CredEntry

Powered by

APPENDIX H.9
TECHNICAL STANDARD
PLATFORM SKDs

Table of Contents

Technical Standard PS: Platform SDKs

Purpose

This document details how CredEntry will design, implement, and maintain its platform SDKs to meet compliance, interoperability, and security requirements. The SDKs are central to enabling wallet integration, ensuring standards alignment, supporting developers with tooling, and maintaining a secure and reliable environment for issuance and verification of digital credentials.

SDK Requirements and Implementation

PS-1: Wallet Integration SDK

- **Requirement**: A wallet integration SDK must be provided that supports all platform capabilities.
- **Implementation by CredEntry**:
    - A fully documented wallet SDK is provided with APIs supporting issuance, presentation, revocation, and credential lifecycle operations.
    - The SDK is demo-ready and will be included in developer onboarding packs.
    - API reference documentation and example projects will ensure fast integration.

PS-2: Cryptographic Binding to Secure Area

- **Requirement**: SDK must support cryptographic binding between a secure area and platform-managed credentials (ISO/IEC 18013-5).
- **Implementation by CredEntry**:
    - Device-specific secure elements (TEE, Secure Enclave, or Android Keystore) will be leveraged for credential binding.
    - The SDK integrates secure key pairs with ISO/IEC 18013-5-compliant flows.
    - All bindings are cryptographically verifiable and enforce non-repudiation.

PS-3: Separation of Concerns in Codebase

- **Requirement**: Clear separation of wallet core logic and integration layer to minimise coupling.
- **Implementation by CredEntry**:
    - Wallet SDK architecture follows clean architecture principles.
    - Core wallet functionality is encapsulated separately from the API and integration logic.
    - This approach simplifies updates, enables extensibility, and ensures compliance with the Digital ID Accreditation Rules 2024.

PS-4: Well-Documented APIs and Extension Points

- **Requirement**: OID4VC SDK must provide APIs and extension points for custom credential formats, DID methods, or cryptographic providers.
- **Implementation by CredEntry**:
    - Comprehensive API documentation will be delivered, with code samples and tutorials.
    - Extension hooks will allow developers to plug in custom DID methods or credential schemas.
    - Open-source compatibility will be reviewed to ensure community standards alignment.

PS-5: Automated Integration Tests

- **Requirement**: Developer tooling must include automated integration tests for end-to-end credential flows.
- **Implementation by CredEntry**:
    - SDK includes a test harness covering issuance, presentation, revocation, and selective disclosure.
    - Automated tests simulate multiple credential/document types.
    - Integration with CI/CD ensures ongoing quality assurance.

PS-6: Automated Security Scanning

- **Requirement**: SDK tooling should incorporate SAST/DAST and dependency vulnerability scanning.
- **Implementation by CredEntry**:
    - GitHub Advanced Security and dependency scanning are integrated into SDK builds.
    - Static (SAST) and dynamic (DAST) analysis are run automatically as part of QA.
    - Security alerts are triaged, tracked, and patched with defined SLAs.

PS-7: Release Management and Updates

- **Requirement**: SDK must have a defined process for releasing updates based on evolving standards.
- **Implementation by CredEntry**:
    - A release management process is documented and integrated into product governance.
    - New standards (ISO/eIDAS updates) are assessed, impact-analysed, and prioritised.
    - Releases include changelogs, migration guides, and backwards-compatibility considerations.

PS-8: Selective Disclosure and User Transparency

- **Requirement**: SDK must allow developers to define attributes to disclose and display verification context.
- **Implementation by CredEntry**:
    - User interface components are included in the SDK to display verifier identity, requested attributes, and intended use.
    - Users can explicitly approve or deny attribute disclosures.
    - Built in accordance with ISO/IEC 18013-5 and ISO/IEC 29100 privacy principles.

PS-9: Inter-Jurisdictional Use Cases

- **Requirement**: SDK should support verification across jurisdictions using ISO/IEC 18013 series standards.
- **Implementation by CredEntry**:
    - SDK designed to verify credentials issued by other jurisdictions (state-to-state interoperability).
    - Conformance testing with ISO/IEC 18013-5 and ISO/IEC 18013-7 ensures compatibility.
    - This ensures WA-issued credentials can be validated nationwide.

PS-10: Background Activities and Push Notifications

- **Requirement**: SDK should support background tasks and push notifications.
- **Implementation by CredEntry**:
    - SDK integrates with native push notification frameworks (APNS, FCM).
    - Background sync is used for refreshing credential status, revocations, or updates.
    - Users are notified securely without exposing sensitive data.

PS-11: SLA Framework for SDKs

- **Requirement**: Supplier must provide SLA framework aligned with ACSC Secure by Design foundations.
- **Implementation by CredEntry**:
    - Draft SLA covers:
        - Release cadence and version support timelines.
        - Vulnerability remediation timeframes.
        - Commitments to transparent disclosure of SDK issues.
    - SLA aligns with ACSC Secure by Design principles, ensuring resilience and developer trust.

Continuous Improvement and Governance

CredEntry commits to maintaining SDK quality and compliance through:

- Ongoing monitoring of emerging standards and interoperability requirements.
- Regular developer community feedback cycles.
- Continuous integration testing and vulnerability scanning.
- Transparent release notes and documentation updates.