. Identify the lifecycle step or wallet operation that must not proceed without the required keystore condition. • **Testing steps:** The tester records the TOE version, platform, OS version, device/test configuration, negative keystore condition, expected safe outcome and evidence type. The tester attempts installation, activation or the selected protected operation under that condition. • **Test result:** The actual result is the unsupported-keystore behaviour: blocked, restricted, unsupported-device error, safe recovery, normal operation, or not evidenced. The result shall state whether normal wallet operation became available. • **Assessment (pass/fail):** PASS if the TOE prevents normal installation, activation or protected operation when the required hardware-keystore condition is not met. FAIL if the TOE operates normally using an unspecified or non-conformant fallback.

Element	Content
TE.HrdwSec1.03	• **Test description with test expectation:** The TOE shall not expose the selected cryptographic secret or reusable equivalent through supported TOE paths and shall process it only through the defined hardware-keystore path. The expected result is no clear/exportable secret and operation only through keystore-controlled use. • **Test preparation:** Use the TOE and selected secret class from TE.HrdwSec1.01. Identify supported paths that could reveal or transfer secret material, such as export, backup, diagnostics, recovery, key-management view or test diagnostic output. • **Testing steps:** The tester records the selected secret class, supported path, prohibited content category and evidence type. The tester invokes the selected paths and performs one operation requiring use of the selected secret. • **Test result:** The actual result is the secret-handling behaviour: non-exportable/non-visible, keystore-controlled use, denied use, exposed, exported, reusable, or not evidenced. The result shall describe representation categories without copying key material. • **Assessment (pass/fail):** PASS if no selected supported path exposes the secret or reusable equivalent and the operation uses the defined hardware-keystore path. FAIL if the secret is visible, exportable, reusable outside that path, or processed through an undocumented software-only path.
Test Verdict	PASS iff TE.HrdwSec1.01, TE.HrdwSec1.02 and TE.HrdwSec1.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify TOE-managed cryptographic secrets, required hardware-keystore mechanisms, supported devices, negative keystore scenarios, observation methods or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.HrdwSec2 (FITCEM PP: Additional security measures)

Element	Content
Additional security measures — AS.HrdwSec2	**The TOE shall not store program data or application content unprotected in the end device's memory. If the TOE stores data on the end device, it must be encrypted unless it is already protected by other means, such as encryption by the WSCA. The key used to encrypt the data must be stored in the end device's key store.**
Required vendor information	
VE.HrdwSec2.01	The vendor shall identify program data and application-content data classes stored by the TOE in the evaluated configuration. The vendor shall specify, for each class, whether protection is provided by TOE encryption, WSCA protection, platform protection or another defined mechanism.
VE.HrdwSec2.02	The vendor shall provide SUG/test instructions, test accounts, test data and conformance-level observation methods allowing the tester to create, store, use and observe protected stored data. Where TOE encryption is used, the vendor shall also define how use of the end-device keystore for the encryption key can be observed without exposing key material.
Required test procedures	
TE.HrdwSec2.01	• **Test description with test expectation:** The TOE shall store the selected program data or application-content data class only in a protected state. The expected result is an observable protected representation and successful later use only through the authorized TOE path. • **Test preparation:** Prepare the TOE in the evaluated configuration, one authenticated test User and one selected stored-data class from VE.HrdwSec2.01. Prepare the vendor-approved observation method for determining whether the selected class is protected, encrypted or protected by other means. • **Testing steps:** The tester records the TOE version, platform, OS version, selected stored-data class, claimed protection mechanism and approved observation method in the test evidence. The tester creates the selected data through a normal TOE flow, observes the protected-storage state through the approved method, and then uses the stored data through the authorized TOE path. • **Test result:** The actual result is the observed storage-protection behaviour: the selected data is shown as protected/encrypted/protected-by-WSCA and remains usable through the authorized TOE path, or it is observable as unprotected, or the protection state is ambiguous. The result shall describe representation categories and operation outcome without copying personal data, secrets, keys or reusable artefacts.

Element	Content
	• **Assessment (pass/fail):** PASS if the selected stored data is not stored unprotected and is usable only through the authorized TOE path. FAIL if the selected data is stored or observable in clear/unprotected form, or if the protected state cannot be demonstrated by the approved method.
TE.HrdwSec2.02	• **Test description with test expectation:** The TOE shall apply the declared protection mechanism for the selected stored-data class, either TOE-side encryption with a keystore-protected key or another specified protection mechanism such as WSCA protection. The expected result is that the observed behaviour matches the declared protection model. • **Test preparation:** Select one stored-data class protected by TOE encryption or, where applicable, one stored-data class claimed to be protected by WSCA or another means. Identify the expected observable evidence for that protection model from VE.HrdwSec2.01 and VE.HrdwSec2.02. • **Testing steps:** The tester records the TOE version, platform, selected data class, declared protection model and expected evidence in the test evidence. The tester creates or retrieves the selected data, observes the approved evidence for the declared model, and attempts the normal operation that uses the data. • **Test result:** The actual result is the observed protection-model behaviour: the data behaves consistently with TOE encryption, WSCA protection or another declared mechanism, or the observed state contradicts the declared model. The result shall describe the model category and visible evidence without exposing protected contents or keys. • **Assessment (pass/fail):** PASS if the selected data class is protected according to the declared model and no unprotected fallback is observed. FAIL if the TOE stores the data unprotected, relies on an undocumented protection model, or falls back to clear local storage.
TE.HrdwSec2.03	• **Test description with test expectation:** Where the TOE encrypts stored data, the encryption key shall be stored in the end device's keystore. The expected result is observable keystore-backed key use and safe handling if the required keystore condition is unavailable or invalid. • **Test preparation:** Select one TOE-encrypted stored-data class from VE.HrdwSec2.01 and identify the device-keystore observation method from VE.HrdwSec2.02. Prepare, where available, a vendor-supported negative keystore condition or unsupported-keystore scenario. • **Testing steps:** The tester records the TOE version, platform, OS version, selected data class, keystore mechanism, observation method and negative keystore scenario if used in the test evidence. The tester stores and uses the selected data under the normal keystore condition, then applies the supported negative keystore condition and observes whether the TOE continues, blocks, recovers or enters a documented safe state. • **Test result:** The actual result is the observed keystore-key behaviour: the encryption key use is evidenced as keystore-backed and data use succeeds normally, while the negative keystore condition is safely blocked/recovered; or the TOE continues with an undocumented software-only key or clear-storage fallback. The result shall describe keystore-state and operation categories without recording key material or encrypted object contents. • **Assessment (pass/fail):** PASS if TOE-side encryption uses a key stored in the end-device keystore and no non-conformant fallback is observed. FAIL if encryption uses an undocumented software-only key, the key storage cannot be evidenced by the approved method, or the TOE stores/uses data unprotected when keystore protection is unavailable.
Test Verdict	PASS iff TE.HrdwSec2.01, TE.HrdwSec2.02 and TE.HrdwSec2.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify stored data classes, protection mechanisms, keystore-key evidence, negative keystore scenarios or observation methods.

Table — VE and TE of AS.HrdwSec3 (FITCEM PP: Additional security measures)

Element	Content
Additional security measures — AS.HrdwSec3	**The TOE shall implement appropriate measures for the confidential storage of personal data on end devices.**
Required vendor information	
VE.HrdwSec3.01	The vendor shall identify all personal-data classes stored on the end device by the TOE in the evaluated configuration and specify the confidentiality measure applied to each class. The vendor shall define

Element	Content
	which measures are observable through UI/platform-supported means and which require documentation, configuration or source-code evidence.
VE.HrdwSec3.02	The vendor shall provide SUG/test instructions, test accounts, test personal data, supported observation paths and one conformance-level confidentiality-negative scenario. Implementation evidence may include storage design, data-flow mapping, protected-storage configuration, keystore/API mapping, WSCA-protection mapping, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.HrdwSec3.01	• **Test description with test expectation:** The TOE shall store the selected personal-data class on the end device under the vendor-defined confidentiality measure. The expected result is an observable or sufficiently evidenced confidential/protected storage state. • **Test preparation:** Prepare the TOE, authenticated test User and one selected personal-data class from VE.HrdwSec3.01. Prepare the relevant observation method or implementation evidence from VE.HrdwSec3.02. • **Testing steps:** The tester records the TOE version, platform, OS version, selected personal-data class, claimed confidentiality measure, observation method and evidence type. The tester creates or retrieves the selected test personal data through a normal TOE flow and verifies the protected state. • **Test result:** The actual result is the personal-data storage state: protected, unprotected, ambiguous, inconsistent or evidenced only. The result shall describe representation categories without copying personal values, secrets or reusable artefacts. • **Assessment (pass/fail):** PASS if the selected personal-data class is stored under the defined confidentiality measure and the evidence is consistent with the evaluated build. FAIL if the data is stored or observable in clear/unprotected form, or if the claimed confidentiality state lacks adequate evidence.
TE.HrdwSec3.02	• **Test description with test expectation:** The TOE shall not disclose stored personal data through selected supported local, platform, diagnostic, export, backup, cache or equivalent paths outside authorised/intended use. The expected result is denial, redaction, omission, protected representation or another documented safe outcome. • **Test preparation:** Use the selected personal-data class from TE.HrdwSec3.01 and select one supported path from VE.HrdwSec3.02. Define prohibited personal-data content categories before execution. • **Testing steps:** The tester records the selected path, authorisation context, prohibited content categories and evidence type. The tester invokes the selected path and observes UI, platform response, export category, diagnostic output, test backend result or implementation evidence. • **Test result:** The actual result is the disclosure behaviour: blocked, authorisation required, redacted, omitted, metadata-only, exposed or evidenced only. The result shall describe content categories and access decision, not personal values. • **Assessment (pass/fail):** PASS if the selected path does not disclose stored personal data outside authorised/intended TOE use and the evidence is consistent. FAIL if personal data is visible, exportable, backed up, logged, cached, diagnosed or otherwise available in clear/reusable form contrary to VE.HrdwSec3.01.
TE.HrdwSec3.03	• **Test description with test expectation:** The TOE shall not store, expose or use personal data in unprotected form when the required confidentiality measure is unavailable, invalid or outside the evaluated configuration. The expected result is safe blocking, recovery, reinitialisation, re-issuance requirement, re-authentication or another documented safe outcome. • **Test preparation:** Select one personal-data class and one vendor-supported confidentiality-negative scenario from VE.HrdwSec3.02. The scenario shall not require uncontrolled storage tampering, root/jailbreak, memory inspection or reverse engineering. • **Testing steps:** The tester records the selected personal-data class, negative scenario, expected safe outcome and evidence type. The tester applies the scenario and attempts to view, store, present, export or otherwise use the affected personal data through the normal TOE path. • **Test result:** The actual result is the confidentiality-failure handling: rejected, recovery required, re-authenticated, safe state, clear exposure, unprotected storage, or use as valid data. The result shall describe the safe state or operation outcome without copying personal data or protection values. • **Assessment (pass/fail):** PASS if the TOE does not store, expose or use affected personal data in unprotected form and follows the documented safe handling path. FAIL if the TOE falls back to clear

Element	Content
	storage, discloses personal data or processes affected personal data as valid despite the failed confidentiality condition.
Test Verdict	PASS iff TE.HrdwSec3.01, TE.HrdwSec3.02 and TE.HrdwSec3.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify personal-data classes, confidentiality measures, supported observation paths, negative confidentiality scenarios or implementation evidence for non-UI-observable controls.

CRYPTO — Cryptography

Table — VE and TE of AS.Crypto1 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto1	**Cryptographic algorithms and mechanisms used by the TOE should implement the recommended procedures and configurations specified in [ECCG-ACM] and, where applicable, [TR-03116-4]. Other procedures may not be used unless they are explicitly permitted. The current version of [ECCG-ACM] or [TR-03116-4] applies in each case. In the event of contradictions between [ECCG-ACM] and [TR-03116-4], [ECCG-ACM] takes priority.**
Required vendor information	
VE.Crypto1.01	The vendor shall provide a Crypto-Annex for the evaluated TOE configuration identifying all cryptographic algorithms, mechanisms, protocols, libraries/providers and relevant parameters used by the TOE. The Crypto-Annex shall include purpose, interface/function, standard reference, selected options, key sizes, curves/groups, modes, hash functions, signature/MAC/KDF/RNG mechanisms, dependency source and applicable TOE version/build.
VE.Crypto1.02	The vendor shall provide a Crypto1 conformance mapping against the current [ECCG-ACM] and, where applicable, [TR-03116-4]. The mapping shall identify recommended mechanisms, explicitly permitted alternatives, deviations, ECCG-ACM priority decisions, observation methods and implementation evidence for non-UI-observable mechanisms, such as configuration snippets, API mapping, build configuration, dependency/SBOM references, limited source-code excerpts or vendor test logs.
Required test procedures	
TE.Crypto1.01	• **Test description with test expectation:** The vendor Crypto-Annex shall identify all TOE cryptographic mechanisms relevant to Crypto1. The expected result is a complete, versioned and TOE-build-specific inventory mapped to TOE functions and interfaces. • **Test preparation:** Prepare the Crypto-Annex, ST, evaluated TOE build, interface list and conformance mapping from VE.Crypto1.01 and VE.Crypto1.02. Prepare a checklist covering mechanism name, purpose, parameters, selected options, provider/library, interface/function and evidence type. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, Crypto-Annex version and checklist categories. The tester reviews whether each declared mechanism is mapped to a TOE function/interface and supported by documentation or implementation evidence. • **Test result:** The actual result is the inventory coverage: complete and mapped, incomplete, ambiguous, inconsistent, or not evidenced. The result shall identify missing mechanism categories or unclear mappings without reproducing secrets or proprietary implementation details unnecessarily. • **Assessment (pass/fail):** PASS if the Crypto-Annex provides a complete and evidenced inventory sufficient to assess Crypto1 for the evaluated build. FAIL if mechanisms, parameters, provider/library sources, standards, applicability decisions or TOE-function mappings are missing, ambiguous or not attributable to the evaluated TOE.
TE.Crypto1.02	• **Test description with test expectation:** Declared TOE mechanisms shall implement recommended procedures and configurations from the current [ECCG-ACM] and, where applicable, [TR-03116-4]. The expected result is that each selected mechanism is recommended or explicitly permitted. • **Test preparation:** Select representative declared mechanisms from VE.Crypto1.01, covering at least one communication and one non-communication mechanism where both exist. Prepare applicable [ECCG-ACM]/[TR-03116-4] versions, vendor mapping and implementation evidence. • **Testing steps:** The tester records the selected mechanisms, parameters, reference versions, recommendation entries and evidence type. The tester compares selected mechanisms and options against the references, checks claimed exceptions, and applies ECCG-ACM priority where a contradiction exists. • **Test result:** The actual result is the standards-conformance status: recommended, explicitly permitted, inconsistent, unresolved, or not evidenced. The result shall identify the affected mechanism and conformance category without including key material or implementation secrets. • **Assessment (pass/fail):** PASS if all selected mechanisms are recommended by the applicable current reference or explicitly permitted and justified. FAIL if any selected mechanism uses a non-recommended, prohibited or unresolved procedure/configuration without explicit permission, or if conflict resolution does not give priority to [ECCG-ACM].

Element	Content
TE.Crypto1.03	• **Test description with test expectation:** Observable or evidenced TOE cryptographic behaviour shall be consistent with the Crypto-Annex and Crypto1 conformance mapping. The expected result is no undocumented, deprecated, prohibited or non-permitted mechanism in the evaluated configuration. • **Test preparation:** Select one or more cryptographic flows or artefacts from VE.Crypto1.02, such as TLS handshake, OpenID4VP/OpenID4VCI artefact, presentation response, issuance response, signature object, certificate metadata or approved diagnostic output. For non-observable mechanisms, prepare the corresponding implementation evidence. • **Testing steps:** The tester records selected flows/artefacts, expected algorithms/parameters, observation method and evidence type. The tester executes observable flows or reviews approved implementation evidence, then compares the result with the Crypto-Annex and conformance mapping. • **Test result:** The actual result is the runtime/evidence consistency: documented and permitted, undocumented, weaker/disallowed, inconsistent, or insufficiently evidenced. The result shall describe algorithm/parameter categories and conformance status, not private keys, tokens or reusable protected artefacts. • **Assessment (pass/fail):** PASS if selected runtime behaviour or implementation evidence matches documented recommended or explicitly permitted mechanisms. FAIL if the TOE negotiates, emits, accepts, configures or evidences undocumented, prohibited, deprecated or non-permitted cryptographic mechanisms.
Test Verdict	PASS iff TE.Crypto1.01, TE.Crypto1.02 and TE.Crypto1.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify cryptographic mechanisms, providers/libraries, applicable reference versions, parameters, conformance mapping, permitted deviations, observation methods or implementation evidence for non-UI-observable mechanisms.

Table — VE and TE of AS.Crypto2 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto2	**The TOE should encrypt communication channels to external entities using state-of-the-art technology (TLS, as a minimum).**
Required vendor information	
VE.Crypto2.01	The vendor shall provide a communication-channel inventory for the evaluated TOE configuration. The inventory shall identify all TOE communication channels to external entities, protocol used, endpoint/entity type, data category, encryption mechanism, TLS/profile version, cipher-suite or configuration policy, and TOE function using the channel.
VE.Crypto2.02	The vendor shall provide the evaluated TOE build, test accounts, test endpoints and observation instructions allowing the tester to observe selected encrypted communication channels. The vendor shall define expected behaviour for encrypted communication, plaintext/unprotected channel attempts, and deprecated or below-policy protocol negotiation attempts.
Required test procedures	
TE.Crypto2.01	• **Test description with test expectation:** The vendor communication-channel inventory shall identify all TOE communication channels to external entities and the encryption mechanism used for each channel. The expected result is that each external channel is mapped to TLS or another state-of-the-art encrypted channel mechanism, with parameters sufficient for conformance assessment. • **Test preparation:** Prepare the Crypto-Annex, communication-channel inventory, evaluated TOE configuration and vendor TLS/profile policy from VE.Crypto2.01. Prepare a checklist covering external entity, TOE function, protocol, endpoint type, data category, TLS/profile version, encryption policy and non-applicability claims. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, inventory version, TLS/profile policy version and checklist categories in the test evidence. The tester reviews the inventory and maps each declared external communication channel to its documented encryption mechanism and applicable TOE function. • **Test result:** The actual result is the observed documentation coverage: all external communication channels are identified and mapped to an encrypted mechanism, some channels are incomplete or

Element	Content
	ambiguous, or one or more channels lack encryption information. The result shall identify missing or unclear channel categories without reproducing secrets, endpoint credentials or proprietary network details unnecessarily. • **Assessment (pass/fail):** PASS if all TOE communication channels to external entities are documented and each channel has a defined state-of-the-art encrypted communication mechanism. FAIL if any external channel is missing, ambiguous, undocumented, or lacks a defined encrypted-channel mechanism.
TE.Crypto2.02	• **Test description with test expectation:** The TOE shall use encrypted communication for selected external communication channels. The expected result is that the selected channel negotiates TLS or an equivalent state-of-the-art encrypted channel according to the documented policy before any sensitive application data is transmitted. • **Test preparation:** Select representative external channels from VE.Crypto2.01, including at least one Wallet Provider/back-end channel and, where applicable, one ecosystem or third-party interface channel. Prepare test accounts, test endpoints, network observation method and non-sensitive test data according to VE.Crypto2.02. • **Testing steps:** The tester records the TOE version, platform, selected channels, endpoint/entity type, expected protocol, expected TLS/profile version and observation method in the test evidence. The tester executes the selected TOE flows and observes the negotiated protocol, TLS/profile version, cipher/configuration category, endpoint scheme and whether application data is sent only after the encrypted channel is established. • **Test result:** The actual result is the observed encrypted-channel behaviour: the selected channels use the documented encrypted protocol and policy, use an undocumented or weaker configuration, or transmit over an unprotected/plaintext channel. The result shall describe protocol and configuration categories without recording credentials, tokens, personal data or reusable protected artefacts. • **Assessment (pass/fail):** PASS if each selected external channel uses the documented state-of-the-art encrypted channel before transmitting application data. FAIL if any selected channel uses plaintext, sends sensitive data before channel encryption is established, or negotiates an undocumented/below-policy configuration.
TE.Crypto2.03	• **Test description with test expectation:** The TOE shall not use plaintext or below-policy encrypted communication when communicating with external entities. The expected result is connection refusal, safe error, restricted operation or another documented safe outcome when an unencrypted or below-policy channel is offered. • **Test preparation:** Select one external channel from TE.Crypto2.02 for which the vendor provides a safe negative test endpoint or controlled test configuration. Prepare a plaintext/unprotected or below-policy protocol negotiation scenario, such as HTTP-only endpoint, cleartext WebSocket, obsolete TLS version, disallowed cipher/profile, or equivalent vendor-defined negative case. • **Testing steps:** The tester records the TOE version, platform, selected channel, negative protocol scenario, expected safe outcome and observation method in the test evidence. The tester attempts the selected TOE flow against the negative endpoint or configuration and observes the TOE UI, network result, diagnostic output, backend response or approved test-endpoint result. • **Test result:** The actual result is the observed fallback/downgrade handling: the TOE refuses the connection, shows a safe error, restricts the operation, or proceeds using plaintext or below-policy encryption. The result shall describe the visible state, protocol category and whether any application data was transmitted. • **Assessment (pass/fail):** PASS if the TOE prevents plaintext or below-policy communication and does not transmit application data over the non-conformant channel. FAIL if the TOE falls back to plaintext, accepts below-policy encrypted communication without documented permission, or sends application data over a non-conformant channel.
Test Verdict	PASS iff TE.Crypto2.01, TE.Crypto2.02 and TE.Crypto2.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify external communication channels, encryption mechanisms, TLS/profile policy, selected test endpoints, expected negative-channel behaviour or observation methods.

Table — VE and TE of AS.Crypto3 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto3	**The TOE shall authenticate communication entities prior to the transmission of sensitive data.**
Required vendor information	
VE.Crypto3.01	The vendor shall provide a communication-entity authentication specification for the evaluated TOE configuration. The specification shall identify communication entities, sensitive-data categories, channels/functions where sensitive data may be transmitted, authentication mechanisms, trust anchors or credentials used, success/failure conditions, and the required ordering rule that authentication occurs before sensitive-data transmission.
VE.Crypto3.02	The vendor shall provide the evaluated TOE build, test accounts, test endpoints/entities, test credentials and observation instructions allowing the tester to verify successful entity authentication and safe handling of unauthenticated or invalid entities. The evidence shall include at least one valid-entity scenario and one invalid, unauthenticated, expired, untrusted or mismatched-entity scenario.
Required test procedures	
TE.Crypto3.01	• **Test description with test expectation:** The vendor specification shall identify communication entities and authentication mechanisms that apply before transmission of sensitive data. The expected result is a complete mapping of selected sensitive-data communication flows to prior entity-authentication requirements. • **Test preparation:** Prepare the Crypto-Annex, communication-channel inventory, entity-authentication specification and evaluated TOE configuration according to VE.Crypto3.01. Prepare a checklist covering entity type, TOE function, channel, sensitive-data category, authentication mechanism, credential/trust source, success condition, failure condition and ordering rule. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, specification version and checklist categories in the test evidence. The tester reviews the specification and maps each selected sensitive-data flow to the required authenticated communication entity and the mechanism that must complete before sensitive data is transmitted. • **Test result:** The actual result is the observed documentation coverage: the selected flows are mapped to prior entity authentication, are mapped incompletely or ambiguously, or lack required entity/authentication/sensitive-data ordering information. The result shall identify missing or unclear categories without exposing credentials, trust secrets or proprietary endpoint details unnecessarily. • **Assessment (pass/fail):** PASS if the specification clearly identifies communication entities, sensitive-data flows and authentication mechanisms that must complete before transmission. FAIL if communication entities, sensitive-data categories, authentication mechanisms or ordering rules are missing, ambiguous or not attributable to the evaluated TOE.
TE.Crypto3.02	• **Test description with test expectation:** The TOE shall authenticate the selected valid communication entity before transmitting sensitive data in the selected flow. The expected result is that sensitive data is transmitted only after the documented entity-authentication success condition is reached. • **Test preparation:** Select one representative sensitive-data communication flow from VE.Crypto3.01, such as issuance, presentation, backend synchronisation, status check, WSCA/WSCD-related interface or peer-wallet interaction. Prepare a valid authenticated test entity, test account, non-production sensitive test data and approved observation method from VE.Crypto3.02. • **Testing steps:** The tester records the TOE version, platform, selected entity, selected flow, sensitive-data category, expected authentication mechanism and observation method in the test evidence. The tester executes the selected flow with the valid entity and observes the authentication event, success state, protocol metadata, signed request/response status, certificate/trust status, endpoint result or approved diagnostic output before observing any sensitive-data transmission. • **Test result:** The actual result is the observed valid-entity behaviour: entity authentication succeeds before sensitive data is transmitted, authentication is ambiguous, or sensitive data is transmitted before the authentication success condition. The result shall describe authentication status, ordering category and data-transmission category without recording real personal data, secrets, tokens or reusable protected artefacts. • **Assessment (pass/fail):** PASS if the TOE authenticates the valid communication entity before transmitting sensitive data and the transmission occurs only after the documented success condition. FAIL if sensitive data is transmitted before authentication, if the authentication state cannot be

Element	Content
	determined by the approved method, or if the observed mechanism differs materially from the specification.
TE.Crypto3.03	• **Test description with test expectation:** The TOE shall not transmit sensitive data to or through an unauthenticated, invalid, untrusted or mismatched communication entity. The expected result is safe refusal, user-safe error, restricted operation, or another documented safe outcome before sensitive data is transmitted. • **Test preparation:** Use the selected flow from TE.Crypto3.02 or another representative sensitive-data flow from VE.Crypto3.01. Prepare an invalid, unauthenticated, expired, untrusted or mismatched test entity according to VE.Crypto3.02, and define the expected safe failure outcome before testing. • **Testing steps:** The tester records the TOE version, platform, selected flow, invalid-entity scenario, sensitive-data category, expected failure condition and observation method in the test evidence. The tester attempts the selected flow against the invalid entity and observes TOE UI, protocol result, endpoint response, diagnostic output, trust/certificate/signature status or backend/test-service decision. • **Test result:** The actual result is the observed invalid-entity behaviour: the TOE refuses the flow before transmitting sensitive data, displays a safe authentication/trust error, restricts the operation, or transmits sensitive data despite failed or absent entity authentication. The result shall describe the failure category and whether any sensitive-data transmission occurred, without storing real sensitive payloads. • **Assessment (pass/fail):** PASS if the TOE prevents sensitive-data transmission when entity authentication fails or is absent. FAIL if the TOE transmits sensitive data to an unauthenticated, invalid, untrusted or mismatched entity, or silently falls back to an unauthenticated communication path.
Test Verdict	PASS iff TE.Crypto3.01, TE.Crypto3.02 and TE.Crypto3.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify communication entities, sensitive-data flows, authentication mechanisms, ordering rules, valid/invalid test entities or observation methods.

Table — VE and TE of AS.Crypto4 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto4	**The TOE's cryptographic methods should be designed so that they can be adapted to new cryptographic findings, e.g. by changing cryptographic primitives or key lengths.**
Required vendor information	
VE.Crypto4.01	The vendor shall provide a crypto-agility design specification for the evaluated TOE configuration. The specification shall identify adaptable cryptographic mechanisms, adaptation method, configurable or replaceable parameters, affected TOE functions/interfaces, dependencies on platform/WSCA/WSCD/backend components, and which evidence is observable or requires documentation, configuration or source-code evidence.
VE.Crypto4.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints and one controlled adaptation scenario for the evaluated TOE build. Implementation evidence may include Crypto-Annex references, crypto-policy mapping, configuration snippets, provider/library mapping, update or migration description, limited source-code excerpts or vendor test logs.
Required test procedures	
TE.Crypto4.01	• **Test description with test expectation:** The TOE shall have a documented method for adapting selected cryptographic methods to new cryptographic findings. The expected result is a clear mapping between mechanisms, adaptation methods and affected TOE functions. • **Test preparation:** Prepare the Crypto-Annex, crypto-agility specification, evaluated TOE build and dependency information from VE.Crypto4.01. Prepare a checklist covering mechanism, parameter/profile, adaptation method, dependency and limitation. • **Testing steps:** The tester records the TOE version, build identifier, selected mechanisms, adaptation method and evidence type. The tester reviews whether selected cryptographic methods have a documented and evidenced adaptation path.

Element	Content
	• **Test result:** The actual result is the design coverage: complete, partial, ambiguous, missing or not evidenced. The result shall identify affected mechanism categories without exposing secrets or unnecessary proprietary details. • **Assessment (pass/fail):** PASS if selected cryptographic methods have a documented and evidenced adaptation path for the evaluated TOE. FAIL if adaptation methods, affected functions, parameters, dependencies or limitations are missing, ambiguous or not attributable to the evaluated build.
TE.Crypto4.02	• **Test description with test expectation:** The TOE shall support the vendor-defined controlled adaptation scenario for one selected cryptographic method. The expected result is application of the updated permitted parameter, profile or mechanism, or a documented safe migration/update state. • **Test preparation:** Select one adaptation scenario from VE.Crypto4.02, such as key-length change, algorithm-profile change, TLS/JOSE/COSE policy change, provider replacement or application update path. Prepare the expected post-adaptation state and observation method. • **Testing steps:** The tester records the selected method, pre-adaptation setting, adaptation method, expected post-adaptation setting and evidence type. The tester applies or reviews the controlled adaptation scenario and executes the affected TOE function where observable. • **Test result:** The actual result is the adaptation behaviour: updated and operational, safely migrated, safely restricted, unchanged, undefined, or not evidenced. The result shall describe algorithm/profile/parameter categories without recording keys, tokens or protected payloads. • **Assessment (pass/fail):** PASS if the selected adaptation scenario changes or migrates the cryptographic method as documented and the affected function operates or fails safely. FAIL if the TOE ignores the adaptation, applies an undocumented mechanism, or enters an undefined state without documented safe handling.
TE.Crypto4.03	• **Test description with test expectation:** After adaptation, the TOE shall not use, negotiate or accept a deprecated, disabled or below-policy cryptographic mechanism for the affected function. The expected result is refusal, migration, restricted operation or use of the updated permitted mechanism only. • **Test preparation:** Use the adapted TOE state from TE.Crypto4.02 or an equivalent vendor-defined post-adaptation state. Prepare a negative scenario involving the pre-adaptation, deprecated, disabled or below-policy mechanism. • **Testing steps:** The tester records the affected function, disabled/deprecated mechanism, expected safe outcome and evidence type. The tester executes the negative scenario or reviews evidence showing post-adaptation enforcement. • **Test result:** The actual result is the deprecated-mechanism handling: refused, migrated, restricted, accepted, fallback used, or not evidenced. The result shall state whether the affected TOE function completed under the old or below-policy mechanism. • **Assessment (pass/fail):** PASS if the TOE no longer uses or accepts the disabled/deprecated mechanism after adaptation and follows the documented safe outcome. FAIL if the TOE falls back to the old mechanism, accepts below-policy cryptography, or cannot evidence the post-adaptation policy.
Test Verdict	PASS iff TE.Crypto4.01, TE.Crypto4.02 and TE.Crypto4.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify adaptable mechanisms, adaptation methods, affected functions, dependencies, post-adaptation policy, negative deprecated-mechanism handling or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Crypto5 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto5	**The key material stored by the Wallet Instance shall only be used after successful user authentication and authorization.**
Required vendor information	
VE.Crypto5.01	The vendor shall provide a key-material use specification for the evaluated TOE configuration. The specification shall identify key material stored by the Wallet Instance, related TOE functions, required user-authentication mechanism, authorization decision, session validity rules, timeout/re-authentication conditions, and expected failure behaviour.

Element	Content
VE.Crypto5.02	The vendor shall provide the evaluated TOE build, test accounts, test key material or test credentials, test endpoints and observation methods allowing the tester to verify successful key use after authentication/authorization and denied key use before authentication, after session expiry, or when authorization is absent or cancelled.
Required test procedures	
TE.Crypto5.01	• **Test description with test expectation:** The vendor specification shall identify Wallet-Instance-stored key material and the authentication/authorization conditions required before its use. The expected result is a complete mapping of selected key material to TOE functions, user-authentication mechanism, authorization decision and failure handling. • **Test preparation:** Prepare the Crypto-Annex, key-material use specification, evaluated TOE configuration and user-authentication policy from VE.Crypto5.01. Prepare a checklist covering key-material class, storage location category, TOE function, required authentication, required authorization, session validity, timeout and failure behaviour. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, specification version and checklist categories in the test evidence. The tester reviews the specification and maps each selected Wallet-Instance-stored key-material class to the TOE operation where it is used and to the required authentication/authorization condition. • **Test result:** The actual result is the observed documentation coverage: selected key-material classes are mapped to authentication and authorization conditions, are mapped incompletely or ambiguously, or lack required session/failure rules. The result shall identify missing or unclear categories without exposing key values, secrets or proprietary implementation details unnecessarily. • **Assessment (pass/fail):** PASS if the specification clearly defines key-material use conditions for the evaluated TOE, including both user authentication and authorization. FAIL if key material, affected functions, authentication conditions, authorization conditions or failure handling are missing, ambiguous or not attributable to the evaluated TOE.
TE.Crypto5.02	• **Test description with test expectation:** The TOE shall use the selected Wallet-Instance-stored key material only after successful user authentication and authorization of the selected operation. The expected result is that the key-dependent operation succeeds only after both conditions are satisfied. • **Test preparation:** Select one key-dependent TOE operation from VE.Crypto5.01, such as signing, decryption, presentation, credential operation, key agreement or protected backend request. Prepare a valid test User, valid authentication factors, required authorization step, test endpoint and non-production data according to VE.Crypto5.02. • **Testing steps:** The tester records the TOE version, platform, selected key-material class, selected key-dependent operation, expected authentication condition, expected authorization condition and observation method in the test evidence. The tester performs the user-authentication step, authorizes the selected operation through the TOE-supported path, and observes the resulting key-dependent artefact, endpoint result, protocol metadata, UI status or approved diagnostic output. • **Test result:** The actual result is the observed authorised-use behaviour: authentication succeeds, authorization is obtained, and the selected key-dependent operation completes as specified; or the operation completes without one of the required conditions; or the authentication/authorization state is ambiguous. The result shall describe status and operation categories without recording private keys, secrets, tokens or reusable cryptographic artefacts. • **Assessment (pass/fail):** PASS if the selected key material is used only after successful user authentication and authorization, and the resulting operation matches the documented TOE behaviour. FAIL if the operation uses key material before authentication, without authorization, or with an authentication/authorization state that cannot be evidenced by the approved method.
TE.Crypto5.03	• **Test description with test expectation:** The TOE shall prevent use of selected Wallet-Instance-stored key material when user authentication is absent/expired or when authorization is absent, denied or cancelled. The expected result is safe refusal, re-authentication request, authorization request, operation cancellation or another documented safe outcome before key use occurs. • **Test preparation:** Use the selected key-dependent operation from TE.Crypto5.02. Prepare at least one unauthenticated or expired-session scenario and one no-authorization or cancelled-authorization scenario according to VE.Crypto5.02. • **Testing steps:** The tester records the TOE version, platform, selected key-material class, selected operation, negative scenario, expected safe outcome and observation method in the test evidence. The tester attempts the selected operation before authentication, after logout/lock/timeout where

Element	Content
	applicable, and after denying or cancelling the required authorization step, then observes the TOE UI, endpoint result, protocol artefact, operation status or approved diagnostic output. • **Test result:** The actual result is the observed denied-use behaviour: the TOE refuses the operation, requests authentication or authorization, cancels safely, or uses key material despite absent/expired authentication or absent/denied authorization. The result shall state whether any key-dependent artefact, signature, decryption, presentation, credential operation or protected request was produced. • **Assessment (pass/fail):** PASS if no selected key material is used unless both successful user authentication and required authorization are present. FAIL if the TOE produces a key-dependent result in an unauthenticated, expired, cancelled or unauthorized state, or silently reuses a previous authorization beyond the documented validity rule.
Test Verdict	PASS iff TE.Crypto5.01, TE.Crypto5.02 and TE.Crypto5.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify Wallet-Instance-stored key material, related TOE functions, authentication conditions, authorization conditions, session validity rules, negative scenarios or observation methods.

Table — VE and TE of AS.Crypto6 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto6	**The TOE shall protect key material stored, processed or otherwise managed by the Wallet Instance against disclosure, unauthorised access, unauthorised export, unauthorised backup/migration and unintended reuse.**
Required vendor information	
VE.Crypto6.01	The vendor shall provide a key-material protection specification for the evaluated TOE configuration. The specification shall identify TOE-managed key-material classes, lifecycle states, protection measures, storage/processing location category, access rules, export/backup/migration rules, deletion/invalidation handling, and TOE/WSCA/WSCD/platform boundary.
VE.Crypto6.02	The vendor shall provide SUG/test instructions, test accounts, test credentials or test key material, test endpoints and approved observation methods allowing the tester to verify selected key protection without exposing private keys, symmetric keys, seeds, tokens or reusable cryptographic artefacts. Implementation evidence may include Crypto-Annex references, keystore/API mapping, protected-handle evidence, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Crypto6.01	• **Test description with test expectation:** The vendor specification shall identify TOE-managed key material and the protection measures applied to it. The expected result is a clear mapping of selected key-material classes to protection measures, lifecycle states and permitted/forbidden operations. • **Test preparation:** Prepare the Crypto-Annex, key-material protection specification, evaluated TOE build and boundary description from VE.Crypto6.01. Prepare a checklist covering key-material class, protection measure, storage/processing category, access rule and prohibited operations. • **Testing steps:** The tester records the TOE version, build identifier, platform, evaluated configuration, specification version and evidence type. The tester reviews the specification and maps each selected TOE-managed key-material class to its documented protection and boundary assumptions. • **Test result:** The actual result is the specification coverage: complete, partial, ambiguous, missing or inconsistent. The result shall identify missing key-material categories or protection rules without exposing key values or unnecessary proprietary details. • **Assessment (pass/fail):** PASS if TOE-managed key material is clearly identified and protected by documented measures attributable to the evaluated TOE. FAIL if key-material classes, protection measures, prohibited operations or TOE/WSCA/WSCD/platform boundaries are missing, ambiguous or inconsistent.
TE.Crypto6.02	• **Test description with test expectation:** The TOE shall create, store and use the selected TOE-managed key material under the documented protection measure. The expected result is protected/non-exportable state and successful key-dependent operation through the documented path.

Element	Content
	• **Test preparation:** Select one TOE-managed key-material class and one key-dependent TOE operation from VE.Crypto6.01. Prepare the approved observation method or implementation evidence from VE.Crypto6.02. • **Testing steps:** The tester records the selected key-material class, lifecycle state, protection measure, key-dependent operation, observation method and evidence type. The tester triggers key creation, provisioning or activation and performs the selected key-dependent operation. • **Test result:** The actual result is the protected-use behaviour: protected/non-exportable, ambiguous, inconsistent, unprotected, exportable or not evidenced. The result shall describe state and operation categories without recording key values, seeds, private keys, tokens or reusable artefacts. • **Assessment (pass/fail):** PASS if the selected key material is protected as specified and used only through the documented protected path. FAIL if it is visible, exportable, stored unprotected, processed through an undocumented path or not sufficiently evidenced.
TE.Crypto6.03	• **Test description with test expectation:** The TOE shall prevent exposure or unintended reuse of TOE-managed key material through supported export, backup, migration, diagnostic, deletion, invalidation or failure-handling paths. The expected result is denial, redaction, protected representation, safe migration, safe deletion/invalidation or another documented safe outcome. • **Test preparation:** Use the selected key-material class from TE.Crypto6.02 and select one supported lifecycle or negative scenario from VE.Crypto6.02. Define the expected safe outcome before testing. • **Testing steps:** The tester records the selected lifecycle/negative scenario, expected safe outcome, observation method and evidence type. The tester invokes the selected supported path and observes whether key material is exposed, protected, migrated, deleted, invalidated or still usable contrary to the rule. • **Test result:** The actual result is the lifecycle-protection behaviour: denied, protected, safely migrated, deleted/invalidated, exposed, reused, retained or not evidenced. The result shall state whether any key-dependent artefact was produced after deletion/invalidation. • **Assessment (pass/fail):** PASS if the TOE follows the documented safe outcome and does not expose, export, reuse or retain TOE-managed key material contrary to the specification. FAIL if key material is disclosed, exported in clear/reusable form, backed up without protection, reused after deletion/invalidation or protected by an undocumented fallback.
Test Verdict	PASS iff TE.Crypto6.01, TE.Crypto6.02 and TE.Crypto6.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify TOE-managed key material, protection measures, lifecycle states, prohibited operations, TOE/WSCA/WSCD/platform boundary, approved observation methods or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Crypto7 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto7	**The TOE shall ensure that the key material is used only for its intended purpose.**
Required vendor information	
VE.Crypto7.01	The vendor shall provide a key-purpose specification for the evaluated TOE configuration. The specification shall identify each class of TOE-relevant key material, its intended purpose, permitted operations, prohibited operations, permitted protocols/functions, applicable key-usage constraints, algorithm constraints, credential/attestation binding, lifecycle state and TOE/WSCA/WSCD/platform boundary.
VE.Crypto7.02	The vendor shall provide the evaluated TOE build, test accounts, test key material or test credentials, test endpoints and approved observation methods allowing the tester to verify one permitted key-use scenario and one prohibited wrong-purpose scenario without exposing private keys, secrets, seeds or reusable cryptographic artefacts.
Required test procedures	
TE.Crypto7.01	• **Test description with test expectation:** The vendor key-purpose specification shall define the intended purpose and permitted use conditions for selected TOE-relevant key material. The expected

Element	Content
	result is a clear mapping of key-material classes to allowed operations, prohibited operations, protocols/functions and purpose constraints. • **Test preparation:** Prepare the Crypto-Annex, key-purpose specification, evaluated TOE configuration and boundary description according to VE.Crypto7.01. Prepare a checklist covering key-material class, intended purpose, permitted operation, prohibited operation, protocol/function binding, algorithm/key-usage constraint, credential/attestation binding and lifecycle state. • **Testing steps:** The tester records the TOE version, build identifier, evaluated configuration, key-purpose specification version and checklist categories in the test evidence. The tester reviews the specification and maps each selected key-material class to its documented intended purpose and permitted/prohibited use conditions. • **Test result:** The actual result is the observed documentation coverage: selected key-material classes have clear intended-purpose mappings, have incomplete or ambiguous mappings, or lack purpose/use restrictions. The result shall identify missing or unclear purpose categories without exposing key values or proprietary implementation details unnecessarily. • **Assessment (pass/fail):** PASS if selected TOE-relevant key material is clearly mapped to intended purposes and permitted/prohibited use conditions for the evaluated TOE. FAIL if intended purposes, prohibited use cases, protocol/function bindings or boundary assumptions are missing, ambiguous or not attributable to the evaluated TOE.
TE.Crypto7.02	• **Test description with test expectation:** The TOE shall use the selected key material successfully for its documented intended purpose. The expected result is that the selected key-dependent operation completes and the resulting artefact or protocol outcome is consistent with the documented purpose constraints. • **Test preparation:** Select one key-material class and one permitted key-dependent operation from VE.Crypto7.01. Prepare a valid test User, required authentication/authorization state, test credential or attestation, test endpoint and approved observation method according to VE.Crypto7.02. • **Testing steps:** The tester records the TOE version, platform, selected key-material class, intended purpose, selected permitted operation, expected purpose constraints and observation method in the test evidence. The tester executes the selected operation through the normal TOE path and observes the resulting signature, key-agreement result, encryption/decryption result, presentation artefact, credential operation, endpoint response or approved diagnostic output. • **Test result:** The actual result is the observed intended-purpose behaviour: the key-dependent operation completes under the documented purpose constraints, completes with unclear purpose binding, or produces an artefact inconsistent with the declared key purpose. The result shall describe operation and metadata categories without recording private keys, secrets, tokens or reusable protected artefacts. • **Assessment (pass/fail):** PASS if the selected key material is used only for the documented permitted purpose and the observed operation outcome matches the key-purpose specification. FAIL if the selected operation uses the wrong key, produces an artefact outside the documented purpose, or cannot evidence the intended-purpose binding by the approved method.
TE.Crypto7.03	• **Test description with test expectation:** The TOE shall prevent use of selected key material for a purpose other than its documented intended purpose. The expected result is safe refusal, operation cancellation, unsupported-purpose error, policy rejection or another documented safe outcome before any wrong-purpose cryptographic artefact is produced. • **Test preparation:** Select one prohibited wrong-purpose scenario from VE.Crypto7.01 and VE.Crypto7.02, such as wrong credential binding, wrong protocol function, wrong key usage, wrong algorithm/profile, wrong requester context, or attempted signing/decryption outside the permitted purpose. Define the expected safe outcome before testing. • **Testing steps:** The tester records the TOE version, platform, selected key-material class, intended purpose, prohibited purpose scenario, expected safe outcome and observation method in the test evidence. The tester attempts the prohibited operation through the vendor-supported negative test endpoint, malformed/wrong-purpose request, policy configuration or diagnostic scenario and observes the TOE UI, protocol result, artefact acceptance/rejection, endpoint response or approved diagnostic output. • **Test result:** The actual result is the observed wrong-purpose handling: the TOE rejects the operation before key use, requires a different permitted key/purpose, cancels safely, or produces a cryptographic artefact using the selected key outside its documented purpose. The result shall state

Element	Content
	whether any wrong-purpose signature, decryption, presentation, credential operation or protected request was produced, without storing sensitive payloads. • **Assessment (pass/fail):** PASS if the TOE prevents use of the selected key material outside its documented intended purpose and does not produce a wrong-purpose key-dependent artefact. FAIL if the TOE uses the key for a prohibited purpose, silently substitutes an undocumented purpose, or accepts a request that violates the documented key-purpose constraints.
Test Verdict	PASS iff TE.Crypto7.01, TE.Crypto7.02 and TE.Crypto7.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify key-material classes, intended purposes, permitted/prohibited operations, purpose constraints, selected positive/negative scenarios or observation methods.

Table — VE and TE of AS.Crypto8 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto8	**The TOE shall establish and manage cryptographic keys used within the Wallet Instance and its interfaces under a defined lifecycle covering generation, derivation or establishment, distribution where applicable, storage, access control, use restrictions, rotation or replacement, revocation, and destruction.**
Required vendor information	
VE.Crypto8.01	The vendor shall provide a cryptographic key lifecycle specification for the evaluated TOE configuration. The specification shall identify key classes used within the Wallet Instance and its interfaces, owner/component boundary, lifecycle states, generation/derivation/establishment method, distribution where applicable, storage category, access-control rule, use restriction, rotation/replacement rule, revocation/invalidation rule and destruction rule.
VE.Crypto8.02	The vendor shall provide SUG/test instructions, test accounts, test credentials or test key material, test endpoints and approved observation methods for selected key lifecycle events. Implementation evidence may include Crypto-Annex references, lifecycle state diagrams, keystore/API mapping, WSCA-interface mapping, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Crypto8.01	• **Test description with test expectation:** The vendor specification shall define the lifecycle of selected cryptographic keys used within the Wallet Instance and its interfaces. The expected result is a clear lifecycle mapping for each selected key class. • **Test preparation:** Prepare the Crypto-Annex, key lifecycle specification, evaluated TOE build and TOE/WSCA/WSCD/platform boundary description from VE.Crypto8.01. Prepare a checklist covering lifecycle states and management rules. • **Testing steps:** The tester records the TOE version, build identifier, selected key classes, lifecycle states, boundary assumption and evidence type. The tester reviews whether each selected key class is mapped to all applicable lifecycle rules. • **Test result:** The actual result is the lifecycle specification coverage: complete, partial, ambiguous, missing or inconsistent. The result shall identify missing lifecycle states or unclear boundary assignments without exposing key values. • **Assessment (pass/fail):** PASS if selected key classes have complete and evidenced lifecycle definitions attributable to the evaluated TOE or its defined interfaces. FAIL if lifecycle states, management rules, ownership boundary or evidence are missing, ambiguous or inconsistent.
TE.Crypto8.02	• **Test description with test expectation:** The TOE shall establish and use the selected key class according to its defined lifecycle and restrictions. The expected result is successful key-dependent operation only in a valid lifecycle state. • **Test preparation:** Select one key class and one key-dependent operation from VE.Crypto8.01. Prepare the approved observation method or implementation evidence from VE.Crypto8.02. • **Testing steps:** The tester records the selected key class, establishment method, lifecycle state, access-control condition, use restriction and evidence type. The tester triggers the normal establishment/use flow and observes the operation result or approved evidence.

Element	Content
	• **Test result:** The actual result is the valid-lifecycle behaviour: established and used as specified, blocked, ambiguous, inconsistent or not evidenced. The result shall describe lifecycle state and operation category without recording keys, seeds, tokens or reusable artefacts. • **Assessment (pass/fail):** PASS if the selected key is established and used only under the documented valid lifecycle state and restrictions. FAIL if the TOE uses an undocumented key state, bypasses access/use restrictions, or cannot evidence the lifecycle state.
TE.Crypto8.03	• **Test description with test expectation:** The TOE shall enforce the selected key lifecycle transition and prevent use of key material outside its resulting lifecycle state. The expected result is replacement, revocation, invalidation, destruction or another documented safe outcome. • **Test preparation:** Use one key class from TE.Crypto8.02 and select one vendor-supported lifecycle transition from VE.Crypto8.02, such as rotation, replacement, revocation, invalidation or destruction. Define the expected post-transition behaviour. • **Testing steps:** The tester records the selected transition, pre-transition state, expected post-transition state and evidence type. The tester executes the transition and attempts the selected key-dependent operation using the old, revoked, invalidated or destroyed state where applicable. • **Test result:** The actual result is the lifecycle-transition behaviour: replaced, revoked, invalidated, destroyed, safely denied, still accepted, or not evidenced. The result shall state whether any key-dependent artefact was produced after the key should no longer be valid. • **Assessment (pass/fail):** PASS if the TOE enforces the lifecycle transition and does not use key material outside the documented valid state. FAIL if old, revoked, invalidated or destroyed key material remains usable contrary to the lifecycle specification.
Test Verdict	PASS iff TE.Crypto8.01, TE.Crypto8.02 and TE.Crypto8.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify key classes, lifecycle states, owner/component boundary, lifecycle rules, observation methods, supported lifecycle-transition scenarios or implementation evidence for non-UI-observable controls.

Table — VE and TE of AS.Crypto9 (FITCEM PP: Cryptography)

Element	Content
Cryptography — AS.Crypto9	**The TOE shall validate public-key certificates, certificate chains, trust lists, trust anchors and certificate status information according to approved certificate policies and permitted key usage or purpose constraints before relying on a communication partner, issuer, relying party, WSCA/WSCD attestation or validation source.**
Required vendor information	
VE.Crypto9.01	The vendor shall provide a certificate and trust-validation specification for the evaluated TOE configuration. The specification shall identify validation contexts, certificate types, certificate-chain rules, trust lists, trust anchors, certificate-status sources, approved certificate policies, permitted key usage/purpose constraints, cache/update rules, ordering before reliance, failure behaviour, and which controls require documentation, configuration or source-code evidence.
VE.Crypto9.02	The vendor shall provide SUG/test instructions, test accounts, test endpoints, test certificates, test trust-list entries, test trust-anchor material and certificate-status responses for one valid and one invalid/non-conformant validation scenario. Implementation evidence may include trust-policy mapping, validation-flow diagrams, certificate-validation API mapping, configuration snippets, limited source-code excerpts or vendor test logs for the evaluated TOE build.
Required test procedures	
TE.Crypto9.01	• **Test description with test expectation:** The vendor specification shall define certificate/trust validation before reliance for selected validation contexts. The expected result is a clear mapping of validation inputs, policies, constraints and failure outcomes. • **Test preparation:** Prepare the Crypto-Annex, certificate/trust-validation specification, evaluated TOE build and boundary description from VE.Crypto9.01. Prepare a checklist covering certificate chain, trust source, status source, policy, key usage/purpose and reliance decision.

Element	Content
	• **Testing steps:** The tester records the TOE version, build identifier, selected validation contexts, specification version and evidence type. The tester reviews whether each selected context has a documented and evidenced validation rule before reliance. • **Test result:** The actual result is the validation-specification coverage: complete, partial, ambiguous, missing or inconsistent. The result shall identify missing validation categories without exposing private keys or production trust material. • **Assessment (pass/fail):** PASS if selected validation contexts have complete and evidenced rules for certificate, chain, trust, status, policy and key usage/purpose validation before reliance. FAIL if validation inputs, trust sources, constraints, failure behaviour or ordering before reliance are missing, ambiguous or not attributable to the evaluated TOE.
TE.Crypto9.02	• **Test description with test expectation:** The TOE shall rely on the selected valid entity or artefact only after the required certificate/trust validation succeeds. The expected result is successful reliance-dependent operation under the documented valid validation state. • **Test preparation:** Select one valid scenario from VE.Crypto9.02, such as valid issuer, relying party, communication partner, WSCA/WSCD attestation or validation source. Identify the expected certificate policy, status and permitted key usage/purpose. • **Testing steps:** The tester records the selected validation context, valid certificate/trust material, expected policy/status/key-usage condition, reliance-dependent operation and evidence type. The tester executes the selected flow and observes UI, protocol result, endpoint response, diagnostic output or implementation evidence. • **Test result:** The actual result is the valid-validation behaviour: accepted after validation, rejected, ambiguous, or inconsistent with evidence. The result shall state whether the reliance-dependent operation completed without recording secrets, private keys or production trust credentials. • **Assessment (pass/fail):** PASS if the TOE completes the reliance-dependent operation only under the documented valid validation state. FAIL if the valid state cannot be evidenced, if reliance occurs before validation, or if the observed decision contradicts the validation specification.
TE.Crypto9.03	• **Test description with test expectation:** The TOE shall reject or safely handle the selected invalid or non-conformant certificate/trust-validation state before reliance occurs. The expected result is safe refusal, trust error, operation cancellation, restricted operation or another documented safe outcome. • **Test preparation:** Select one invalid scenario from VE.Crypto9.02, such as revoked, expired, untrusted, wrong-policy, wrong-purpose/key-usage, broken-chain, stale-status or tampered-trust-list scenario. Define the expected safe failure outcome and affected reliance-dependent operation. • **Testing steps:** The tester records the selected validation context, invalid condition, expected safe outcome and evidence type. The tester attempts the selected TOE flow using the invalid certificate, trust-list/trust-anchor material, status response or validation source. • **Test result:** The actual result is the failed-validation behaviour: rejected, trust error, restricted operation, accepted, or not evidenced. The result shall state whether any reliance-dependent action occurred, without storing sensitive payloads or production trust material. • **Assessment (pass/fail):** PASS if the TOE prevents reliance when certificate/trust validation fails or required policy/status/key-usage/purpose constraints are not met. FAIL if the TOE relies on an invalid, untrusted, revoked, wrong-policy or wrong-purpose entity/artefact, or silently falls back to an undocumented trust decision.
Test Verdict	PASS iff TE.Crypto9.01, TE.Crypto9.02 and TE.Crypto9.03 all PASS. FAIL if any TE fails or if VE evidence is insufficient to identify validation contexts, certificate-chain rules, trust-list/trust-anchor sources, status-check rules, approved certificate policies, key usage/purpose constraints, positive/negative scenarios, observation methods or implementation evidence for non-UI-observable controls.