WE BUILD CONSORTIUM

Deliverable D4.1

Architecture & Integration Blueprint Reference Document

Version: 1.0
March 2026

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Project and document data

Item	Description
Project title	**W**allet **E**cosystem for **B**usiness and payments **U**se cases on **I**dentification, **L**egal representation and **D**ata sharing
Grant Agreement no	Project 101224751 — WE BUILD
Deliverable title	D4.1 Architecture & Integration Blueprint Reference Document
Deliverable type	R — Document, report
Responsible party	Swedish Agency for Digital Government (DIGG), 900474378
Authors	Sarah Amandusson, Digg, SE Sander Dijkhuis, Cleverbase, NL George Fourtounis, GRNet, GR Benjamin Hansson, iGrant.io, SE Leif Johansson, SIROS Foundation, SE Svilena Rakshieva, Evrotrust Technologies, BG Sebastian Elfors, IDNow, FR George J Padayatti, iGrant.io, SE Giuseppe De Marco, Dipartimento per la trasformazione digitale, IT Ronald Koenig, Spherity, DE
Contributing parties	Steffen Piel, Governikus, DE Malin Norlander, Bolagsverket, SE Lal Chandran, iGrant.io, SE Aleksandar Simsic, ICTU, NL Esther Maakay, Signicat, NL Hristian Daskalov, Evrotrust Technologies, BG Thodoris Papadopoulos, GRNet, GR Eelco Klaver, Credenco, NL Viky Manaila, Intesi Group, IT Andreas Abraham, Validated ID, ES Alejandro Nieto, Digitel TS, ES Stefan Hadjistoytchev, Evrotrust Technologies, BG Andrew Freund, D-Trust, DE Miguel Aguilar, Bolagsverket, SE Dr. Oliver Froitzheim, Bundesanzeiger Verlag GmbH, DE Sarah Gräfer, Bundesanzeiger Verlag GmbH, DE
Reviewers	Andriana Prentza, University of Piraeus, GR Ard van der Heijden, ICTU, NL Marie Austenaa, Data Craft, NO Stef Haartman, NL Xavier Juredieu, FR
Dissemination level	PU - Public

History of changes

Version	Date	Change
1.0	2026-03-20	First complete version with full content

Table of contents

List of figures

List of tables

1. Introduction and Context

1.1 Background

WE BUILD is a Large Scale Pilot (LSP) funded by the European Commission. The project tests how the European Digital Identity (EUDI) Wallet and the European Business Wallet (EBW) can support cross-border business processes across the EU.

The goal is practical: reduce administrative barriers that slow down companies when they operate across borders, such as opening a bank account, exchanging trusted business documents, or registering a branch in another country.

WE BUILD is organised around 13 concrete use cases that demonstrate how digital identity wallets can support real business processes. These use cases are grouped into three main areas:

- Business (WP2) – processes such as company registration, mandates and representation.
- Supply Chain (WP2) – logistics, transport documentation and electronic invoicing.
- Payments & Banking (WP3) – secure payments and simplified onboarding to financial services.

1.2 WE BUILD's Role in the EUDI and EBW Journey

WE BUILD operates within the emerging EUDI and EBW ecosystem, but it is not the final production environment. Instead, the project acts as a large-scale pilot where technical solutions, governance models and interoperability rules can be tested through real use cases.

In the final EUDI ecosystem, every EU citizen will receive an EUDI Wallet at Level of Assurance (LoA) High. WE BUILD focuses in particular on the EBW, designed for economic operators to manage mandates, exchange trusted business documents such as electronic invoices, and receive legally valid notifications. Some EBW functions, such as onboarding and data portability, will operate at LoA Substantial.

1.2.1 Bridging ARF Gaps through Specifications and Testing

The future EUDI ecosystem is defined through the Architecture Reference Framework (ARF). Because the ARF is still evolving, it does not yet cover every implementation detail. In the WE BUILD pilot environment, the full certification and qualification schemes used in production cannot always be applied.

To address these gaps, WE BUILD defines project-specific implementation rules through:

- **WE BUILD Conformance Specifications (WBCS)** – technical rules that implementations must follow.
- **Architectural Decision Records (ADRs)** – documented architecture decisions that guide the project.

In the final ecosystem, wallets and services must undergo formal certification by national supervisory bodies.

WE BUILD does not	WE BUILD does
certify EUDI wallets	provide WE BUILD wallets that pass ITB testing
rely on eIDAS-eID	provide WE BUILD eID with fictitious but realistic identities
create eIDAS-qualified e-signatures	define WE BUILD qualification of e-signatures focused on technical interoperability
use real MS registries	use real public sector bodies where possible, otherwise simulate them using fictitious
use the EC List of Trusted Lists (LoTL)	provide a WE BUILD LoTL
use the MS Trusted lists (TL)	provide WE BUILD TLs with input from supervisory bodies where available
reach production-level legal liability	operate within the WE BUILD agreement and trust framework, not within eIDAS
deal with national policy-making	use the WP5 MS Forum to indicate alignment with national policy-making
deal with universal definitions	define WE BUILD semantics within the available timeframe
issue EUDIW-RP access certificates	issue WE BUILD RP access certificates
issue eIDAS-QEAA	define WE BUILD QEAA focused on technical interoperability

Table 1: WE BUILD Scope and Limitations

1.3 Work Package 4 (WP4) - General Capabilities

The technical groups in WP4 - Architecture, Semantics, Wallet Providers, PID & EBWOID Providers, Qualified Trust Service Provider (QTSP), Trust Registry Infrastructure, and Test Infrastructure - provide the technical capabilities that support the use cases.

WP2 and WP3 use cases are expected to use the capabilities provided by WP4 rather than developing parallel technical solutions.

To ensure interoperability across participants, WE BUILD uses three levels of documentation:

1. This **Architecture & Integration Blueprint (D4.1)** - the high-level architecture and system overview.

2. **Architectural Decision Records (ADR)** [1]- explains major architecture decisions and the reasoning behind them.
3. **WE BUILD Conformance Specifications (WBCS)**[2] – defines the detailed technical requirements that implementations must follow.

The governance process behind ADRs and WBCS, including how decisions are proposed and adopted, is described in Chapter 8.

The Interoperability Testbed (ITB) is a first step toward understanding conformity assessment requirements. In a controlled consortium environment, regulatory and technical specifications are translated into executable interoperability scenarios.

1.4 Scope and goal of the Blueprint

This document defines the WE BUILD Architecture & Integration Blueprint, providing architectural guidelines and integration requirements for interoperable implementation.

It outlines the system architecture, key components, and their interactions, and serves as a reference for aligning implementations across participants.

1.5 Target audience

This document is intended for technical and architectural stakeholders including:

- Technical teams implementing wallet, issuer, and relying party components
- Architects responsible for system design and integration
- Trust service providers and infrastructure providers
- Public and private sector participants in WE BUILD use cases

For detailed technical requirements, see the WE BUILD Conformance Specifications (WBCS) and Architectural Decision Records (ADRs).

1.6 Versioning

This document represents a formal deliverable with a defined submission date under the Grant Agreement. It reflects the state of the WE BUILD Architecture & Integration Blueprint at the time of submission.

The architecture evolves throughout the project as new insights are gained. The online version of the blueprint is continuously updated and should be considered the authoritative and most up-to-date source. [3]

---

[1] https://github.com/webuild-consortium/wp4-architecture/tree/main/adr

[2] https://github.com/webuild-consortium/wp4-architecture/tree/main/conformance-specs

[3] https://webuild-consortium.github.io/wp4-architecture/blueprint/blueprint.html
https://webuild-consortium.github.io/wp4-architecture/blueprint/blueprint.pdf

This Word document may be updated at specific milestones (e.g. project reviews), but does not reflect ongoing changes between these updates.

2. Regulatory and Foundational Alignment

The WE BUILD architecture aligns with two key regulatory instruments: Regulation (EU) No 910/2014, as amended by Regulation (EU) 2024/1183 (commonly referred to as the amended eIDAS Regulation), and the proposed European Business Wallet proposal for economic operators.

2.1 The EUDI Framework

WE BUILD aligns with the legal and technical framework for EUDI wallets. Users can authenticate and present identity and attribute information while retaining control over what data is shared through selective disclosure and explicit consent.

The amended eIDAS Regulation is supported by several implementing acts defining the technical and governance framework for the EUDI Wallet ecosystem, most importantly:

Core Wallet Architecture and Technical framework

- 2024/2979 – Integrity and core functionalities
- 2024/2982 – Protocols and interfaces
- 2024/2981 – Certification framework
- 2024/2980 – Notification obligations within the Wallet ecosystem

Person Identification Data (PID) and Electronic Attestations of Attributes (EAA)

- 2024/2977 – PID and electronic attestations of attributes
- 2025/1569 – Electronic attestations of attributes

Wallet Ecosystem Governance and Relying Parties

- 2025/848 – Registration of wallet-relying parties
- 2025/849 – Submission of information on certified European Digital Identity Wallets
- 2025/847 – Reactions to Wallet security breaches
- 2025/846 – Cross-border identity matching for natural persons

Trust Services and QSCD Framework

- 2025/1566 – Verification of identity and attributes (QTSs)
- 2025/1567 – rQSCD management
- 2025/1570 – Notification of certified or cancelled QSCDs
- 2025/1572 – QTSP initiation, notification and verification

2.1.1 Standardisation and Technical Specifications

The European Commission, together with the European Digital Identity Cooperation Group, has published:

- The ARF specifies main functionalities, roles and responsibilities, architecture and design principles, attestation formats and protocols, trust model, certification, and risk management of the EUDI Wallet ecosystem.
- The Technical Specifications specify more technical details of selected topics derived from the ARF. The technical specifications describe various topics such as Relying Party registration, zero-knowledge proofs, attestation rulebooks, schemas and catalogues.

Furthermore, there are several standardisation organisations that contribute with standards for the EUDIW ecosystem.

- **ETSI ESI** is a European Standardisation Organisation (ESO) that creates technical standards and European Norms for electronic identity and signatures supporting the eIDAS regulation. ETSI ESI has published approximately 80 standards for QTSP conformity assessment, protocols and formats for digital signatures, as well as protocols and formats for the EUDI Wallet. ETSI ESI has received the standardisation request STF 705 from the EU Commission to create and/or update several standards for the EUDI Wallet ecosystem.
- **CEN Technical Committee 224 (TC224)** is an ESO that has published several standards related to identification and devices with secure elements. More specifically, CEN TC224 WG17 are standardizing Common Criteria protection profiles of QSCD/WSCA, CEN TC224 WG18 develop standards related to biometric solutions, whilst CEN TC224 WG20 are creating standards related to EUDI Wallet onboarding and access control.
- **ISO/IEC:** ISO is an international standardisation organisation and the International Electrotechnical Commission (IEC) develops international standards for electronic technologies. The international standardisation activities related to digital identities are performed within ISO/IEC Joint Technical Committee (JTC) 1 "Information Security". More specifically, several ISO/IEC standards are applicable to Common Criteria certification, conformity assessment and evaluation of the EUDI Wallet solutions. Furthermore, ISO/IEC has standardised the mobile driving license (ISO mDL) in ISO 18013-5, which is a PID format for the EUDI Wallet.
- **The Internet Engineering Task Force (IETF)** creates technical standards that comprise the internet protocol suites. More specifically, IETF PKIX covers secure data exchanges and formats in the area of electronic signatures, PKI and trust services. Most notably, IETF has published standards for PKIX X.509 certificate and CRL profiles, OCSP, TLS and SD-JWT, which are relevant for the EUDI Wallet ecosystem. Furthermore, some of the IETF standards are used as basis by ETSI ESI, which have created European profiles of Qualified Certificates, AdES signature formats, SD-JWT VC, etc.

- **OpenID Foundation:** The OpenID Foundation is an industrial standardisation organisation that develops open standards for identity, federation and security. The following OpenID standards are relevant for the EUDIW technical architecture: OpenID Connect Core (OIDC), OpenID For Verifiable Credential Issuance (OID4VCI), OpenID For Verifiable Presentations (OID4VP), and OpenID High Assurance Interoperability Profile (HAIP). OID4VP, OID4VCI and HAIP are used as the foundation for the ETSI TS 119 472 standardisation of EUDI Wallet protocols.
- **W3C:** The World Wide Web Consortium (W3C) is an international standardisation organisation. The following W3C standards are relevant to the EUDI Wallet technical architecture: W3C Verifiable Credentials Data Model, W3C Web Authentication (WebAuthn), and W3C Digital Credentials API. More specifically, the W3C Verifiable Credentials Data Model is referenced as basis for an ETSI TS 119 472 EAA profile.
- **Cloud Signature Consortium (CSC):** The Cloud Signature Consortium (CSC) is an international standardisation organisation focusing on compliant digital signature creation in the cloud. The CSC specification "CSC API v2 - Architectures and protocols for remote signature applications" is referenced by the EUDI Wallet architecture and is used as basis for the ETSI TS 119 432 standard.

In addition to the aforementioned standardisation organisations, the European Cybersecurity Agency (ENISA) is developing the EUDI Wallet Certification Scheme, which will be published as an implementing regulation under the Cybersecurity Act. The purpose of the EUDI Wallet Certification Scheme is to harmonise the national certifications of the EU Member States' EUDI Wallets.

2.2 The EBW Framework

The EBW framework is introduced through the European Commission's Digital Package proposal as part of its 2025 Work Programme. The proposal aims to establish the EBW as a harmonised digital solution that reduces administrative burden and allows companies and public authorities to identify, authenticate and exchange data with legal effect across the European Union.

The EBW framework complements the EUDI framework by addressing the needs of economic operators and public authorities. It supports the digital management of representation rights and mandates, provides a secure channel for exchanging official documents and attestations, and includes support for a common directory. Interoperability with the EUDI Wallet is a core requirement.

The proposal supports the management and use of EAA, including owner identification data with selective disclosure. It defines requirements for authenticating owners and authorised users through (Q)EAAs and enables links between EAAs and other attestations. Access to EAAs by relying parties requires proper authorisation.

The framework relies on existing eIDAS trust services such as qualified electronic signatures, seals, timestamps and registered delivery services.

The proposal also introduces a European Digital Directory maintained by the Commission. The directory functions as a trusted internal system where EBW providers notify relevant service information and where digital addressing can be supported. Detailed requirements will be defined in future implementing acts.

The regulation supports role-based access so that multiple authorised users can operate a wallet. It also enables secure data exchange between EBWs, EUDI Wallets and relying parties, while allowing additional functionalities provided that core features remain unaffected.

From a technical perspective, the framework promotes the use of common protocols for sharing attestations. It requires secure onboarding using eID with a LoA of at least Substantial and mandates interoperability, secure communication interfaces, and mechanisms for validation and revocation. Further requirements will be defined in implementing acts.

Within WE BUILD, the proposed EBW framework is treated as a primary regulatory and architectural reference for business-focused identity and data exchange scenarios. Use case design and pilot activities align with the EBW model's legal structure, interoperability requirements and trust-service framework, while taking forthcoming implementing acts into account.

3. Architecture Overview

3.1 Architectural Principles

While the previous chapter describes the regulatory and architectural frameworks that WE BUILD aligns with, this chapter introduces the architectural principles guiding the design of the WE BUILD ecosystem.

- **Interoperability:** Wallet providers, issuers and verifiers interact across organisational and national boundaries.
- **Reusability:** The architecture builds on existing EU digital infrastructure and results from previous Large Scale Pilots.
- **Security by design:** Security controls are integrated into the architecture from the start.
- **Privacy by design:** Users retain control over personal and organisational data through selective disclosure and explicit consent.

3.2 The Ecosystem at a Glance

The EUDI Wallet and EBW ecosystem follows the common three-party attestation model. In this model, three primary actors interact: issuer, holder and verifier. A trust framework supports these actors by providing the trust anchors used for validation.

1. **Holder** – the wallet controlled by a natural or legal person.
2. **Issuer** – an entity that issues attestations to the Holder.
3. **Verifier** – a relying party that receives and validates attestations presented by the Holder.
4. **Trust framework** – the infrastructure used to validate trust relationships between ecosystem participants (described in Chapter 7).

3.3 System Landscape

The diagram below illustrates the baseline trust topology of the EU wallet ecosystem. Issuers provide attestations to holders, holders present them to verifiers, and all actors validate trust relationships using the trusted lists.

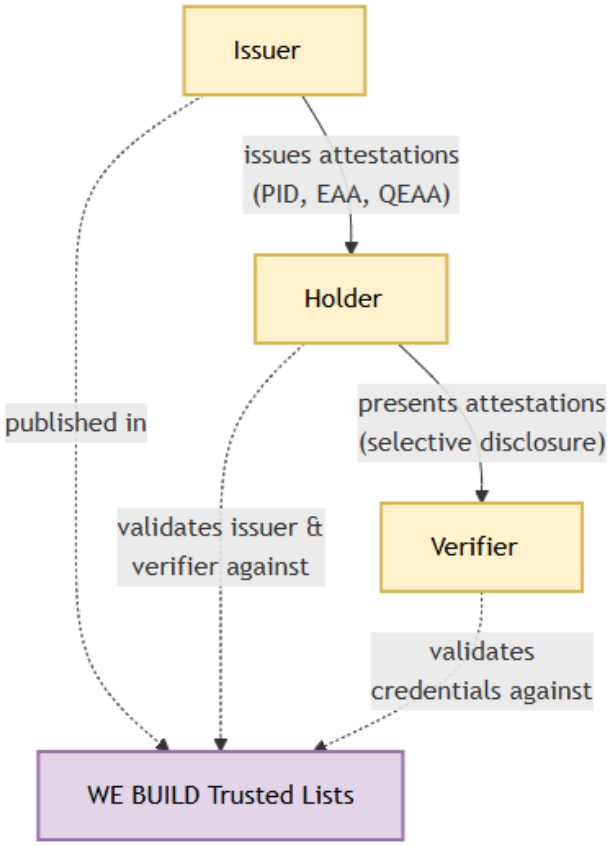

Figure 1: Baseline trust topology of the EUDI Wallet ecosystem.

WE BUILD focuses primarily on wallets for economic operators and public sector bodies. In these scenarios, qualified electronic registered delivery services (QERDS) support trusted messaging between recognised participants. Accordingly, interactions between data senders and receipients may be routed through a Qualified Trust Service Provider (QTSP) providing a QERDS. The QTSP is recognised in a scheme for trust services, just like in the previous diagram, enabling other participants to verify QERDS evidence issued by the QTSP. The WE BUILD Digital Directory (simulating the European Digital Directory) provides economic and public sector bodies with digital addressing for secure routing of documents and notifications.

While this model can apply to any data transmission, the senders, recipients and their QERDS providers can take the issuer-holder-verifier roles as illustrated as above. The QERDS provides an additional layer in the WE BUILD ecosystem:

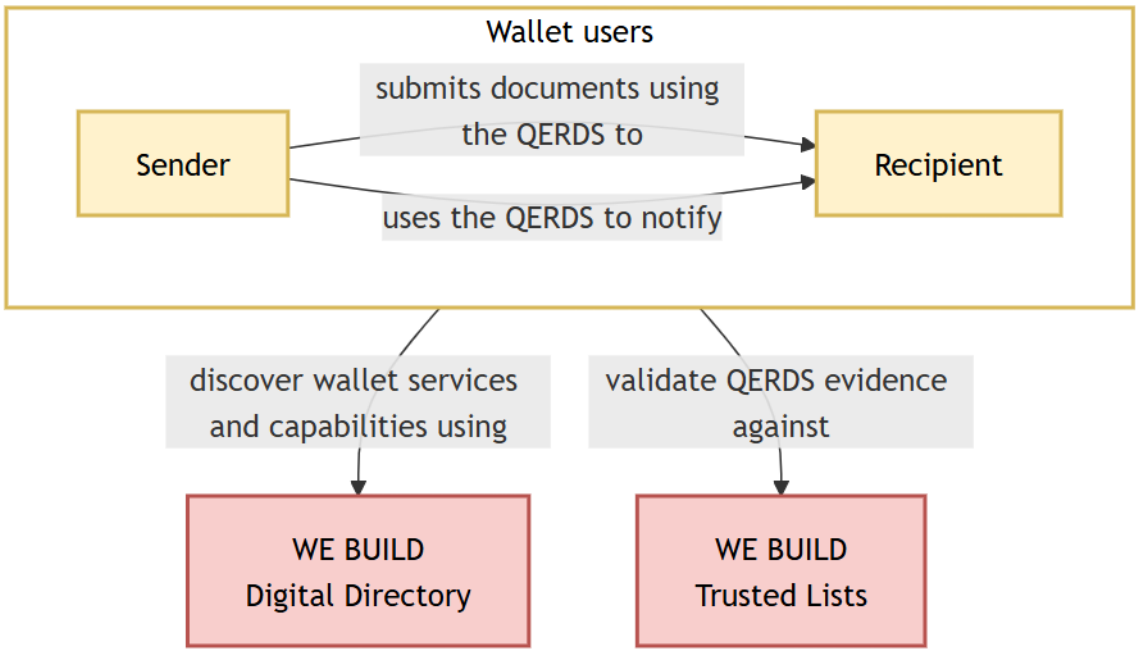

Figure 2: Baseline trust topology of the WE BUILD EBW ecosystem

3.4 Wallet Types in WE BUILD

WE BUILD supports wallet solutions for both natural persons and economic operators.

Natural persons interact through EUDI Wallets, which enable individuals to authenticate and present personal identity attributes. Economic operators interact through EBW, which enable organisations to manage and present business-related attestations such as representation rights or organisational attributes.

From a deployment perspective, wallet solutions can be implemented in several ways depending on the target users, operational requirements, and cryptographic architecture. In practice, three main implementation approaches are relevant within the WE BUILD ecosystem.

Wallet type	**Typical context**	**Characteristics**
Mobile wallets (on-device)	Natural persons	Wallet application running on a user's smartphone, with credentials stored and used locally on the device.
Server or Web-based wallets	Economic operators	Wallet services operated in backend infrastructure and accessed through Web interfaces or enterprise systems.
Hybrid wallets	Both contexts	Combine device-based interaction with backend cryptographic infrastructure.

Table 2: WE BUILD Wallet types

The underlying cryptographic architecture of wallets is defined in the ARF and related standards. This Blueprint therefore focuses on the interactions and interoperability

patterns relevant for WE BUILD rather than repeating the detailed wallet architecture definitions.

In practice, most deployments follow a mobile-first approach for natural persons and a server-based or enterprise-integrated approach for economic operators. Hybrid architectures may also be used to combine device-based user interaction with backend cryptographic services.

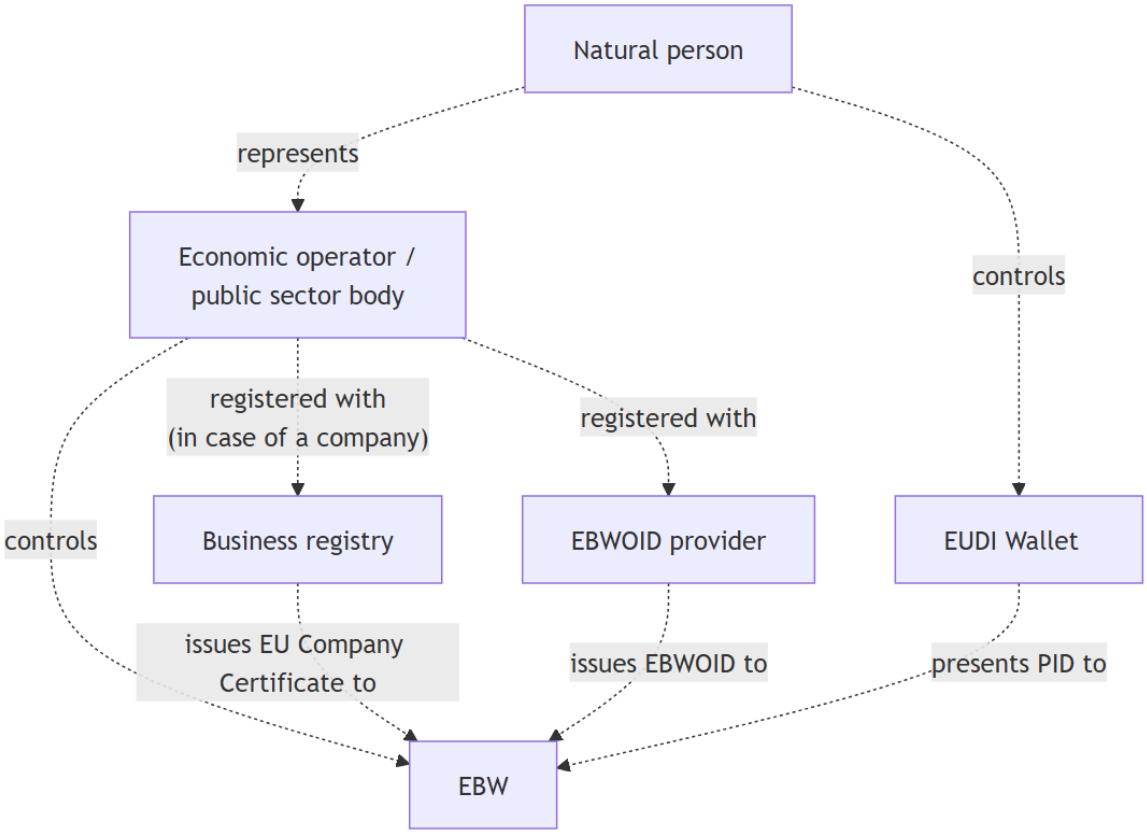

Figure 3: Concept model of relation between the EUDI Wallet and EBW

4. How the Wallet Interacts with Services

Chapter 4 introduced the main actors in the WE BUILD ecosystem. This chapter describes how these actors interact through wallet-based service flows.

4.1 Interaction Pattern: Attestation Issuance

The WBCS for high-assurance credential issuance[4] defines the requirements used in the project to ensure interoperable issuance of verifiable digital credentials between wallets and issuers. For reference on qualified electronic attestation of attributes, see the QEAA documentation.[5]

The WE BUILD ecosystem mainly supports two credential issuance models, which differ in which actor initiates the process: wallet-initiated issuance and issuer-initiated issuance. If the credential cannot be issued immediately, deferred issuance is used. The wallet retries periodically until the credential is issued or an unrecoverable error occurs.

4.1.1 Wallet-initiated Issuance

This issuance flow is initiated by the user:

1. The user opens their wallet and selects the credential type to be issued (for example, a PID or a QEAA).
2. The wallet connects with the corresponding issuer and requests the credential.
3. The user authenticates with the issuer, following the procedure specified by the issuer itself.
4. The issuer requests the user's consent to issue the credential and send it to their wallet.
5. The issuer generates the credential and delivers it to the wallet.
6. The wallet verifies the authenticity of the credential and stores it. From this point, the user becomes responsible for managing the issued credential.

[4] https://github.com/webuild-consortium/wp4-architecture/blob/blueprint/updates-jan/conformance-specs/cs-01-credential-issuance.md

[5] https://github.com/webuild-consortium/wp4-architecture/blob/main/blueprint/04-integration-model.md#qeaa-documentation

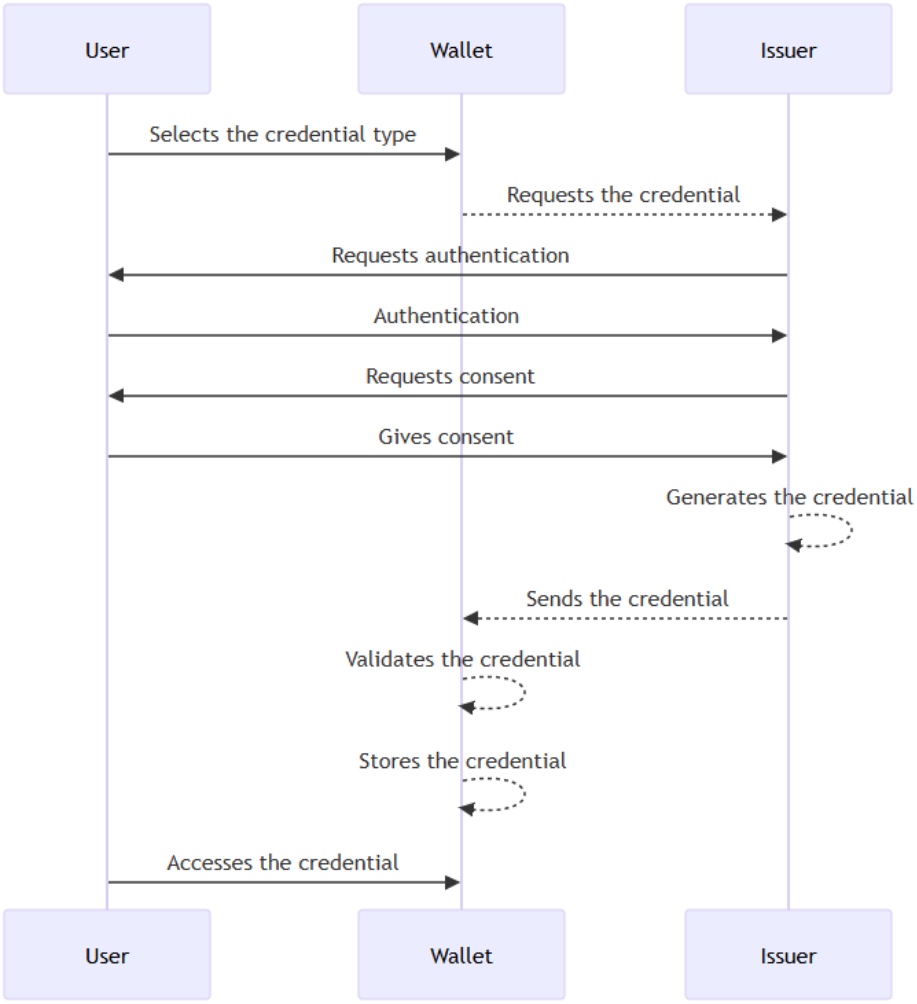

Figure 4: Wallet-initiated Issuance

4.1.2 Issuer-initiated Issuance

This issuance flow is initiated by the issuer:

1. The user interacts with the issuer (for example, during a digital onboarding process).
2. The issuer prepares one or more credentials.
3. The issuer offers these credentials to the user. This can be done in several ways, both same-device and cross-device:

    - By displaying a QR code that the user shall scan with their wallet.
    - By sending a link to the wallet.

4. The wallet displays the offer and requests confirmation from the user.
5. The user authenticates with the issuer, following the procedure specified by the issuer itself.
6. The issuer requests the user's consent to issue the credential and send it to their wallet.

7. The issuer generates the credential and delivers it to the wallet.
8. The wallet verifies the authenticity of the credential and stores it. From this point, the user becomes responsible for managing the issued credential.

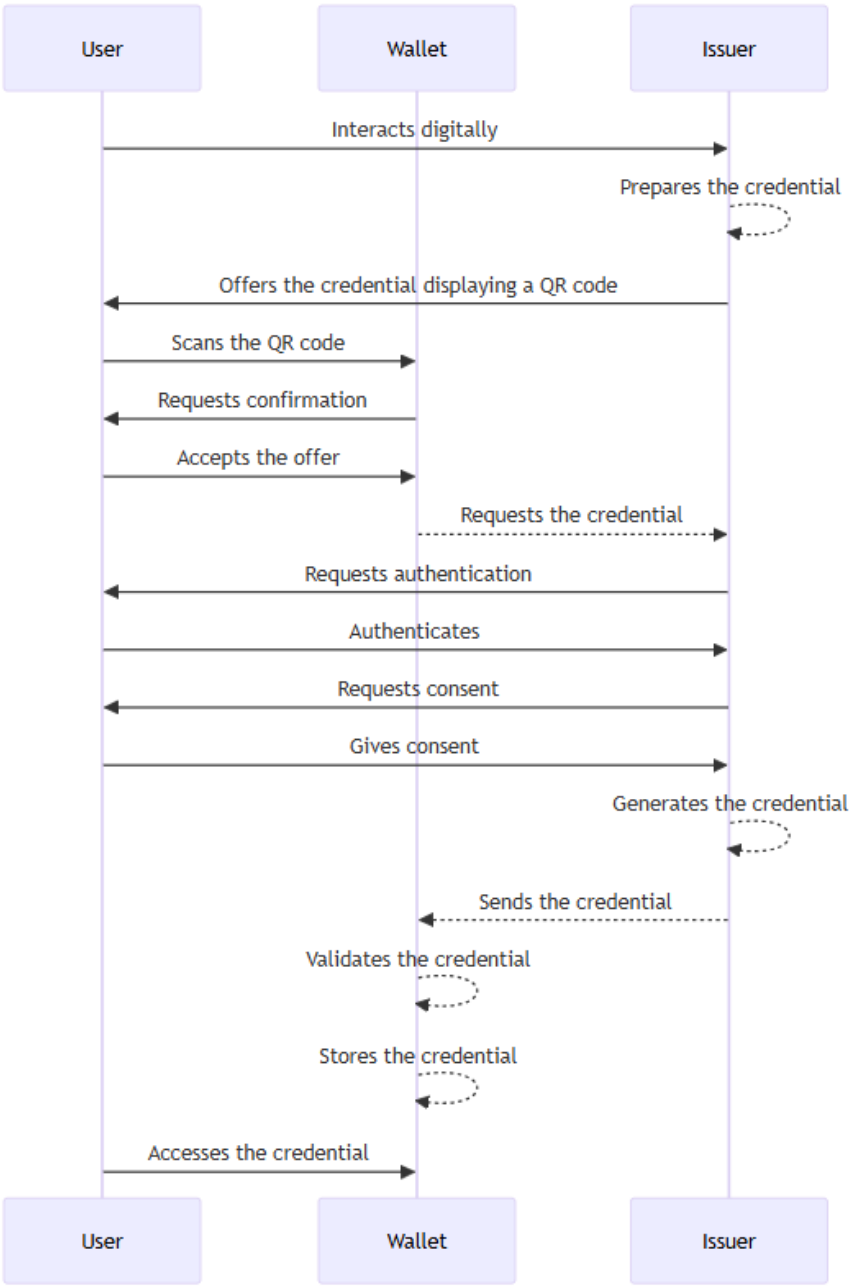

Figure 5: Issuer-initiated Issuance

4.2 Interaction Pattern: Attestation Presentation (Receiving)

In this pattern, a verifier requests specific attestations from the wallet. The wallet presents the requested information, typically using selective disclosure mechanisms, and the verifier validates the received data.

The WE BUILD Conformance Specification for Credential Presentation[6] describes how wallets and relying parties interoperate within the WE BUILD ecosystem. It covers presentation (request and response flows), interfaces between wallets and relying parties as well as security, privacy and interoperability requirements and same-device and cross-device invocation patterns.

4.3 Signature and Seal Integration

Wallets in WE BUILD provide the ability to create qualified electronic signatures and seals. This section describes the various integration models. For reference, see the QES documentation.[7]

WE BUILD supports both wallet-centric and QTSP-operated approaches for electronic signatures and seals. In both models, the wallet provides the user interaction layer, while cryptographic operations may take place either locally or in remote infrastructure operated by a QTSP.

Both approaches are compatible with the architectural patterns described in the ARF. However, during the WE BUILD pilot phase not every wallet provider or QTSP is expected to implement every possible model. For interoperability across the consortium, WE BUILD therefore treats remote signing and sealing through QTSP-managed services with standardised interfaces as the common baseline. Local signing models may still be supported by individual wallet implementations, but they are not assumed as a uniform baseline for interoperability within the project.

This section aligns with the WP4 interoperability baselines defined for issuance and presentation flows. Proximity-based signing scenarios are currently outside the baseline protocol scope of the WE BUILD pilots.

In the WE BUILD pilot and ITB environment, eIDAS-qualified status cannot be achieved because the ITB operates outside the formal eIDAS certification framework. Any reference to “qualified” in WE BUILD therefore represents a technical demonstration only and does not constitute a legally valid qualified electronic signature. The prerequisites for eIDAS-qualified status remain unchanged, including use of a Qualified Signature Creation Device (QSCD) and a qualified certificate issued by a QTSP that is listed on an official national Trusted List.

4.3.1 Wallet-centric Signing Model

In the wallet-centric model, the EUDI Wallet is the central component of the electronic signature process. Three distinct signing processes are considered, depending on where the Signature Creation Application (SCA) runs and where the Signature Creation Device (SCD) is hosted.

[6] https://github.com/webuild-consortium/wp4-architecture/blob/blueprint/updates-jan/conformance-specs/cs-02-credential-presentation.md

[7] https://github.com/webuild-consortium/wp4-architecture/blob/main/blueprint/04-integration-model.md#qes-documentation

1) Remote Signing with External SCA

The user initiates signing from the wallet, while the SCA is external. The document is sent to the external SCA for review and consent, after which the signing request is forwarded to a remote SCD that creates the signature and returns the result.

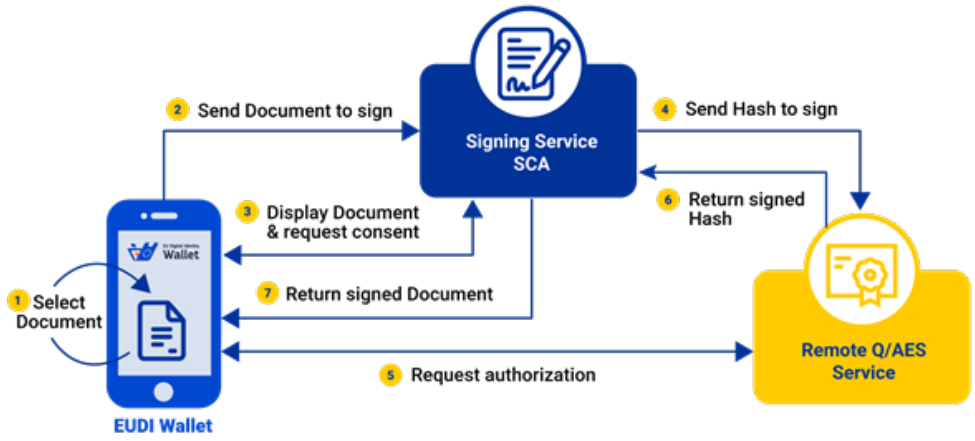

Figure 6: Remote Signing with External SCA

2) Remote Signing with Local SCA (Wallet as SCA)

The user initiates signing from the wallet, which also acts as the SCA. The document is presented to the user within the wallet for review and consent. After approval, the wallet forwards the signing request to a remote SCD that produces the signature and returns the result.

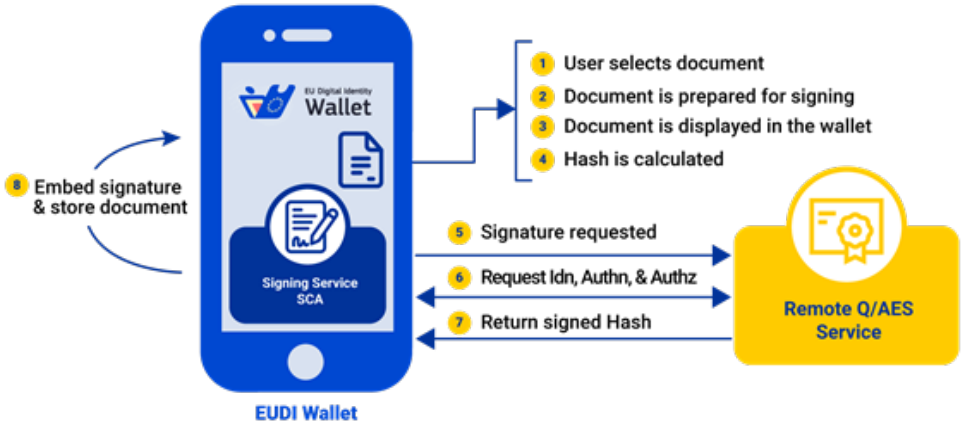

Figure 7: Remote Signing with Local SCA (Wallet as SCA)

3) Local Signing

The user initiates signing from the wallet, which also acts as the SCA. The document is presented to the user within the wallet for review and consent. After approval, the signature is created locally using a SCD integrated in the user's device.

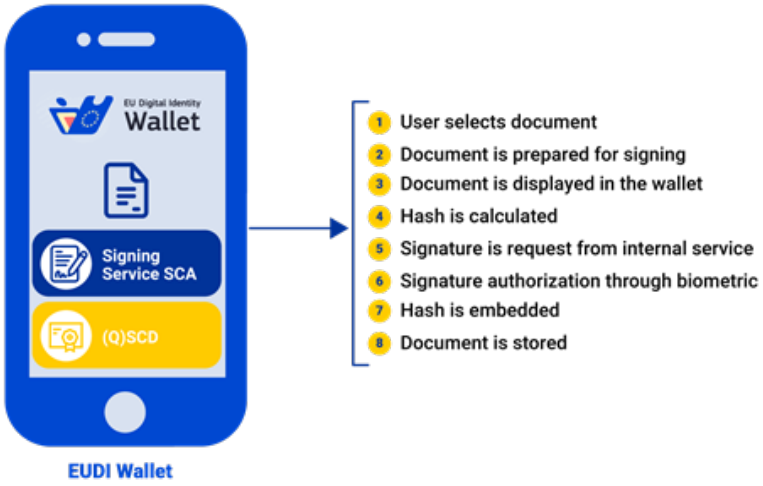

Figure 8: Local Signing

4.3.2 QTSP-centric Signing and Sealing Model

In the QTSP-centric model, the trust service provider operates the signing or sealing process. The cryptographic key material used for signature or seal creation is generated, stored, and used within infrastructure controlled by or on behalf of the QTSP, typically in secure hardware environments. From the perspective of the user, signing and sealing are therefore remote operations. External components interact with the trust service through defined interfaces, while the cryptographic operation itself is performed within the QTSP-controlled environment.

Within this architecture, the EUDI Wallet may act as a client-side orchestration component. It can authenticate the user, capture user intent, and trigger a signing operation. However, it does not operate the signature creation environment, does not manage signing credentials, and does not assume the responsibilities of a trust service provider.

In this model, the QTSP remains responsible for identity binding, credential issuance, and compliance with applicable ETSI standards. Signature or seal creation data remains under controlled conditions consistent with the required assurance level, and activation mechanisms enforce the conditions required for advanced or qualified signatures, including sole control where applicable.

The pilot implementation aims to remain technically aligned with qualified signing requirements. Pilot trust validation is described below and relies on consortium trusted lists.

4.3.3 CSC Interoperability Profile for Remote Signing and Sealing

For remote signing and sealing flows, WE BUILD uses the Cloud Signature Consortium (CSC) interoperability framework. CSC APIs expose standardised interfaces that allow wallets and client applications to interact with QTSP-operated signing services. The detailed WE BUILD CSC interoperability profile will be defined in a WBCS. That specification will describe the concrete integration details, including authorisation mechanisms, endpoints, supported formats and algorithms, and interoperability constraints used in the ITB. Until such a profile is published, CSC API v2.2.0.0 serves as the base reference specification for CSC-based interactions.

During the pilot phase, trust validation relies on consortium reference trust mechanisms. Wallet components or external SCAs validate participating QTSPs and trust anchors using WE BUILD trusted lists. The reference trusted list may include participating QTSP entries and their registered issuing certificate authorities for pilot purposes.

When a signing request is processed, the SCA validates the signer's certificate chain against issuing certificate authorities listed in the WE BUILD trusted list and checks the QTSP status within the pilot trust framework. Revocation status is validated using OCSP responders or CRL distribution points operated by participating QTSPs. Where registration status checks are required, registrar processes are simulated through mock registrar services and endpoints.

4.3.4 Organisational Signing: Individuals Signing on Behalf of a Company

WE BUILD supports signing scenarios where an individual signs on behalf of an organisation. This model reuses the wallet-centric and QTSP-operated signing approaches described above. The wallet provides the user interface for document review and approval, while the QTSP performs the signature or seal creation within its controlled environment.

In these scenarios, the transaction must bind both the natural person and the organisation represented. The natural person identity is represented by the PID, while the organisation context is represented through the EBWOID, which acts as the cross-border minimum organisation identifier.

At signing time, identifiers or references to both the PID and the EBWOID are included in the transaction data presented to the user and subsequently authorised or signed. This ensures that the resulting signature or seal can be unambiguously linked to both the individual and the organisation.

The exact representation of these bindings is use-case specific and will be defined in rulebooks and WBCS (see Chapter 6 for the semantic and schema model).

4.4 Secure Communication Channel

This section describes how secure message exchange is integrated into the WE BUILD wallet ecosystem.

In WE BUILD, the secure communication channel is implemented through Qualified Electronic Registered Delivery Services (QERDS) operated by QTSPs. Whenever legal-grade delivery assurance is required, messages are routed through QERDS. QERDS providers ensure mutual authentication, end-to-end integrity and confidentiality, and interoperability across access points. This "registered delivery" pattern is positioned as an enabler for interactions between and across public sector bodies and economic operators.

Because the QERDS and the EU Digital Directory designated for the production European Business Wallet are not yet available, WE BUILD designates the pre-production QERDS specified by WP4 for use in WE BUILD business wallets. For reference, see the QERDS documentation.[8]

4.4.1 From "Registered Delivery" to "Digital Identity Wallets"

As a baseline, a classic B2G/B2B situation is used: an authority notifies an economic operator, the economic operator responds, and the relying party requires evidence. With QERDS, both sides use their QERDS providers to register sending and receiving, so that delivery is not just transport, but is a process that produces trustworthy evidence.

In this model, WE BUILD takes the next step: wallets become the user-facing endpoints ("wallet-centric delivery"). The sender wallet and recipient wallet remain the places where users read, approve, and manage messages, or where they configure connections to backend systems to perform these actions. QERDS providers form the delivery layer underneath, handling routing, inter-provider exchange, and evidence creation, while wallets provide identity/authentication and user control.

4.4.2 Technical Flow (WE BUILD High-Level)

WE BUILD follows the QERDS architecture decomposition and the four-corner delivery pattern:

1. Sender identification and authentication is performed at the sender's QTSP (wallet-driven).
2. Message submission is performed from the sender's wallet or connected backend system to the sender QERDS (QTSP A).
3. Discovery of the recipient's QERDS endpoint and capabilities is performed via common services (e.g., the WE BUILD Digital Directory, simulating the EU Digital Directory from the EBW proposal).
4. Handshake and relay is performed between QTSP A and QTSP B (QERDS-to-QERDS interoperability).
5. Recipient notification is issued, followed by recipient authentication at QTSP B.
6. Consignment and handover of the message and its metadata is performed to the recipient's wallet or connected backend system.

[8] https://github.com/webuild-consortium/wp4-architecture/blob/main/blueprint/04-integration-model.md#qerds-documentation

7. Evidence is made available to sender and recipient wallets (submission/dispatch and receipt/consignment or non-delivery). Evidence is protected by qualified sealing and, where required, qualified timestamping. Where applicable, the evidence can be pushed to the sender's and the recipient's backend systems as well.

4.5 Enterprise and System-to-System Wallet Interactions

Some WE BUILD scenarios involve interactions between backend systems rather than direct end-user actions. In these cases, wallet functionality may be integrated into enterprise platforms, APIs, or automated services.

This is particularly relevant for EBW scenarios such as supply chain credentials, Digital Product Passports, and automated B2B or B2G data exchange. In such cases, credential issuance and presentation may be initiated by backend systems while still following the interoperability patterns defined in this blueprint.

Although the interaction is system-driven, the same trust framework, credential formats, and verification mechanisms apply as in user-driven wallet interactions.

5. Information Inside the Wallet

While the previous chapter describes how wallets interact with services, this chapter describes the data and semantic structures used inside the wallet and in the exchanged attestations.

5.1 Semantic Model of the European Business Wallet

The semantic model is organised into three layers:

1. Terminology – defines terms and their abstract concepts and establishes relationships between them.
2. Vocabulary – defines classes, properties, and individuals linked to the terminology.
3. Attestation mapping – maps vocabulary terms to the elements used in attestations.

The terminology and vocabulary layers are independent of attestation formats, while the mapping layer depends on the format used.

Currently, only W3C VCDM 2.0 supports machine-readable semantic mappings directly within credentials. For mDoc and SD-JWT-VC, the meaning of data fields must instead be defined in attestation rulebooks. In these cases, semantic interoperability between attestations is not automatically enforced.

5.1.1 WE BUILD Terminology

The WE BUILD terminology[9] is published online and serves as a reference model for the terminology of the European Digital Identity Framework. The terminology is defined using the Simple Knowledge Organisation System (SKOS).[10]

5.1.2 European Business Wallet Vocabulary

The European Wallet vocabulary is maintained in GitHub. It is defined using the Web Ontology Language (OWL)[11], which specifies classes, properties and individuals of the vocabulary.

To support semantic interoperability, credential subjects used within the EBW framework are modelled in the vocabulary. These vocabulary terms are then mapped to the corresponding elements used in attestations.

If the credential format supports machine-readable semantic contexts, the mapping between credential data and the vocabulary can be embedded in the credential itself. Otherwise, the meaning of the data fields must be defined in attestation rulebooks, and

[9] https://sanastot.suomi.fi/en/terminology/webuild

[10] https://www.w3.org/2004/02/skos/

[11] https://www.w3.org/OWL/

the rulebook owner is responsible for mapping those fields to the vocabulary definitions.

Reuse of existing vocabularies The EBW vocabulary defines the domain-specific vocabulary used in the WE BUILD attestations. Existing vocabularies are reused where possible, including those for credential metadata, proof mechanisms, security, decentralised identifiers, and credential status. Domain vocabularies from other sectors may also be reused (for example, digital product passports, supply chains, education, railway and data spaces).

5.2 Attestation Rulebooks and Credential Schemas

WE BUILD defines rulebooks and credential data schemas for the attestations used in the project's use cases. Rulebooks describe requirements, roles, processes, and conformance criteria for specific attestations. They also define how credential data fields relate to the semantic vocabulary used in the project.

Credential schemas define the structure of credential data and support implementers in producing interoperable credentials and validating that the data follows the agreed format.

Rulebook descriptions are currently provided by the use cases using a common template and are maintained in the project collaboration portal. As part of the ongoing work to formalise rulebooks and credential schemas, these descriptions will be consolidated in a shared repository such as the WE BUILD Attestation Rulebooks repository[12], including rulebooks for key credentials such as PID and EBWOID.

[12] https://github.com/webuild-consortium/webuild-attestation-rulebooks-catalog/tree/main/rulebooks

6. Trust, Security and Governance

The previous chapter described the structure of the information stored in wallets and exchanged as attestations. This chapter describes the trust infrastructure that allows ecosystem participants to validate those attestations.

6.1 Trust Ecosystem

The trust infrastructure for the EUDI and EBW ecosystem is based on three complementary processes: registration/onboarding of participants, notification of certain entities to the European Commission, and publication of Trusted Lists (or Lists of Trusted Entities) that provide cryptographic trust anchors for validation.

6.2 Establishing Trust Between Participants

WE BUILD defines the onboarding processes (how entities get registered), the trust framework (which rules apply), the PKI architecture (which certificates are used and how), the APIs used to query trust information programmatically, and the trust evaluation logic used by participants at runtime.

The infrastructure is based on the Trusted List model defined in the eIDAS Regulation and the ARF. It follows the European model in which a List of Trusted Lists (LoTL) points to Trusted Lists. Each Trusted List contains entries for authorised participants such as PID Providers, Attestation Providers (QEAA, PuB-EAA, non-qualified EAA), Wallet Providers, and Relying Parties.

The onboarding processes define how participants join the ecosystem. This includes how:

- **Relying Parties** register, accept policies and configure access controls
- **PID and Attestation Providers** register, declare supported attestation types and obtain registration and access certificates
- **Wallet Providers** register and issue wallet instance attestations
- **Trust Service Providers** register and publish relevant certificates

Once onboarding is completed, participants use the trust infrastructure to evaluate each other during normal operation. WE BUILD therefore defines a set of trust evaluation scenarios covering how participants verify each other at runtime.

These scenarios include:

- a Wallet Unit evaluating a Credential Issuer before requesting a PID or attestation
- a Credential Issuer evaluating the Wallet Unit before issuing
- a Wallet Unit evaluating a Relying Party before presenting attributes
- a Relying Party evaluating presented credentials (PID, QEAA, PuB-EAA, non-qualified EAA)
- discovery and consumption of the LoTL and TLs.

For detailed information on authorities, registries and responsibilities, see Appendix C - Trust Ecosystem.

For reference on relying party access certificates and relying party registration certificates, see the RPAC/RPRC documentation.[13]

6.2.1 Trust infrastructure architecture (overview)

In the Appendix - Trust Ecosystem there is a diagram that summarizes the roles of Member State and European Commission, the split between registration and notification, and how Trusted Lists and the LoTL are produced and consumed. A simplified version used in WE BUILD is shown below.

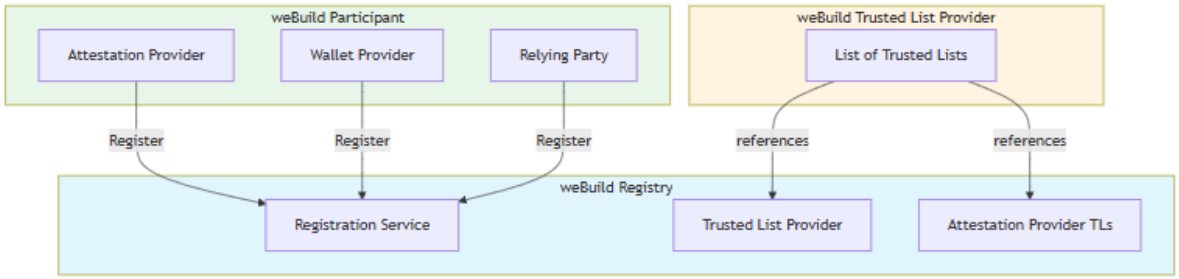

Figure 9: A simplified version of the Trust Ecosystem used in WE BUILD.

WE BUILD participants select the registry in which they register.

6.3 Revocation

Revocation ensures that attestations that are no longer valid can no longer be trusted or used.

WE BUILD distinguishes between attestation revocation, which is handled by issuers, and revocation or withdrawal of providers and services, which is reflected in the trust infrastructure.

6.3.1 Technical realisation

Revocation of PID, EBWOID and attestations is implemented by issuers. In WE BUILD, attestation revocation follows the agreed mechanism defined in the ADR on Attestation Revocation, based on the IETF Token Status List and aligned with OpenID4VC HAIP.

Short-lived attestations (valid for 24 hours or less) are not subject to revocation.

Revocation or withdrawal of providers and services is reflected in the trust infrastructure through status changes in Trusted Lists and, where applicable, invalidation of certificates.

[13] https://github.com/webuild-consortium/wp4-architecture/blob/main/blueprint/06-trust-and-security.md#rpacrprc-documentation

6.3.2 Provider Obligations

To maintain a trusted ecosystem, PID and EBWOID providers agree to:

- Define and publish revocation policies.
- Ensure that only the issuing authority can revoke its attestations.
- Publish revocation status information within a reasonable time frame.

6.3.3 Conditions for Mandatory Revocation

According to the rules, a provider must revoke without delay if:

- The holder explicitly requests it.
- The security of the wallet app itself (the unit certificate) is compromised.
- Any of the specific situations defined in the provider's public policy occur.

7. Architecture Governance: ADRs and WBCS

While the previous chapter described the operational trust infrastructure, this chapter describes the governance model used to define and maintain the technical architecture of the WE BUILD ecosystem. In a project as large as WE BUILD, interoperability between independently developed components must be ensured without requiring every developer to participate in all coordination meetings.

To maintain alignment, the project uses a technical governance model based on consensus, commitment and clear documentation. Technical choices are driven by the needs of the 13 use cases implemented in the project.

7.1 Architectural Decision Records (ADR)

The ADRs is essentially our project's "logbook" for major decisions.

- **Purpose:** The ADR process is where we formally capture and justify significant technical choices, such as which specific protocols and formats to use. Instead of having these decisions buried in a slide deck or a long email chain, we document the rationale and context so that everyone can understand the "Why" behind a choice.
- **Classification:** We maintain a lightweight ADR for any software-related decision that affects how different systems work together (interoperability). This ensures alignment with external rules like the eIDAS Regulation and the ARF.
- **Lifecycle:** ADRs are managed on GitHub (webuild-consortium/wp4-architecture/adr[14]). They move from a "Proposed" state to "Accepted" once the Architecture Group and relevant stakeholders reach consensus.

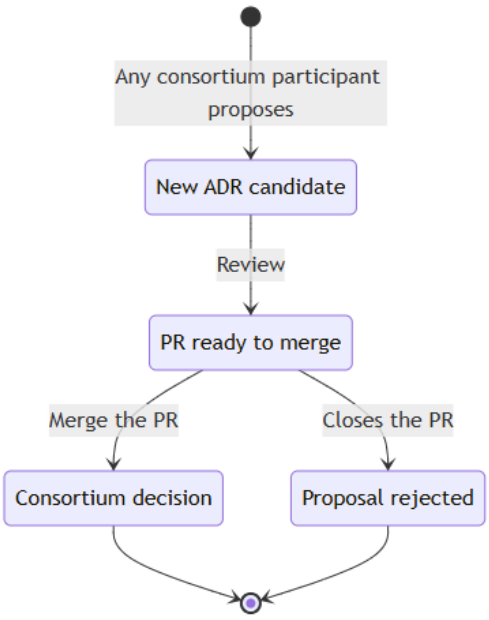

Figure 10: ADR process

[14] https://github.com/webuild-consortium/wp4-architecture/adr

7.2 WE BUILD Conformance Specifications (WBCS)

If ADRs capture the rationale ("why"), the WBCS[15] define the implementation requirements ("how").

- **Operationalising Intent:** We use the WBCS to turn high-level architectural goals into detailed technical rules. These specifications define the exact interfaces for wallets, issuers, and verifiers.
- **A Commitment to Implement:** This is the most important part: An approved WBCS is not just a suggestion**.** When a specification is approved, it signifies a commitment from the participating organizations to actually build that interface into their services.
- **Defining Implementation Requirements:** Because the WBCS define how interfaces and protocols must be implemented, they allow us to achieve interoperability across the whole consortium. If you follow the WBCS, you avoid building an "interoperable island" where your service only works with a few specific partners.
- **The Link to Testing:** Our Interoperability Testbed (ITB) uses these specifications as its primary rulebook. Implementations that do not follow the WBCS will not pass the ITB tests and are therefore not eligible for pilot participation.

[15] https://github.com/webuild-consortium/wp4-architecture/tree/main/conformance-specs

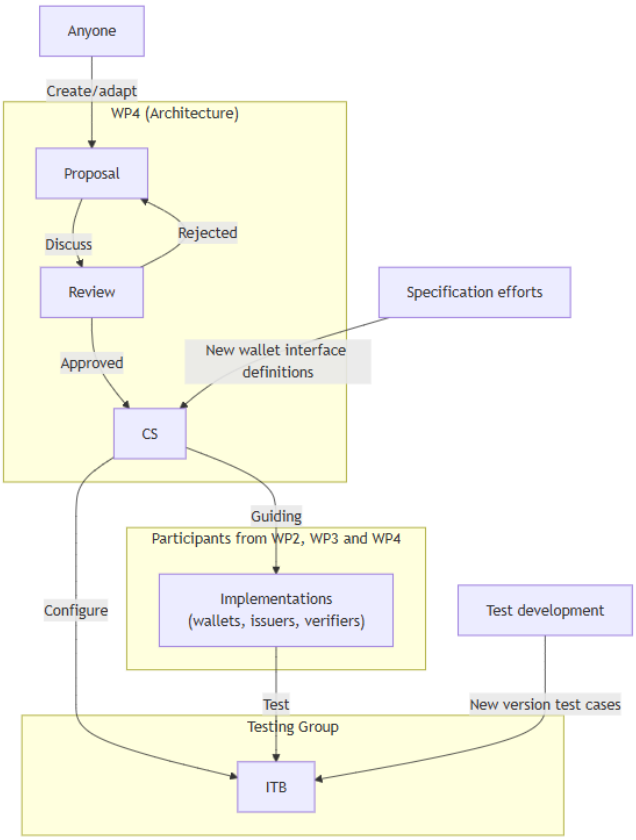

Figure 11: WBCS Process

7.3 Document Lifecycle

WE BUILD moves fast, and our documentation needs to keep up. We don't wait for "perfect" documents; we iterate as the use cases mature.

- The Blueprint as a Living Framework: This Blueprint (D4.1) sets the high-level structure, but it is supported by the more agile ADRs and WBCS that live on GitHub. As we learn, we update these records and specifications.
- The Hybrid Working Flow: To keep things moving, we use a "hybrid" approach to our document lifecycle:
    - GitHub: This is our source of truth for all accepted specifications and decision records.
    - Slack: The ITB uses a dedicated channel for implementation support, where developers can ask questions and help each other in real-time.
    - Meetings: We hold interface-alignment meetings to discuss progress, resolve gaps, and gain final agreement on new specifications.
- Maturing Together: As the project moves forward, we will add more detailed definitions to the documentation stack. This approach allows the Blueprint to evolve from a high-level architectural reference into a practical guide for implementing the WE BUILD ecosystem.

8. Testing and the Interoperability Testbed (ITB)

Once architectural decisions and specifications are defined, interoperability must be verified in practice. This chapter describes how interoperability is verified through the WE BUILD testing strategy and the Interoperability Testbed (ITB).

8.1 Testing Strategy

The Architecture Group coordinates the architectural building blocks and ensures alignment with the project use cases.

The Testing Group develops test cases and test suites for:

- Generic test cases based on WBCS.
- Functional test cases for required features (based on WBCS and, when needed, rulebooks and/or data schemas).
- End-to-end and piloting test cases for WP2/WP3 use cases (based on existing WBCS, rulebooks and data schemas).

To implement tests in the ITB, the Testing Group needs the specification artefacts: WBCS, rulebooks, data schemas, namespaces, and related metadata. The Architecture Group ensures that these artefacts are complete and consistent with the overall architecture and supports WP4 groups and WP2/WP3 use cases in providing the required input.

Most specification artefacts are produced within WP4:

- The Semantics Group: attestations (data schemas, namespaces, and relevant rulebook parts).
- The Wallet, PID/EBWOID and QTSP Group: WBCS and commitment to implement them.
- The Architecture Group: Architecture Decision Records (ADRs) that define the allowed scope for WBCS.
- The Trust Infrastructure Group: validation and verification requirements to be reflected in test cases.

For piloting-specific test suites, the Testing Group collaborates directly with the relevant use case(s). The Architecture Group acts as a facilitator to ensure consistency across the involved specifications.

8.2 Test Requirements

Test cases are derived from the WBCS.

WBCS must stay within the scope defined by the published ADRs. If a WBCS needs functionality beyond that scope, it requires an ADR discussion. Testing focuses on features implemented by multiple parties, since interoperability requires multi-party implementations.

Implementing participants discuss WBCS together with the use cases that require the functionality.

The ITB initially includes two credential-agnostic test suites:

- Issuing (based on OpenID4VCI v1.0)[16]
- Verifying (based on OpenID4VP v1.0)[17]

If a use case requires different functionality, it can propose a new or adapted (draft) WBCS. Once the WBCS and required supporting artefacts are available, the Testing Group implements the corresponding test cases in the ITB.

Some test cases require additional artefacts beyond the WBCS, such as rulebooks for attestation-specific requirements, and the corresponding data schemas, namespaces, and metadata.

When the required artefacts are available, the Testing Group implements the test cases in the ITB and communicates their availability to the consortium.

8.3 Additional Documentation

The ITB on GitHub[18]

A user guide[19] on how to onboard and execute tests.

Documentation on the ITB and integrations[20]

[16] https://github.com/webuild-consortium/wp4-architecture/blob/main/conformance-specs/cs-01-credential-issuance.md

[17] https://github.com/webuild-consortium/wp4-architecture/blob/main/conformance-specs/cs-02-credential-presentation.md

[18] https://github.com/webuild-consortium/wp4-interop-test-bed

[19] https://github.com/webuild-consortium/wp4-interop-test-bed/blob/main/docs/user-guide-interoperability-test-bed.md

[20] https://github.com/webuild-consortium/wp4-interop-test-bed/blob/main/docs/reference-implementation-interoperability-test-bed.md

9. What's Next & Scaling Up

This document, together with the initial attestation data models and schemas, establishes a common technical baseline for the wallet ecosystem. The architecture will continue to evolve throughout the project as the focus shifts towards implementation and testing.

WP4 - General Capabilities operates on a defined timeline with key milestones and deliverables while adapting to feedback from the use cases as well as business, regulatory and technological developments. The deliverable is expected to evolve through updates and refinements as new ADRs and WBCS are added or revised. The document therefore serves as a living reference for the WE BUILD architecture.

The roadmap for months 8–19 focuses on maturing the trust infrastructure and validating cross-border interoperability through the project's automated testbed.

9.1 Key Milestones and Deliverables (Months 8–19)

Month	Type	Reference and Title	Connection to D4.1
10	Deliverable	D4.3 Attestation Data Models & PID/EBWOID Rulebook	Finalizes the data structures for the issuance patterns defined in D4.1
11	Milestone	MS5 WP2 Pilot Design Complete (Business)	Finalizes the scope of business use journeys based on D4.1 architectural patterns
11	Milestone	MS10 WP3 Pilot Design Complete (Payments)	Finalizes the scope of payments and banking use journeys based on D4.1 architectural patterns
12	Deliverable	D2.1 WP2 Pilot Design Report & PID/EBWOID issuance rulebook	Maps user journeys to the D4.1 reference implementations
12	Deliverable	D3.1 WP3 Pilot Design Report	Maps payment journeys to the D4.1 reference implementations
13	Deliverable	D4.4 Trust Infrastructure Guidelines	Details the technical signature/seal flows introduced in D4.1
17	Milestone	MS6 WP2 Pilot Implementation Complete	Validates local business infrastructure against D4.1 specifications
17	Milestone	MS11 WP3 Pilot Implementation Complete	Validates local payment infrastructure against D4.1 specifications
19	Milestone	MS7 WP2 Cross-Border Readiness	Proves interoperability of WP2 and WP4 business ecosystem
19	Milestone	MS12 WP3 Cross-Border Readiness	Proves interoperability of WP3 and WP4 payments and banking ecosystem

19	Milestone	MS17 Wallets & Trust Infrastructure Ready	Final evidence of wallet/trust interoperability as per D4.1

Table 3: Milestones and Deliverables connected to D4.1

Alongside these milestones and deliverables, WP4 will continue to iterate on the architecture, specifications and supporting artefacts throughout the project. All WP4 groups will incorporate feedback from the piloting phase and adapt to new requirements and emerging EUDI standards and other technological developments. The ITB will remain a vital tool for continuous integration and testing, ensuring that the solutions remain interoperable, secure, and scalable.

Appendix A. Glossary

Terms and Definitions

This appendix defines the key terms, regulatory frameworks, and technical specifications utilised throughout the WE BUILD ecosystem.

While this document avoids abbreviations as much as possible, commonly used abbreviations are included for reference.

Term	Abbreviation	Definition
Architectural Decision Record	ADR	A document used to capture and justify significant technical choices. ADRs serve as the project's "logbook" to ensure transparency regarding the rationale behind protocol and standard adoption.
Architecture and Reference Framework	ARF	The reference architecture for the European Digital Identity Wallet ecosystem published by the European Commission in cooperation with the Member States. It defines roles, trust models, protocols and interoperability requirements for the ecosystem.
Attestation Rulebook	-	A document describing the governance, requirements and semantic interpretation of a specific attestation type, including how credential data maps to vocabulary terms and schemas.
Blueprint	—	The high-level architecture and integration document (D4.1) describing the WE BUILD ecosystem, architectural patterns, interaction flows and governance model.
Business Wallet Unit Attestation	BWUA	A specific type of Wallet Unit Attestation issued for a European Business Wallet (EBW) instance.
EAA Provider	—	An entity that relies on authentic sources of information to issue attestations to a wallet.
EBW Instance	—	A unique deployment or installation of a European Business Wallet (EBW) solution, controlled by an Owner (legal person or economic operator).
EBW Provider	—	A Wallet Provider specifically authorized to issue and manage European Business Wallets (EBW).
EBWOID Provider	—	An entity responsible for verifying the identity of a legal person or economic operator and issuing EBW Owner Identification Data (EBWOID).
EBW Owner Identification Data	EBWOID	A set of attributes used to uniquely identify a legal person or economic operator within the European Business Wallet ecosystem.
Economic operator	—	Any natural or legal person or public entity which offers products or services on the market; the primary user of the European Business Wallet.
Electronic Attestation of Attributes	EAA / QEAA / PuB-EAA	Digital credentials that prove specific attributes (e.g., professional qualifications, representation rights) with either qualified (QEAA) or public sector body-issued (PuB-EAA) or non-qualified (EAA) legal status.
Electronic Identification, Authentication and Trust Services	eIDAS / eIDAS 2.0	The legal framework for electronic identification and trust services for electronic transactions in the European Single Market.

European Business Wallet	EBW	A wallet designed for economic operators or public sector bodies to manage business data such as mandates, electronic invoices, and administrative and professional documents and notifications.
European Digital Identity Wallet	EUDI Wallet	A mobile or cloud-based solution for natural persons to manage and share identity data.
EUDIW Instance	—	A specific deployment of an EUDI Wallet solution for a natural person.
	—	See *Wallet User* instead.
Interoperability	-	The ability of independently developed systems and components to exchange information and correctly interpret the exchanged data.
Interoperability Testbed	ITB	The automated testing environment used in WE BUILD to verify that implementations conform to the agreed specifications and remain interoperable.
	—	See *EAA Provider* instead.
Large Scale Pilot	LSP	A project funded by the European Commission to test the practical implementation of the EUDI Wallet framework across various cross-border use cases.
Legal Person	—	An entity (such as a corporation or public body) recognized by law as having rights and duties, distinguished from a natural person.
	LPID	See **EBW Owner Identification Data** instead.
Level of Assurance	LoA	A classification of the degree of confidence in the electronic identification of a natural person, a legal person, or a natural person representing a legal person. Recognised levels are: Low, Substantial, High.
List of Trusted Lists	LoTL	A list that references national or ecosystem Trusted Lists, allowing participants to discover and validate trusted entities.
Natural Person	—	An individual human being acting in their own capacity.
Owner	—	The legal person or economic operator that has legal control over and responsibility for an EBW Instance.
Personal Identification Data	PID	A mandatory set of attributes issued to a natural person to uniquely identify them at Level of Assurance (LoA) High.
PID Provider	—	An entity responsible for verifying the identity of a natural person and issuing Personal Identification Data (PID).
Qualified Electronic Registered Delivery Service	QERDS	A secure communication channel that provides legal evidence of the handling of transmitted data.
Qualified Trust Service Provider	QTSP	A regulated entity providing electronic trust services (e.g., signatures, seals, or delivery services) with full legal effect under eIDAS.
Relying Party	RP	An entity that requests and receives attestations from a wallet to verify specific attributes or identities.
Selective Disclosure JSON Web Token	SD-JWT	A format allowing holders to share only specific parts of a credential while keeping other data private.
Trust Framework	—	The set of governance rules, standards, and trust infrastructure used to establish and verify trust relationships between ecosystem participants.
Trusted List	TL	A machine-readable list of trusted service providers or entities used to validate trust relationships within the ecosystem.
	—	See *Relying Party* instead.
Wallet Application	—	The user-facing software component of a Wallet Solution providing the interface for managing credentials.

Wallet Core Component(s)	—	The technical module(s) of a Wallet Solution handling cryptographic operations and protocol implementations.
Wallet Instance	—	A specific operational instance of a wallet solution running on a device or cloud environment.
Wallet Instance Attestation	WIA	A short-lived, signed information object issued by a Wallet Provider that contains information about the Wallet Instance. It is device-bound and presented to PID or Attestation Providers to authenticate the instance, but it does not require a WSCD/WSCA for key management and does not contain revocation information.
Wallet Provider	—	An organization that provides a Wallet Solution and manages its lifecycle.
Wallet Secure Cryptographic Device / Application	WSCD / WSCA	The hardware or software environment used to manage cryptographic keys securely within the wallet.
Wallet Solution	—	A specific implementation of a wallet consisting of a Wallet Application and Wallet Core Component(s).
Wallet Unit Attestation	WUA	A signed information object issued by a Wallet Provider that describes the capabilities and security properties of a Wallet Unit (especially the WSCD/WSCA). It is device-bound and allows PID or Attestation Providers to verify compliance, bind credentials to the unit, and check for revocation.
Wallet User	—	The natural or legal person that controls and operates a wallet instance.
WE BUILD	—	The consortium and project focused on pioneering the European Business Wallet and EUDI Wallet use cases.
WE BUILD Conformance Specifications	WBCS	Detailed technical rules that operationalize architectural intent. Approval of a WBCS signifies a commitment from partners to implement those interfaces.

Appendix C. Trust Ecosystem

The appendix describes the full trust ecosystem for the EUDI Wallet and European Business Wallet, including roles and responsibilities of Member States and the European Commission.

It provides a more complete view of the target ecosystem beyond the WE BUILD pilot context, which operates with adapted or simulated components.

The trust infrastructure for the EU Digital Identity and European Business Wallet ecosystem rests on three distinct but complementary processes: **registration/onboarding** of participants, **notification** of certain entities to the European Commission, and **publication of Trusted Lists** (or Lists of Trusted Entities) that provide cryptographic trust anchors for validation. WE BUILD aligns with the EUDI Wallet Architecture and Reference Framework (ARF) and the trust-infrastructure model described in the WP4 Trust Group deliverables.

Trust infrastructure authorities and registries

- **Member State Registrar:** Manages registration and operational authorization of **PID Providers**, **Attestation Providers**, and **Relying Parties**. Registration yields registry entries (used for entitlement verification and online lookup via common APIs such as TS5/TS6) and triggers access certificate issuance.
- **European Commission:** Compiles, signs/seals, and publishes Trusted Lists for Wallet Providers, PID Providers, Access Certificate Authorities (Access CAs), and Providers of Registration Certificates. It maintains the **List of Trusted Lists (LoTL)** and publishes LoTL location and trust anchors in the Official Journal of the European Union (OJEU).
- **Member State Trusted List Provider (MS TLP):** Compiles, signs, and publishes national Trusted Lists for **non-qualified EAA Providers** and **Member State QTSP Trusted Lists** for **QEAA Providers** (per Article 22 eIDAS), and submits the Trusted List URLs to the Commission for inclusion in the LoTL.
- **Access Certificate Authority:** Issues access certificates to registered entities (PID Providers, Attestation Providers, Relying Parties). Notified by Member States to the Commission; does not register with Registrars.
- **Provider of Registration Certificates:** Optionally issues registration certificates that detail entitlements; notified by Member States to the Commission.

Registration vs Trusted List publication: Registration defines *who is allowed to do what* (entitlements, attributes, intended use) and is consumed via registries and optional registration certificates. Trusted List publication establishes *cryptographic trust anchors* (keys, certificates) and, via profile-specific extensions defined in **ETSI TS 119 602** (for example the Pub-EAA and national non-qualified EAA Provider LoTE profile in Annex H, including its additionalInfo structures), can also publish **which attestation**

types an Attestation Provider is authorised to issue. Trusted Lists follow ETSI TS 119 612 and TS 119 602 (Lists of Trusted Entities) and are consumed per ETSI TS 119 615. Wallet Providers, Access CAs, and Providers of Registration Certificates are **not** registered with Registrars; these entities are notified by Member States to the Commission.

Responsibilities matrix

The Task 2 trust-infrastructure schema defines the following responsibilities matrix for registration and Trusted List compilation:

Entity Type	Registration Process	Trusted List Compilation (EC / MS TLP)	Member State TLP Role
PID Provider	Register with MS Registrar	European Commission (EU-level TL for PID Providers)	None (no national TL for PID Providers)
Attestation Provider	Register with MS Registrar	Member State / MS TLP (national QTSP TL for QEAA Providers; national TL for non-qualified EAA Providers)	Compiles, signs, and publishes national Trusted Lists (QTSP TL for QEAA Providers per Article 22; EAA Provider TL for non-qualified EAA Providers)
Relying Party (RP)	Register with MS Registrar	N/A (uses Access Certificates and Registry)	None (not listed in Trusted Lists)
Wallet Provider	Notification only (by MS to EC)	European Commission (EU-level TL for Wallet Providers)	Not applicable in MVP (notification from MS to EC only)
Access CA	Notification only (by MS to EC)	European Commission (EU-level TL for Access CAs)	Not applicable in MVP (notification from MS to EC only)
Reg. Cert. Provider	Notification only (by MS to EC)	European Commission (EU-level TL for Reg. Cert. Providers)	Not applicable in MVP (notification from MS to EC only)

This blueprint section mirrors the Task 2 responsibilities matrix so that architectural roles are consistent across WP4.

Working group scope: [MVP] and [MVP+]

In line with the EUDI Wallet ARF, the WP4 Trust Group focuses on defining **architectural patterns and profiles**, not on specifying Member State-specific policies or operating production infrastructure. To make this concrete, the trust and security work is scoped in two steps:

- **[MVP] (Minimum Viable Prototype)**:
    - Implements the **core onboarding scenarios** from Task 1 (Subtask 1.1) for PID Providers, Attestation Providers, Wallet Providers, Relying Parties, and Certificate Authorities.
    - Implements the **basic trust-registry scenarios** from Task 1 (Subtask 1.2) needed to create, publish and consume Trusted Lists / Lists of Trusted Entities and registry entries for these actors.
    - Demonstrates end-to-end flows for **registration, access-certificate issuance, trust-anchor publication and consumption**, reusing the ARF

and ETSI TS 119 602/119 612/119 615 patterns without introducing new normative profiles.

- **[MVP+] (Extended prototype)**:
    - Completes the remaining Task 1 onboarding and trust-registry scenarios, including more advanced **evaluation, maintenance, revocation and discovery** cases.
    - Covers **richer combinations of participants and roles** (e.g. multiple types of Attestation Providers and more complex Relying Party ecosystems) while staying within the Task 2 trust framework and trust-infrastructure schema.
    - May introduce **pilot-specific configurations or conventions** (e.g. additional metadata, policy examples, or Trusted List extensions) as long as these remain compatible with the underlying ARF and ETSI models and are clearly marked as non-normative.

The **boundary of the working group** is therefore to: (a) define the trust-infrastructure architecture, profiles, and flows needed for [MVP] and [MVP+]; (b) document how to apply ETSI TS 119 602/119 612/119 615 and the ARF in these scenarios; and (c) leave Member State policy choices (approval criteria, national extensions, operational SLAs) and long-term production operation out of scope.

Trust infrastructure architecture (overview)

The following diagram summarizes the roles of Member State and European Commission, the split between registration and notification, and how Trusted Lists and the LoTL are produced and consumed. Source: WP4 Trust Group Task 2, trust-infrastructure schema. Go to Appendix C online for higher resolution.

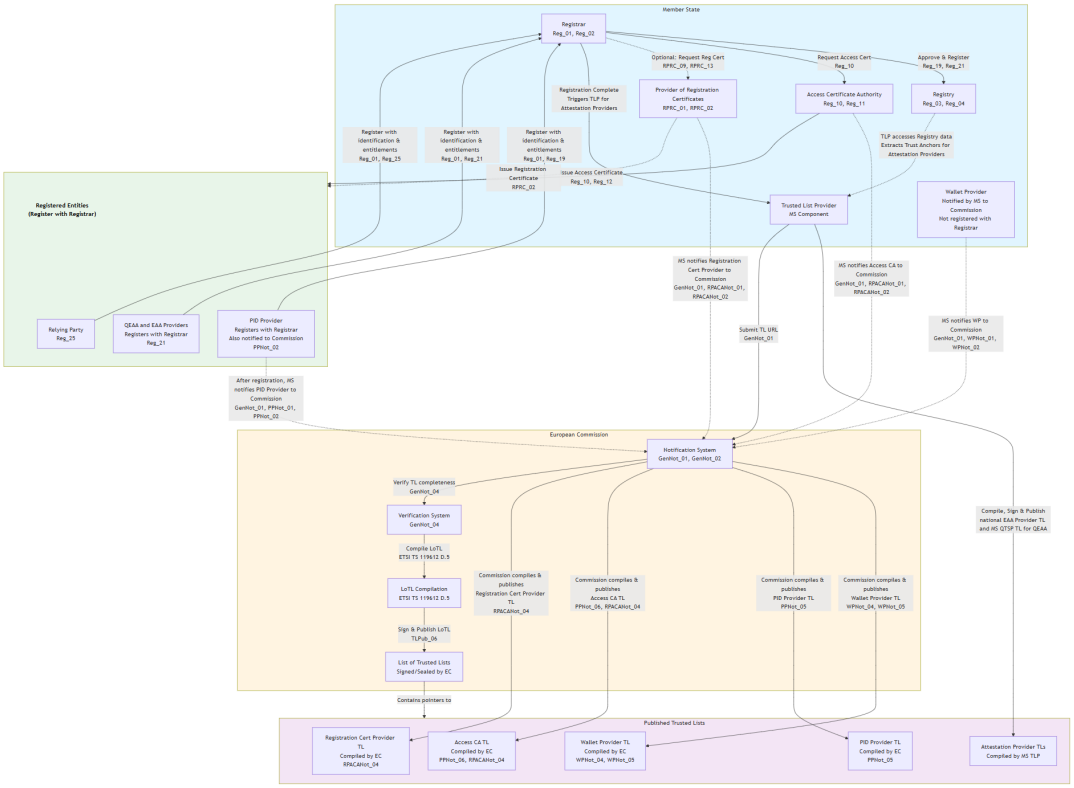

Figure 12: Overview of the rust infrastructure architecture.

Note: WP4 is going to expose trust infrastructure depicted on this diagram, mimicking the infrastructure of at least one Member State. In the case of mimicking more than a single Member State, WE BUILD participants willing to register are going to be able to select a registry in which they are going to be registered, or the WP4 registrar, if single, is going to place them in one of the registries. For the sake of simplicity, in such a case, not all the technical components depicted in the diagram within a Member State will have to be multiplied; e.g. there may be multiple registries but a single trusted list across the countries to which all wallet-relying parties are provisioned.

Security Measures

From an architectural perspective, security in the wallet ecosystem is structured in four dimensions. (1) **Trust anchor layer**: cryptographic validation and key lifecycle management, including revocation of keys, certificates, and services. Trust Anchors are published by Trusted Lists and related mechanisms (ETSI TS 119 602/119 612/119 615), applying to the entities and roles described in the Trust Ecosystem (PID Providers, Attestation Providers, Wallet Providers, Access CAs, etc.). (2) **Identity assurance**: the level of assurance (LoA) of the identities involved is maintained across the full lifecycle of those identities. (3) **Device and execution environment**: the security of the devices and execution environments that host Wallet Instances and cryptographic material (WSCA/WSCD) is addressed by the wallet secure cryptographic application/device (WSCA/WSCD) architecture for wallet-side components, to be specified by the Architecture and Wallets groups (and, for remote WSCA/WSCD, together with the QTSP group). Secure environments operated by PID Providers, Pub-EAA and QEAA Providers, and Trust Service Providers (TSPs) are addressed by applicable eIDAS and ETSI requirements for TSP and provider infrastructure. (4) **Protocol and policy layer**:

authentication (verifying who or what is acting) and authorization (what the subject is allowed to do, driven by policies) are realised in conformance with the ARF and related technical specifications, per application-specific flow and per attestation data format.

Authentication

Authentication in the WE BUILD architecture closely follows the standards and flows of the underlying protocols (**OpenID for Verifiable Credentials**, **ISO 18013-5**, and other application-specific communication protocols).

For **organizational entities** that register with the Registrar (PID Providers, Attestation Providers, Relying Parties), authentication is primarily established using **access certificates** issued by the Access Certificate Authority, validated against up-to-date Trusted Lists or Lists of Trusted Entities (ARF and ETSI TS 119 602 / 119 612 / 119 615). **Wallet Providers** are notified by Member States to the European Commission (they do not register with the Registrar); their authenticity is established via the Wallet Provider Trusted List and related attestations (e.g. Wallet Unit Attestation). Mutual trust is strictly required in application-specific protocol flow specifications, and consolidated through OpenID HAIP and ARF HLRs, with certificate-bound tokens and protocol-level message signatures along with endpoint authentication and message integrity.

According to the ARF, the Relying Party **cannot request the Wallet Unit Attestation (WUA) during the presentation flow**. Presentation requests address **PID and attestations** only (ARF Topic 1, **OIA_01**); the WUA is presented to the PID Provider or Attestation Provider **during issuance** of a PID or device-bound attestation, not to the Relying Party (ARF Topic 9, **WUA_03**, **WUA_05**, **WUA_05a**). The ARF explicitly states that there is no separate mechanism for the Relying Party to verify the revocation status of a Wallet Unit directly with the Wallet Provider (ARF Section 6.6.3.12). Trust in the Wallet Unit is therefore **mediated by a trusted third party**: the PID Provider or Attestation Provider that belongs to a Trust Anchor (Trusted List) and that received the WUA at issuance. That trust is **indirect** from the Relying Party's perspective. The PID Provider or EAA Provider **periodically checks the revocation status** of the Wallet Unit to which it has issued credentials (using the revocation information in the WUA received at issuance; ARF Topic 9 **WUA_02**, ARF Section 6.6.2.4). If the Wallet Unit is revoked, the PID Provider or Attestation Provider **SHALL revoke** the credentials it issued to that Wallet Unit (Article 5, 4.(b), European Digital Identity Regulation; ARF Section 6.6.2.4). By verifying the revocation status of the PID or attestation, the Relying Party implicitly relies on the issuer's verification of the Wallet Unit.

For **attestation holders** (individual end-users and the corresponding wallets), authentication requirements leverage possession-based proofs and, where applicable, **identity schemes notified to the European Commission** that attest **Level of Assurance High** according to the eIDAS Framework. Local user authentication (e.g., via PIN, password, device biometrics) must fulfill the minimum Level of Assurance

required by the credential or attestation type and is enforced by policy prior to any issuance or presentation flow.

Wallet Units are expected to combine protocol-specific authentication mechanisms (as per OpenID4VC and ISO 18013-5) with validation of trust anchors and up-to-date Trusted Lists, covering both participant and credential authenticity.

Authorization and policies

Authorization determines what actions a subject is permitted to perform after authentication. In line with the Task 2 trust framework and the EUDI Wallet ARF, WE BUILD **assumes** that **the default is "allow all"** at the ecosystem level, and that **policies (expressed via Trusted List extensions, registration/entitlement data, and optional registration certificates) can tighten this to an effective "deny all except explicitly allowed"** model for specific contexts and participants. When such policies apply, only the actions and attribute uses listed in the applicable allow-lists are permitted; all others are denied. Trust marks (for Credential Issuers, Wallet Solutions, and Relying Parties), together with Trusted List extensions and registration certificates, carry authorization semantics (e.g. authorised credential types, attribute groups, purposes, scope restrictions) and are used in policy evaluation and collision prevention as specified in the Task 2 trust framework and the trust-infrastructure schema.

According to the EUDI Wallet ARF v2.8, when present and applicable, policies and default authorisations may be overridden by **user will**: the Wallet Unit SHALL ensure the User approved the presentation of any attribute(s) prior to presenting those attributes and SHALL always allow the User to refuse presenting an attribute requested by the Relying Party or Verifier Wallet Unit (ARF Topic 6, **RPA_07**); if Relying Party authentication fails, the Wallet Unit SHALL either not present the requested attributes or give the User the choice to present or not (**RPA_06a**).

Certificates and cryptographic anchors

- **Trusted Lists / Lists of Trusted Entities (LoTE)** (ETSI TS 119 612, TS 119 602) are pivotal trust anchors in the ecosystem. LoTE entries publish the keys and related metadata for the entity types described in the Trust Ecosystem (Wallet Providers, PID Providers, Attestation Providers, Access CAs, Registration Cert Providers). Validation of trust service outputs against these lists SHALL follow **ETSI TS 119 615** (procedures for using and interpreting EUMS national trusted lists).
- **Access certificates** are issued by the Access Certificate Authority to registered PID Providers, Attestation Providers, and Relying Parties. Issuance SHALL comply with **ETSI TS 119 411-8**; the Authority SHALL comply with at least **ETSI EN 319 411-1** Normalised Certificate Policy (NCP) requirements. Each Relying Party receives a **separate access certificate per Relying Party Instance**. Access

certificates authenticate entities in protocol exchanges and are validated by Wallet Units using the trust anchors in the Access CA LoTE entries.

- **Registration certificates** (optional) may be issued by the Provider of Registration Certificates to detail registration status and entitlements. When the User opts to verify RP (or issuer) registration, Wallet Units use the registration certificate when provided and/or registry lookup, as specified in ARF RPRC_16 to RPRC_21.

For reference on relying party access certificates and relying party registration certificates, see the RPAC/RPRC documentation.

Key lifecycle and Trusted Lists

Key lifecycle for trust anchors and for services listed in Trusted Lists is reflected in list content and status (e.g. service status, status determination approach, and history where required by the profile). Updates and revocation of listed services follow the applicable ETSI trusted list profiles (TS 119 612 / TS 119 602) and are consumed per ETSI TS 119 615. Certificate policies and Certificate Transparency (SCT) where applicable are specified in the referenced ETSI standards.

Relying Party Registration & Access Certificates

Relying Parties (verifiers) **register with the Member State Registrar** before being able to securely identify themselves to Wallet Units. Registration includes identification data, the **attributes** the RP intends to request from Wallet Units, the **intended use** (purpose), and, if applicable, use of intermediaries. The Registrar approves the RP (per ARF Reg_25) and publishes the entry in the **Registry**. The **Access Certificate Authority** then issues access certificates to the RP as described under Certificates and cryptographic anchors, with a **separate access certificate per Relying Party Instance** (Reg_10a). Optionally, the Registrar may request a **registration certificate** from the Provider of Registration Certificates (RPRC_09) that summarises registration status and entitlements. Wallet Units authenticate RPs by validating RP access certificates against the Access CA trust anchors and by verifying registration and requested attributes in the Registry (RPA_04, RPRC_16, RPRC_21). The common API for RP registration information (e.g. TS5) and the common set of RP information (e.g. TS6) are specified in the EUDI Wallet technical specifications. Detailed flows and diagrams are in the WP4 Trust Group trust-infrastructure schema. Certificate issuance aspects are coordinated with the QTSP group.

The following sequence illustrates how a Wallet Instance discovers and validates Relying Party policy during presentation (WRPAC = Relying Party Access Certificate; WRPRC = Relying Party Registration Certificate). Source: WP4 Trust Group, wallet-policy-discovery. Go to Appendix C online for higher resolution.

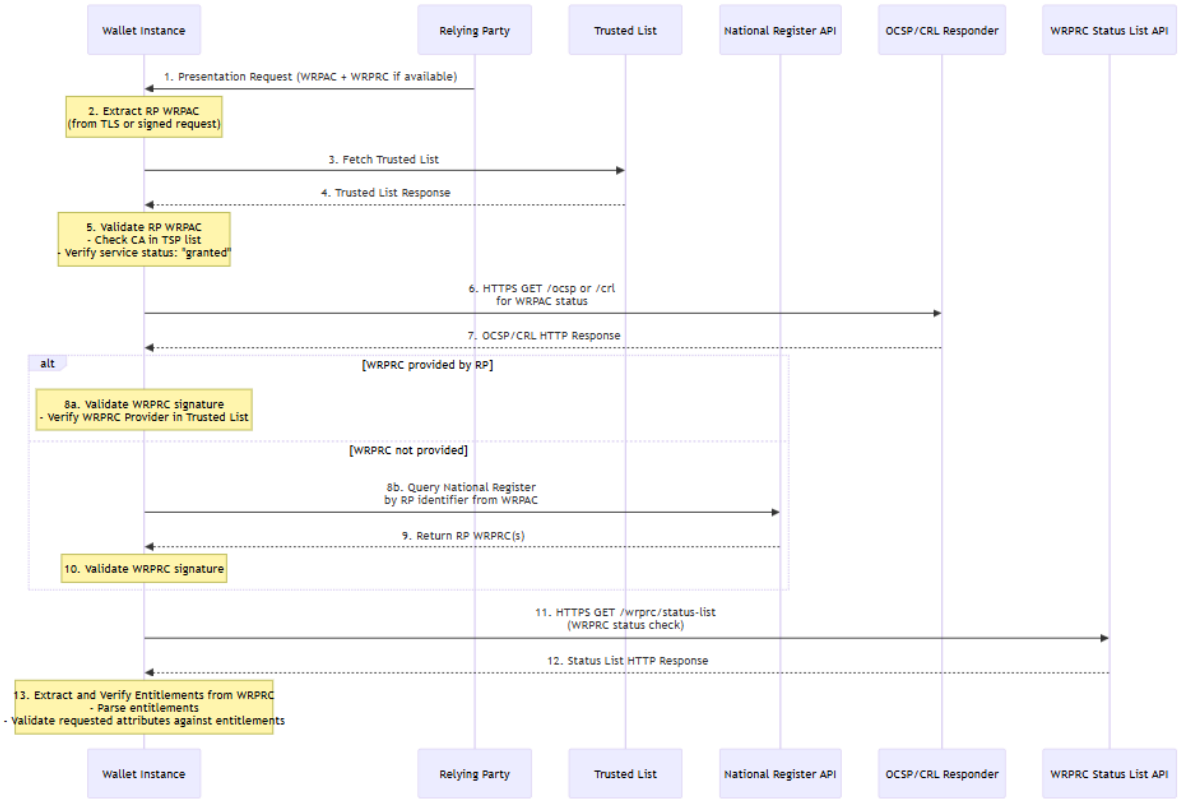

Figure 13: Discovery and validation of a Relying Party policy by a Wallet Instance during presentation.

Validation Functions for Relying Parties

Relying Parties need to **authenticate and validate** the Person Identification Data (PID) and attestations (e.g. LPID/EBWOID, EAA) received from Wallet Units. Validation relies on the trust infrastructure as follows:

PID and LPID/EBWOID: Relying Parties validate the PID (or equivalent) signature using the **List of Trusted Entities (LoTE)** for PID Providers, i.e. the PID Provider Trusted List compiled by the European Commission and referenced from the LoTL. Procedures for authenticating the LoTL and national/EU trusted lists and for obtaining listed services are given in **ETSI TS 119 615**.

Attestations (EAA): Relying Parties validate **qualified EAA (QEAA)** signatures using the **Member State QTSP Trusted Lists** (per Article 22 eIDAS) and **PuB-EAA** (and, where applicable, national non-qualified EAA) using the relevant Attestation Provider Trusted Lists. Trust anchors and service types are defined in ETSI TS 119 602 profiles (e.g. Annex H for Pub-EAA and national EAA provider lists).

Wallet and RP side: Wallet Units verify RPs via the Registry and Access CA Trusted Lists (see Relying Party Registration & Access Certificates). PID Providers and Attestation Providers verify Wallet Providers against the Wallet Provider Trusted List before issuing credentials.

The PID Providers group defines validation semantics for PID/LPID/EBWOID; the QTSP group covers EAA and certificate aspects. Exact validation requirements (e.g. OIA_12, OIA_13, OIA_14) and consumption procedures are documented in the WP4 Trust Group trust-infrastructure schema and ETSI trusted lists implementation profile.

Establishing trust with a Credential Issuer

Wallet Units and Relying Parties establish trust in a Credential Issuer (PID Provider or Attestation Provider) and in the credentials they issue by combining **Trusted List** entries (trust anchors and, where applicable, authorised attestation types) with **registration** data (Registry or registration certificate). The catalogue of attestation schemes defines *what* credential types exist; Trusted Lists and registration define *who* is allowed to issue which types. The flow below is aligned with the WP4 Trust Group credential catalogue and issuer constraints: the verifier accepts a credential only if the issuer is present in the applicable Trusted List and the credential's attestation type is among the issuer's authorised types (ISSU_24, ISSU_24a, ISSU_34, ISSU_34a for Wallet Units; OIA_12, OIA_13, OIA_14, OIA_15 for Relying Parties). Go to Appendix C online for higher resolution.

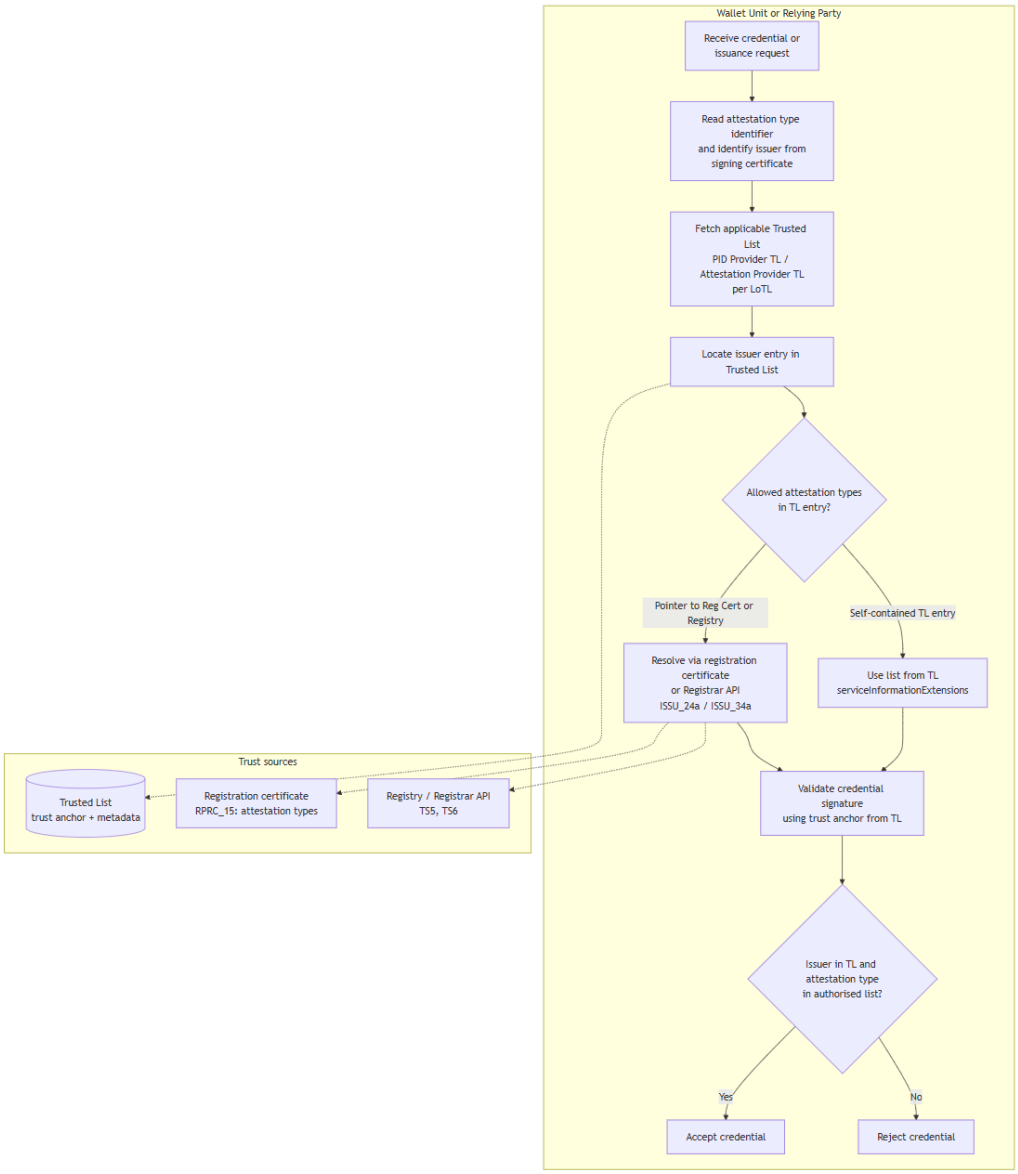

Figure 14: How trust is established between a Credential Issuer and Wallet Units / Relying Parties.

Establishing trust with a Wallet Solution

Trust in a **Wallet Solution** (Wallet Provider and Wallet Unit) is established by **Credential Issuers** (PID Providers and Attestation Providers) before issuing a PID or attestation. The flow is mandated by the ARF (Topic 9 Wallet Unit Attestation, Topic 31 notification and Trusted Lists), uses **OpenID4VCI 1.0** for the issuance protocol and issuer metadata (ISSU_22, ISSU_22a, ISSU_22b), and is reinforced by **OpenID4VC High Assurance Interoperability Profile (HAIP) 1.0** for authenticity, holder authentication, and certificate-bound tokens. The Wallet Unit presents a **Wallet Unit Attestation (WUA)** to the Issuer during the issuance request; the Issuer verifies the Wallet Provider against the **Wallet Provider Trusted List** and validates the WUA (ARF **ISSU_19**, **ISSU_21** for PID; **ISSU_28**, **ISSU_30** for Attestation Providers; **WUA_02**, **WUA_03**, **WUA_05**, **WUA_05a**). The Wallet Provider Trusted List is

compiled by the European Commission from Member State notifications (**WPNot_01**--**WPNot_05**).

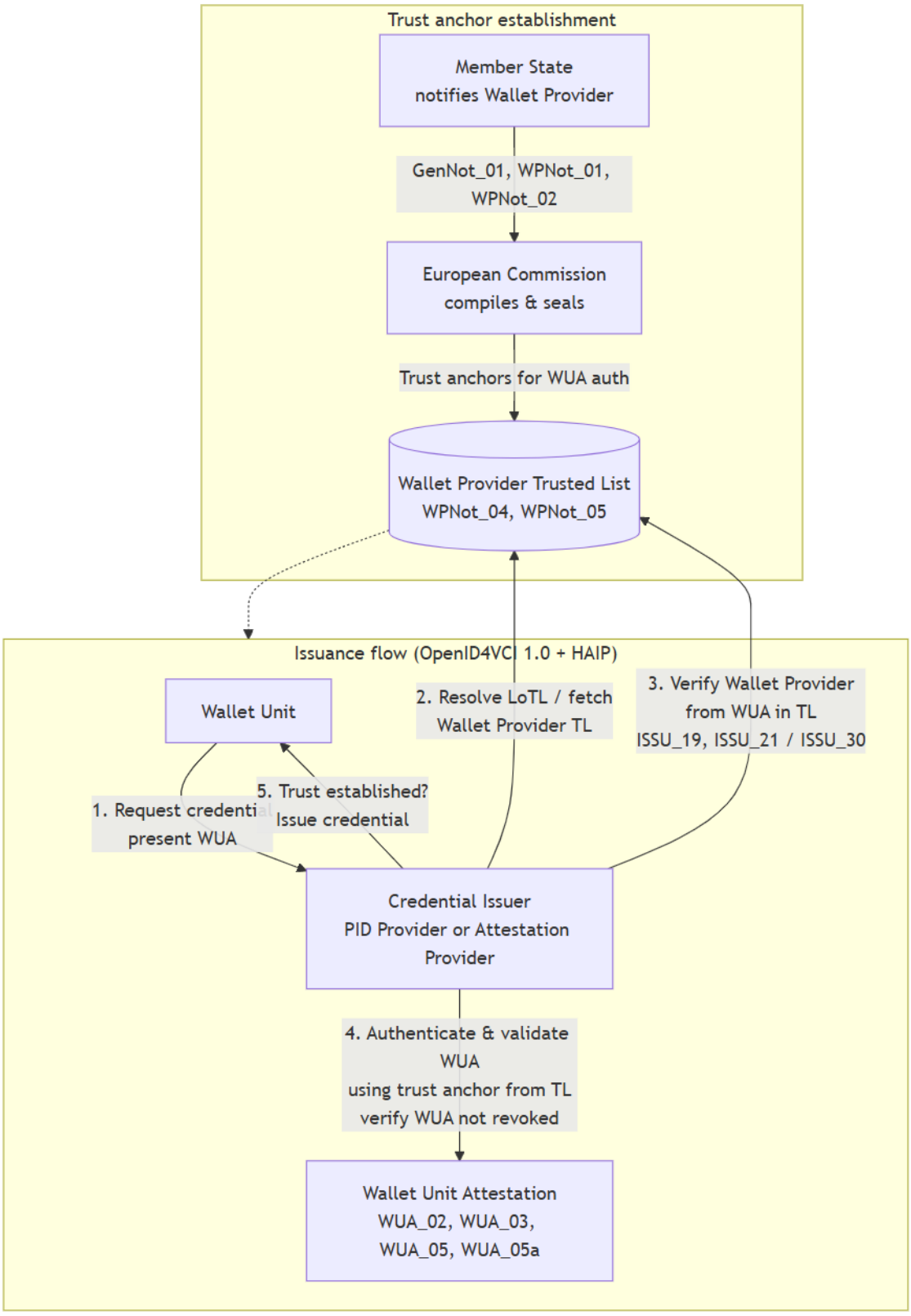

Figure 15: Trust established by a an issuer before issuing an attestation to a Wallet solution.

Governance Responsibilities

Responsibility	Owner (normative)	[MVP]	[MVP+]
Registration of PID Providers, Attestation Providers, Relying Parties	Member State Registrar	WE BUILD (mock Registrar)	MS Registrar or WE BUILD
Registry publication and common API (e.g. TS5, TS6)	Member State	WE BUILD (the WP4 reference registry or an other WP4 registry listed in LoTL)	Member State or WE BUILD
Access certificate issuance	Access Certificate Authority (notified by MS to Commission)	WE BUILD (Access CAs listed in LoTL)	Access CA (notified) or WE BUILD (Access CAs listed in LoTL)
Optional registration certificates	Provider of Registration Certificates (notified by MS to Commission)	WE BUILD (WP4 RegCert Providers listed in LoTL)	RegCert Provider (notified) or WE BUILD (WP4 RegCert Providers listed in LoTL)
Compilation and publication of Wallet Provider, PID Provider, Access CA, RegCert Provider Trusted Lists	European Commission	WE BUILD (reference TLs)	European Commission or WE BUILD
Compilation and publication of national EAA Provider TLs and MS QTSP Trusted Lists for QEAA	Member State Trusted List Provider	WE BUILD (mock MS TLP)	MS TLP or WE BUILD
List of Trusted Lists (LoTL), OJEU publication	European Commission	WE BUILD (reference LoTL)	European Commission or WE BUILD
Trust evaluation in Wallet Units and RPs (use of TL/LoTE and Registry per ARF)	Wallet / PID / Attestation / RP implementations	Wallet / PID / Attestation / RP implementations	Wallet / PID / Attestation / RP implementations

Trust and security responsibilities are split between **Member States**, the **European Commission**, and **participating entities** as follows:

Within WE BUILD, the consortium clarifies **which roles are assumed by MVP infrastructure** (e.g. mock or real Registrars, TLPs, Access CAs) and which are assumed to be provided by Member State or EU infrastructure. **Policy and conflicts** (e.g. credential-type or authorization collisions) are handled according to the dispute resolution and collision-prevention mechanisms described in the WP4 Trust Group authentication-authorization and policy framework.

Appendix D. Business Wallet Definition

Scope and context

This document sets out a non-technical working definition of "business wallet" as introduced in the European Business Wallet regulatory proposal, to support a common interpretation within WE BUILD and in dialogue with the European Commission. It is intended as reference material for the WE BUILD use case and capability work. It does not cover detailed architecture, protocol choices, implementation design, or use case roadmaps.

This document draws on the EUDI Wallet regulations, EWC deliverables^[1]^, and relevant industry and consortium publications, and incorporates the draft Implementing Act on Business Wallet.

Core concepts

Description

A **Business Wallet** is a product and service that enables an organisation to identify itself, manage authorisations, exchange verified attributes and documents, and receive legally relevant notifications in support of administrative and regulatory procedures. Unlike European Digital Identity Wallets, an European Business Wallet does not need to be an eID means under an eID scheme, although it may reuse similar components.

WE BUILD implementation note: *The topic of online business identity, potentially outside of eID schemes, needs to be further discussed within the WP4 Architecture group. It may also have consequences for the WP4 PID/LPID Providers group.*

A technical decomposition (front end, back end, and cryptographic components) is out of scope for this document.

Each Business Wallet has a single **wallet owner**, which is the entity that the wallet represents through its interactions. Note that this is distinct from, for example, the company owner or the wallet provider.

The wallet owner is defined by **European Business Wallet Owner Identification Data (EBW-OID)**, which includes an official name and an EU-unique identifier. These owner identification data are issued into the business wallet as an electronic attestation of attributes.

A Business Wallet can have multiple **wallet users**, meaning natural or legal persons that operate the wallet through a user interface or an application programming interface under roles and mandates set by the wallet owner. These wallet users may apply software applications to access these interfaces. Some users may be **authorised**

representatives, while others may be employees or service providers operating within delegated permissions.

Conceptual model

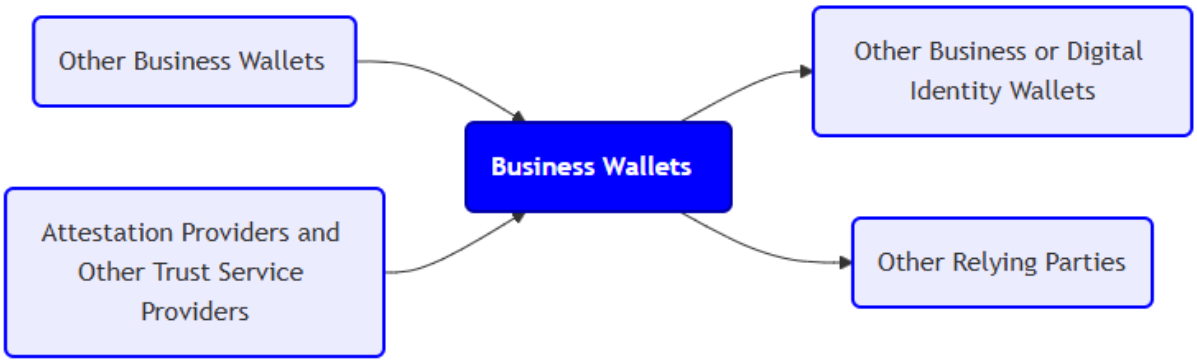

Figure 16: Conceptual Model of Business Wallets

Business Wallet definition

Roles supported

A business wallet enables its owner, amongst other operations, to act as:

- Issuer, holder or verifier of electronic attestations of attributes
- Signatory or origin of sealed data
- Sender or recipient of messages, such as submissions and notifications

These operations are under role-based access control, where recognised roles comprise:

- Wallet owner: the entity that is accountable for the legal consequences of the operation
- Authorised representative: a wallet user with an administrative mandate to act on behalf of the wallet owner, potentially with a limited scope or in limited contexts

In addition, the wallet owner may configure other roles that suit the owner's policies and/or national or EU law.

Other relevant roles are:

- Wallet provider: the entity that provides the business wallet solution to its owner (potentially the owner themselves)
- Owner identification data provider: the entity that verifies the identity of an authorised representative enrolling the wallet owner and attests, using an electronic attestation of attributes, the wallet owner's identification data in accordance with authentic source registrations

Key functions

Wallet lifecycle management

The business wallet enrols its owner via the electronic identification of an authorised representative and facilitates enrolment in connected trust services and directory services. The wallet provider is responsible for attesting to its validity to relying parties and enabling authorised representatives to revoke the business wallet and perform other lifecycle changes. In several cases, the wallet provider is also responsible for notifying authorised representatives and government authorities about lifecycle changes.

WE BUILD implementation note: *This will be the responsibility of the WP4 Wallet Providers group. At least several providers will be ready to manage their wallet solution and issue wallet units under new and changing business wallet requirements.*

Digital document management

The business wallet enables the wallet owner to create, store, use and validate various types of digital documents:

- Electronic attestations of attributes (EAAs, including QEAA, PuB-EAA, and EAA issued by the Commission)
- Business documents, such as electronic invoices
- Qualified certificates for electronic signatures and seals
- Qualified electronic signatures, seals and timestamps
- Evidence, such as provided by trust service providers upon electronic transactions, or by public sector bodies over the single digital gateway

For this purpose, the business wallet implements several applications, including signature creation and secure cryptographic applications.

WE BUILD implementation note: *The WP4 Wallet Providers provide, as part of their business wallet solutions, a subset of the functionalities required by the use cases. For the functionalities that require qualified trust services, such as the issuance of qualified certificates or the sealing of documents with qualified electronic seals, the WP4 QTSP group provides these services within WE BUILD. For reference, see the QTSP documentation.*

Secure communication channel

To enable public and private sector information exchange, such as in B2G eGovernment notifications, B2B/B2G eProcurement business documents and other business use cases, a business wallet implements a secure communication channel with other business wallets, with users of digital identity wallets, or with alternative solutions provided through a gateway. This channel enables cross-border delivery and receipt of

submissions and notifications with legal effect, and provides a trusted channel with public authorities and other regulated parties across the EU. The channel is implemented using a qualified electronic registered delivery service (QERDS). The digital address for the channel is registered in a standard digital directory.

WE BUILD implementation note: *the WP4 QTSP group will explore delivering an interoperable pre-production QERDS, along with CIR (EU) 2025/1944 requirements, as a service to the WP4 Wallet Providers group, working with the WP4 Architecture group on cross-cutting concerns, such as interoperability specifications. This enables wallet providers to provide a business wallet to the use cases with a digital address and access to the designated QERDS. For reference, see the QERDS documentation.*

Access control mechanism

To enable wallet owners, authorised representatives and other authorised users to access the business wallet while preventing unauthorised access, each business wallet implements role-based access control for the assets it protects, including digital documents and the secure communication channel. To identify, authenticate, and authorise wallet users, the access control mechanism relies on electronic identification means, such as digital identity wallets, and, potentially, on trust services for the electronic attestation of attributes.

WE BUILD implementation note: *The WP4 Architecture group, in collaboration with the WP4 Wallet Providers group, will explore the access-control mechanism for business-wallet solutions. This may rely on the EUDI wallets within WE BUILD or on other electronic identification means.*

Digital transaction management

Business wallets keep logs and provide dashboard user interfaces to enable control over transactions, including operations on the wallet lifecycle and on digital documents and messages sent and received over the secure communication channel. In addition, these logs enable dispute resolution regarding potentially unauthorised transactions, failures to meet reporting obligations, or administrative or procedural activities.

Appendix E. Appendix E. Wallet Implementation and Deployment Considerations in WE BUILD

This appendix provides a short overview of wallet implementation and deployment approaches observed among WE BUILD wallet providers. It does not repeat the architectural classifications defined in the ARF, but highlights aspects that are relevant for the WE BUILD pilots.

Different wallet implementations exist in the ecosystem, reflecting different user groups, device capabilities, and deployment environments. In practice, implementations often combine different approaches depending on the supported use cases.

Wallet Types Relevant for WE BUILD

From a deployment perspective, wallet implementations in the WE BUILD ecosystem can broadly be grouped into four practical categories.

Wallet Type	**Typical Deployment**	**Primary Use Cases**
Mobile (on-device)	Smartphone application using device hardware security	Natural person wallets and offline use cases
Web / browser-based	Browser interface with backend cryptographic services	Desktop services and enterprise workflows
Cloud / HSM-based	Server-hosted wallet infrastructure backed by HSMs	Legal person wallets and managed services
Hybrid	Combination of local device security and remote HSM	Mixed use cases requiring both scalability and offline capability

These categories reflect common deployment patterns observed across wallet implementations. The concrete architecture used by a wallet provider depends on the supported use cases, operational requirements, and device capabilities.

Deployment Patterns Observed Among WE BUILD Wallet Providers

The WE BUILD Wallet Provider Group conducted a stocktaking questionnaire covering **31 wallet providers** participating in the project. Providers described the deployment models they currently support.

The results show a clear split between natural person and enterprise wallet deployments.

Deployment Option	**Share of Providers**
Mobile wallet (iOS/Android app)	77%
Server wallet on cloud	55%
Server wallet on-premise	42%
Multi-device or white-label wallet	6%
Wallet functionality via API or SDK	6%

Many providers support multiple modes, typically combining a mobile wallet for natural persons, and a cloud or server-based wallet for legal persons.

Architectural Trends in the WE BUILD Ecosystem

The stocktaking exercise highlights several trends relevant for the WE BUILD pilots.

Mobile and cloud duality

The most common architecture combines:

- a **mobile wallet for natural persons**, and
- a **server-based wallet for enterprise or legal person scenarios**.

This reflects the broader EUDI ecosystem, where personal identity use cases are mobile-centric while organisational use cases often require backend infrastructure.

Increasing use of HSM-backed infrastructure

Several providers indicate the use of remote HSM infrastructure for enterprise wallet deployments. This approach supports large-scale operations and key recovery but requires continuous network connectivity.

Limited visibility of WSCD implementation choices

The questionnaire responses mainly describe the application layer (mobile app, server, or web wallet), rather than the underlying cryptographic architecture.

Only a small number of providers explicitly describe the type of secure cryptographic device used (for example secure hardware on the device or remote HSM infrastructure).

Emerging architectures for legal person wallets

Architectures supporting legal person wallets are still evolving.
Many providers indicate that their legal person wallet solutions will be further developed during the WE BUILD project in alignment with emerging European Business Wallet proposals.

As a result, the architectures described in the stocktaking responses should be understood as initial implementation approaches rather than final designs.

Appendix F. QTSP documentation

The WP4 QTSP group collaborates on internal reference code and documentation to increase interoperability. This appendix lists the entry points for this reference documentation by each provided service.

QES documentation

This is documentation of the WE BUILD: WP4 QTSP group.

Informative references

- Standards
  - Stable
    - CSC API v2.2.0.0
    - CSC DM v1.0.0
    - CSC_data-model-bindings v1.0.0
- Work items
  - eudi-doc-standards-and-technical-specifications#29: CSC standards updates
  - eudi-doc-standards-and-technical-specifications#68: TS 119 432 update
- Reference specifications
  - German EUDI blueprint: QES

QEAA documentation

This is part of the QTSP documentation.

Reference model

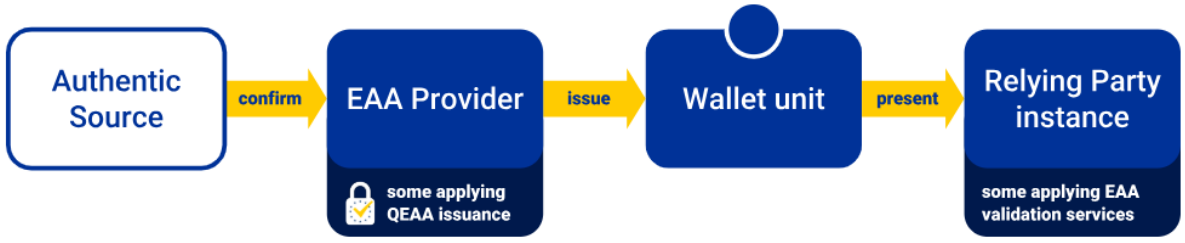

Figure 17: QEAA reference model

Architecture overview

- Architecture overview for QEAA in WE BUILD

Feature definitions

Below is a non-exhaustive overview of QEAA features that use cases may choose to pilot. For each feature in scope for the pilots, the QTSP group develops an interop profile and ensures available service compatibility.

- QEAA issuance to EUDIW
- EAA validation at RP
- Verification of attributes

Schemes for QEAA

Participants of the QTSP group may issue QEAA under any of the schemes referenced below.

- Hello World Attestation

Informative references

- Catalogues
  - Attestation Rulebooks Catalog: Catalogue of schemes for EAA in EUDI
- Standards
  - TS 119 471 v1.1.1: requirements for EAA Providers
  - TS 119 472-1: **DRAFT** Profiles for EAA - General requirements
  - TS 119 472-2: **DRAFT** Profiles for Relying Party Interface to EUDI Wallet
  - TS 119 472-3: **DRAFT** Protocol Profiles for interfacing to services providing Personal Identity Data and Electronic Attestation of Attributes
  - TS 119 612: Policy and security requirements for trust service providers issuing electronic attestation of attributes (EAA)
  - TS 119 602: Electronic signatures and infrastructures (ESI); Policy and security requirements for trust service providers issuing attribute attestations
  - ETSI TS 119 478: **DRAFT** Protocol Interface for Trust Service Provider use of Authentic Sources
- Technical reports
  - TR 119 476 v1.2.1: SD and ZKP for EAA analysis
  - TR 119 476-1 v1.3.1: SD and ZKP for EAA feasibility
  - TR 119 479-2 v1.1.1: EAA extended validation

QERDS documentation

This is documentation of the WE BUILD: WP4 QTSP group.

Scope: Provision of electronic registered delivery services

Reference model

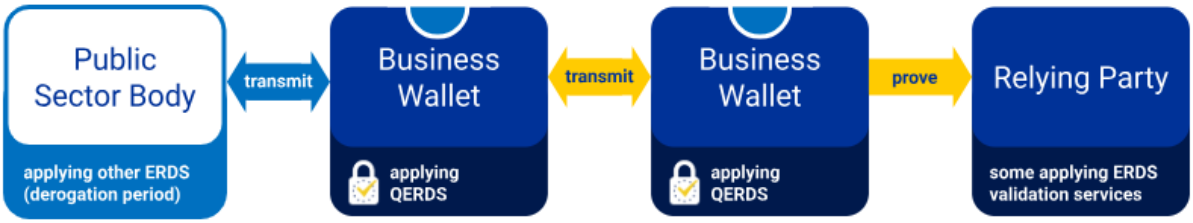

Figure 18: QERDS Reference model

Architecture overview

- Architecture overview for QERDS in WE BUILD

Technical reports

- QERDS interoperability framework requirements

Informative references

None yet.

rWSCD documentation

This is documentation of the WE BUILD: WP4 QTSP group. Scope:

- Management of remote wallet secure cryptographic devices

Informative references

None yet.

RPAC/RPRC documentation

This is documentation of the WE BUILD: WP4 QTSP group. Scope:

- Relying party access certificate issuance
- Relying party registration certificate issuance

Informative references

None yet.

Appendix G. Architecture Decision Records

The consortium maintains a lightweight architecture decision record (ADR) for each software-related decision affecting interoperability, as described in Chapter 7. At the time of submission, the following decisions had been made:

1. Publish consortium trusted lists
2. Baseline protocols
3. Specify PID and eAA formats
4. Provide EBWOID as a stable minimal basis
5. Wallet Unit Attestation and Lifecycle Management (For European Business Wallet)
6. Deliver business wallet data using QERDS
7. Attestation Revocation Mechanism

Appendix H. Conformance Specifications

The consortium maintains WE BUILD Conformance Specifications (WBCS) that define detailed technical requirements for interoperable implementations, as described in Chapter 7. At the time of submission, the following specifications had been established:

1. CS-001 - Credential Issuance - v1.0
2. CS-002 - Credential Presentation - v1.0