DC4EU project is Co-funded by the European Union's Digital Europe Programme
under Grant Agreement no. 101102611

D5.4 Tooling Evaluation Report

Revision: v.1.0

Work package	WP5
Task	T5.5
Due date	28/02/2025
Submission date	05/05/2025
Deliverable lead	GRNET
Version	1.0
Authors	Foteinos Mergoupis-Anagnou (GRNET)
Reviewers	Name Surname (Partner Y)

Abstract	One paragraph
Keywords	

Document Revision History

Version	Date	Description of change	List of contributors(s)
V1.0	01/03/2025	1st version of the deliverable for comments	Foteinos Mergoupis-Anagnou

V1.1	15/03/2025	2nd version of the deliverable for comments	Foteinos Mergoupis-Anagnou

DISCLAIMER

COPYRIGHT NOTICE

Project co-funded by the European Commission in the Digital Europe Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	Public, fully open, e.g. web	☐
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to DC4EU project and Commission Services	

** R: Document, report (excluding the periodic and final reports)*

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc.

EXECUTIVE SUMMARY

The D5.4 Tooling Evaluation Report provides an assessment of available tools and libraries for integrating the European Blockchain Services Infrastructure (EBSI) into the EUDI Wallet Reference Implementation (RI). The evaluation focuses on enabling EBSI-compliant credential issuance, verification, and secure data sharing. It analyses two primary solutions: the EBSI Hub Libraries and the WaltID SDK. The findings indicate that both solutions can be effectively utilized to achieve interoperability, albeit with necessary adjustments to streamline the integration process.

The report also outlines a proposed solution architecture, detailing how each component—the issuer, wallet, and verifier—will interact with the selected tooling. Despite some integration challenges, particularly with the WaltID SDK, the report concludes that the selected tooling provides a viable foundation for the Experimental EBSI Integration Prototype (D5.5), which will demonstrate end-to-end workflows compliant with EBSI standards.

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

TABLE OF CONTENTS

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

LIST OF FIGURES

ABBREVIATIONS

EBSI European Blockchain Services Infrastructure

DID Decentralized Identifier

IPC Inter-process communication

PKI Public-key infrastructure

RI Reference implementation

RPC Remote procedure call

VC Verifiable Credential

VP Verifiable Presentation

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

INTRODUCTION

OBJECTIVE OF THE REPORT

This deliverable (Tooling Evaluation Report) provides an analysis of the tools required to integrate EBSI capabilities into the EUDI Wallet Reference Implementation. It also summarizes the evaluation of the required tools, libraries, and frameworks to support EBSI within the EUDI Wallet Reference Implementation.

This report forms the basis for the subsequent development of an “Experimental EBSI Integration Prototype” (D5.5) that will demonstrate end-to-end issuance and verification workflows in compliance with EBSI standards.

SCOPE

The report focuses on evaluating the current available tooling and libraries to support the following functionalities:

- Requesting EBSI-compliant verifiable credentials from issuers with onboarded DIDs.
- Organizing acquired credentials into verifiable presentations.
- Enabling secure sharing and verification of credentials using EBSI-compliant processes.

The DC4EU project is Co-funded by the European Union’s Digital Europe Programme under Grant Agreement no. 101102611

1. SOLUTION OVERVIEW

Purpose of Task 5.5 is to modify the RI EUDI Wallet codebase [1] so that it becomes compatible with EBSI infrastructures. This means that the RI wallet should become capable of:

- requesting EBSI-compliant verifiable credentials (VCs) from issuers with EBSI-onboarded DIDs.
- organizing acquired credentials as an EBSI-compliant verifiable presentation (VP).
- sharing VPs with verifiers that have EBSI-onboarded DIDs.

To achieve this aim requires that the codebases of the RI EUDI issuer [2] and EUDI verifier [3] be modified as well, so that the standard VC lifecycle can be executed on top of the EBSI Trust Registry. This specifically means that

- the RI issuer should become capable of issuing EBSI VCs
- the RI verifier should become capable of verifying EBSI compliant VPs

Overall, the RI issuer, wallet and verifier components must be modified so that the following credential flow over the EBSI Trust Registry becomes possible:

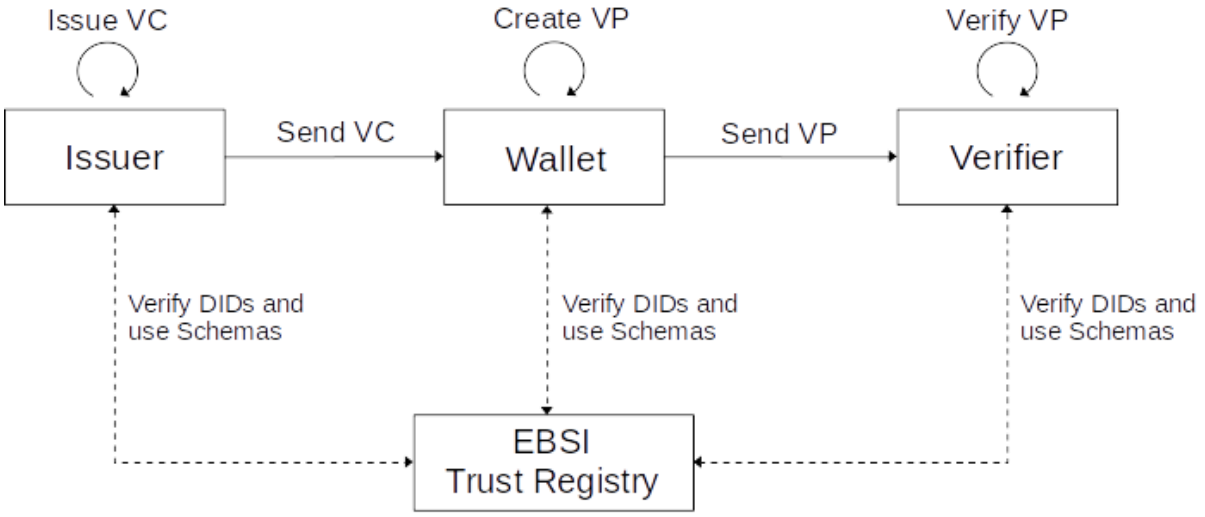

Figure 1 Abstract W3C VC flow over the EBSI

Enhancing the RI with EBSI capabilities entails the need to use dedicated libraries or SDKs that implement the core W3C VC functionality on top of the EBSI Trust Registry while also ensuring that exchanged information complies with its specified schemas and policies. The purpose of this report is to investigate the suitability and availability of relatively stable such libraries or SDKs and accordingly outline a more detailed architecture solution that considers this tooling with respect to the following constraints posed by the RI codebases themselves:

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

- The RI wallet [1] is an Android app written in Kotlin. Given that the application should ensure secure access to the user's private key, the ideal way for achieving EBSI capabilities (which includes signing verifiable presentations) is an android native local plugin (written in Java or Kotlin) that preserves the security properties of this access.

- The RI issuer [2] is a web application written in Python (Flask). Given the fact that no stable EBSI SDKs are available in this language (cf. Sections 2 and 3), the desired capabilities can be achieved through a dedicated service that can be deployed along with the issuer while also having secure access to the latter's PKI (this is because the service should be able to sign verifiable credentials). The issuer may interact with that service through any convenient inter-process communication mechanism, e.g., RPC or REST-API.

- The RI verifier [3] is a web application written in Typescript/JavaScript (AngularJS). Given that the official EBSI SDK (cf. Section 2.1.1) is written in this language, the desired capabilities should be ideally achieved by directly using its libraries.

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

2. AVAILABLE TOOLING

The work performed for this report concerns mainly desk research, identification and testing of individual solutions that can be used to implement the core W3C VC functionality on top of the EBSI Trust Registry. The work performed, contributed to better understand the ecosystem of EBSI library and tools that can be included in the arsenal of solutions that can be used in the development of the experimental prototype of the modified RI, demonstrating the EBSI-compatible issuance and verification workflows. The following sections presents only the main candidates for this implementation, excluding solutions or tools that did not passed the assessment criteria and requirements and have been rejected.

Below is a detailed and comprehensive overview of the features, capabilities, and benefits provided by the EBSI Hub Libraries [4] and the WaltID SDK [5].

2.1. EBSI HUB LIBRARIES

The EBSI Hub Libraries are a comprehensive suite of tools in Typescript designed to support developers in building blockchain-based applications within the EBSI ecosystem. These libraries are specifically tailored to enable the implementation of decentralized identity (DID), verifiable credentials (VCs), and other trust services, ensuring that applications are secure, interoperable, and compliant with European standards. They ensure that applications are compliant with European standards and interoperable with other systems.

Core features

The libraries provide robust support for Decentralized Identifiers (DIDs), which are a foundational component of decentralized identity systems. They include tools for creating, resolving, and managing DIDs using the **`did:ebsi`** method, which is specific to the EBSI ecosystem, as well as other methods such as the **`did:key`** method for natural persons. Using these libraries, developers can generate unique identifiers for users, organizations, or devices and anchor these DIDs on the EBSI blockchain, ensuring their immutability and verifiability. The libraries also enable the resolution of DIDs into DID documents, which contain essential metadata such as public keys, service endpoints, and verification methods. This ensures that DIDs can be easily verified and used across different systems, fostering interoperability and trust.

In addition to DID management, the EBSI Hub Libraries offer extensive functionality for working with Verifiable Credentials (VCs). These libraries enable developers to issue, sign, and verify W3C-compliant VCs in both JSON-LD and JWT formats. Verifiable Credentials are digital attestations that can represent a wide range of claims, such as educational qualifications, professional certifications, or identity proofs. The libraries provide tools for creating customizable credential templates, allowing developers to tailor VCs to specific use cases. They also include robust verification mechanisms to ensure the authenticity and integrity of VCs, including signature validation, expiration checks, and revocation status verification. To enhance user privacy, the libraries support advanced features like selective disclosure and zero-knowledge proofs, enabling users to share only the necessary information without revealing sensitive data.

Interoperability and Compliance

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

The EBSI Hub Libraries are designed with a strong emphasis on interoperability and compliance. They are fully compliant with W3C DID and VC standards, ensuring that applications built using these libraries can seamlessly interact with other identity systems and platforms. This interoperability is critical for enabling cross-border and cross-platform trust services, particularly within the European Union. The libraries also align with EBSI's trust framework, making them an ideal choice for developers building applications that need to comply with European regulatory requirements.

Ease of use

To simplify the development process, the EBSI Hub Libraries offer developer-friendly APIs that are clean, intuitive, and easy to integrate. The libraries are modular in design, allowing developers to include only the components they need, which reduces complexity and improves performance. Comprehensive documentation is provided, including detailed API references, step-by-step tutorials, and code examples. This documentation accelerates the onboarding process and helps developers quickly implement EBSI-compliant solutions. Additionally, the libraries are actively maintained and supported, ensuring that developers have access to the latest features and updates.

Security and Privacy

Security and privacy are central to the design of the EBSI Hub Libraries. They include robust cryptographic tools for key generation, signing, and encryption, supporting popular elliptic curve cryptography algorithms such as Ed25519 and secp256k1. These tools ensure that all operations, from DID creation to VC signing, are performed securely and efficiently. The libraries also leverage secure storage mechanisms to protect sensitive data, such as cryptographic keys and user credentials, ensuring compliance with best practices for mobile and web application security.

Use cases

The EBSI Hub Libraries are suitable for a wide range of use cases, making them a versatile tool for developers. They are ideal for building decentralized identity solutions, such as self-sovereign identity wallets that give users full control over their identity data. They also enable the creation of platforms for issuing and verifying Verifiable Credentials, such as digital diplomas, professional certifications, and identity proofs. Additionally, the libraries facilitate the integration of blockchain-based trust services into applications, leveraging the security and transparency of the EBSI blockchain to build innovative and reliable solutions.

Documentation and Developer Resources

The EBSI Hub Libraries are supported by extensive documentation and developer resources, making it easier for developers to get started and build applications efficiently. The documentation includes:

- **API References**: Detailed descriptions of all available APIs, including parameters, return values, and usage examples.
- **Tutorials**: Step-by-step guides for common tasks, such as creating DIDs, issuing VCs, and verifying credentials.

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

- **Code Examples**: Ready-to-use code snippets that demonstrate how to implement specific features and functionalities.

- **Best Practices**: Recommendations for secure and efficient development, ensuring that applications meet industry standards.

These resources are designed to accelerate the development process and help developers overcome challenges quickly. Additionally, the libraries are actively maintained, with regular updates and improvements based on developer feedback and evolving industry standards.

2.2. WALTID SDK

The Walt.id Android SDK is a powerful toolkit in Kotlin, designed to simplify the integration of decentralized identity (DID) and verifiable credential (VC) functionalities into Android applications. Developed by Walt.id, a leading provider of identity and wallet solutions, it empowers developers to build secure, privacy-preserving, and interoperable identity features into their apps with ease.

Core features

The Walt.id Android SDK provides a comprehensive suite of tools for managing Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs), enabling developers to implement advanced identity solutions in their Android applications.

- **Decentralized identity (DID) management**: The SDK supports the creation, resolution, and management of Decentralized Identifiers (DIDs) using popular DID methods such as **`did:key`**, **`did:web`**, and **`did:ebsi`**. Developers can generate DIDs for users, organizations, or devices, ensuring that each entity has a unique and verifiable identifier. The SDK also provides tools to resolve DIDs into DID documents, which include essential metadata such as public keys, service endpoints, and verification methods. This ensures that DIDs can be easily verified and used across different systems. Additionally, the SDK supports anchoring DIDs on blockchain networks, enhancing trust and verification by leveraging the immutability and transparency of blockchain technology.

- **Verifiable Credentials (VCs)**: The SDK enables the creation, signing, and verification of W3C-compliant Verifiable Credentials (VCs) in both JSON-LD and JWT formats. Developers can issue VCs for various use cases, such as digital diplomas, certifications, or identity proofs, and customize credential templates to meet specific requirements. The SDK also includes tools for verifying the authenticity and integrity of VCs, including signature validation, expiration checks, and revocation status verification. To enhance user privacy, the SDK supports advanced features like selective disclosure and zero-knowledge proofs, allowing users to share only the necessary information without revealing sensitive data.

- **Wallet functionality**: The SDK includes built-in wallet functionality for securely storing and managing DIDs, VCs, and cryptographic keys. It leverages Android's Keystore system for secure storage, ensuring that sensitive data is protected against unauthorized access. The wallet functionality enables the creation of self-sovereign identity wallets, giving users full control over their identity data and allowing them to

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

manage their credentials independently. This user-centric approach aligns with the principles of decentralized identity and enhances user trust and privacy.

Ease of integration

The Walt.id Android SDK is designed to be lightweight and modular, allowing developers to include only the components they need. This modular design reduces app size and improves performance, making the SDK suitable for resource-constrained environments. Built with a Kotlin-first approach, the SDK is optimized for modern Android development and provides a clean and intuitive API for seamless integration. Developers can easily incorporate the SDK into their existing projects and start using its features with minimal setup.

To further simplify the integration process, Walt.id provides comprehensive documentation, including detailed API references, step-by-step tutorials, and code examples. The documentation covers all aspects of the SDK, from basic setup to advanced features, ensuring that developers can quickly get up to speed and start building identity solutions. Additionally, the SDK is actively maintained and supported by the Walt.id team, ensuring that developers have access to the latest features and updates.

Security and Privacy

Security and privacy are at the core of the Walt.id Android SDK. The SDK includes robust cryptographic tools for key generation, signing, and encryption, supporting popular elliptic curve cryptography algorithms such as Ed25519 and secp256k1. These cryptographic tools ensure that all operations, from DID creation to VC signing, are performed securely and efficiently.

The SDK leverages Android's Keystore system for secure storage of cryptographic keys and sensitive data. The Keystore system provides hardware-backed security, ensuring that keys and data are protected against unauthorized access, even on compromised devices. This makes the SDK compliant with best practices for mobile app security and suitable for handling sensitive identity information.

To enhance user privacy, the SDK supports advanced features like selective disclosure and zero-knowledge proofs. Selective disclosure allows users to share only the necessary information from their credentials, while zero-knowledge proofs enable them to prove the validity of their credentials without revealing the underlying data. These privacy-preserving features ensure that users retain control over their identity data and can share it securely and selectively.

Interoperability

The Walt.id Android SDK is designed to be fully interoperable with other identity systems and standards. It is fully compliant with W3C DID and VC standards, ensuring that DIDs and VCs created using the SDK can be used across different platforms and ecosystems. This interoperability is critical for building identity solutions that work seamlessly in a global and interconnected world.

The SDK also supports integration with EBSI (European Blockchain Services Infrastructure), a leading blockchain-based identity ecosystem. This makes the SDK an excellent choice for developers building applications that need to comply with European standards and regulations.

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

Additionally, while the SDK is designed for Android, it can be used in conjunction with Walt.ID's other tools for cross-platform compatibility, enabling developers to build identity solutions that work across multiple platforms and devices.

Use cases

- **Decentralized Identity Wallets**: The SDK enables the creation of self-sovereign identity wallets, giving users full control over their identity data. These wallets can be used to store DIDs, VCs, and cryptographic keys, allowing users to manage their identities independently and securely.

- **Verifiable Credential Issuance and Verification**: The SDK is ideal for applications requiring the issuance and verification of VCs, such as digital diplomas, certifications, or identity proofs. It provides the tools needed to create, sign, and verify VCs, ensuring that credentials are authentic, tamper-proof, and privacy-preserving.

- **Blockchain integration**: The SDK facilitates the integration of blockchain-based identity solutions into Android apps. It supports anchoring DIDs on blockchain networks, enabling developers to build applications that leverage the security and transparency of blockchain technology.

Documentation and Developer Resources

The Walt.id Android SDK is supported by extensive documentation and developer resources, ensuring a smooth onboarding process and efficient development workflows. Key resources include:

- **API Reference**: Detailed documentation covering all SDK classes, methods, and parameters, with Kotlin-specific examples for clarity.

- **Tutorials and Guides**: Step-by-step tutorials for common tasks, such as DID creation, VC issuance, wallet integration, and cryptographic operations.

- **Code Samples**: Ready-to-use code snippets and sample projects on GitHub, demonstrating SDK integration and advanced use cases.

- **Versioning and Changelogs**: Clear documentation of SDK updates, backward compatibility, and migration guides for seamless transitions between versions.

- **Community Support**: Access to support channels such as GitHub Discussions, Slack, or Discord for troubleshooting and collaboration.

- **Best Practices**: Guidelines for secure key management, privacy-preserving design, and compliance with W3C/EBSI standards.

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

3. EVALUATION RESULTS

After extensively experimenting with their relevant APIs, we conclude that both the EBSI Hub Libraries and the WaltID Android SDK can be conveniently used for our purposes (Section 1). Specifically, the EBSI Hub Libraries are appropriate for enhancing the RI issuer and verifier, while the WaltID SDK provides a native way for extending the RI wallet.

- The EBSI Hub APIs are easier to integrate with, mainly because they are intended to be called from within JavaScript runtimes. We could relatively smoothly reproduce the specific credential flows of interest in a sandbox environment by appropriately adapting their documentation to our use case. Note that successfully reproducing these flows presupposes the onboarding of the issuer, holder and verifier DIDs to the EBSI registry. We observed a lack of straightforward documentation regarding the registration process, but this obstacle has been successfully overcome.

- The WaltID SDK is not as straightforward to integrate with, even though it provides an android native solution. This is mainly because (a) plugging its various modules into a Kotlin project is not well documented by means of sufficiently simple examples and guides in the context of Android Studio (b) configuring the EBSI DID resolver to hit the EBSI pilot network is not documented at all. The latter requirement is necessary for being able to successfully resolve DIDs under the hood and apply the correct schemas and policies. We further notice some minor incompatibilities (for example, no “exp” field is included in the VP token while being required by an EBSI verifier, etc.), which might be difficult to overcome; specifically, a VP token generated with the WaltID SDK is not guaranteed be accepted by a standard verifier, as implemented by the latest version of the EBSI Hub Libraries or currently deployed at the official EBSI services.

Based on the abovementioned findings, the proposed plan for the implementation of Task 5.5, is as follows: (a) Exhaust the possibilities of using the WaltID SDK for extending the RI Wallet by directly communicating with the WaltID core developers (b) In case this is impossible in a sufficiently short term, fallback to an android solution that employs the EBSI Hub Libraries for implementing the desired plugin.

The DC4EU project is Co-funded by the European Union’s Digital Europe Programme under Grant Agreement no. 101102611

4. SOLUTION ARCHITECTURE

In this section we elaborate on the architecture outlined in Section 1, i.e., we describe the proposed way to achieve EBSI interoperability by means of the tools presented in Section 2 and evaluated in Section 3. Specifically, Figure 1 is refined as follows:

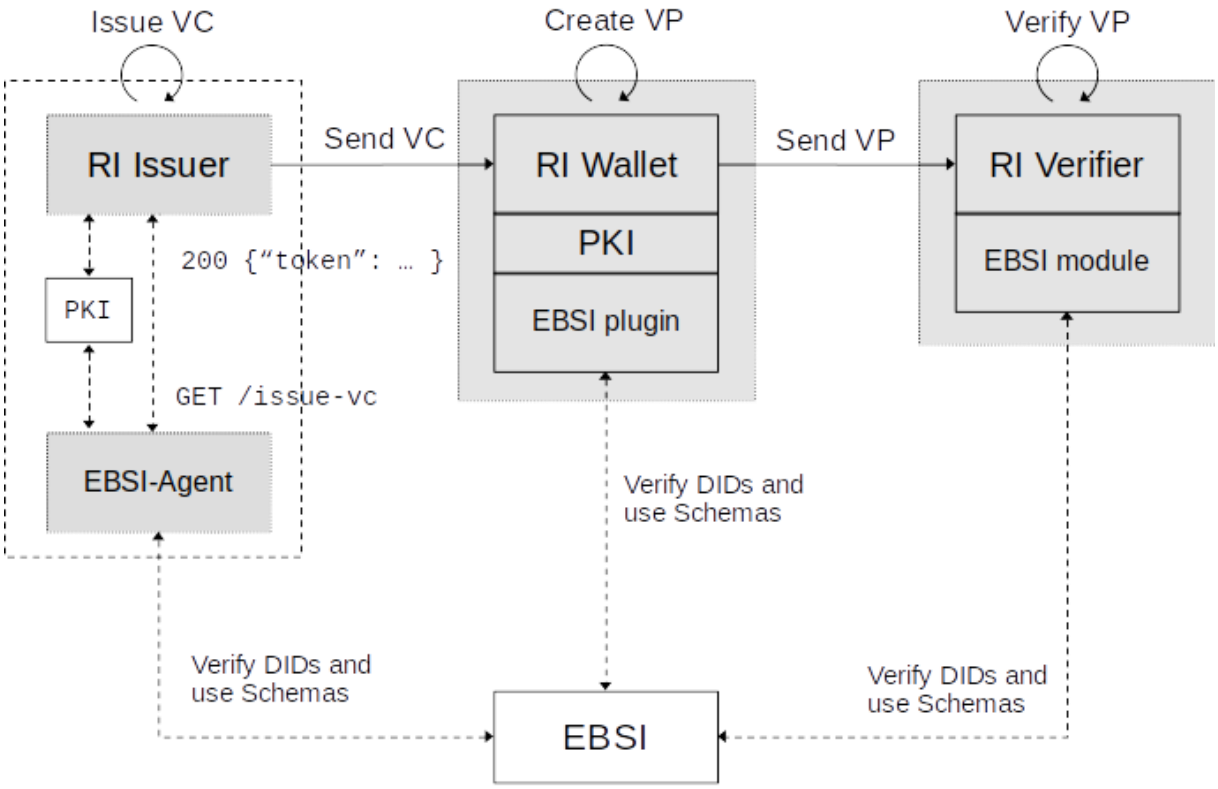

Figure 1 Concrete W3C VC flow over the EBSI

- **Issuer**: The RI Issuer obtains EBSI capabilities by becoming REST client to a service called the **EBSI-Agent**. The EBSI-Agent is a JavaScript microservice (Node.js) that is deployed along with the issuer. It exposes a REST API that makes the public interfaces of the EBSI Hub Libraries available to its clients, with each endpoint corresponding to a core VC operation on top of the EBSI Trust Registry. The main reason for choosing this extension solution is that the EBSI Hub Libraries are written in Typescript/JavaScript while the RI issuer runs in a Python runtime. Note that VC issuance presupposes secure access to the issuer's private key. We achieve this by having the issuer's PKI shared among the main issuer service and the EBSI-agent.

- **Wallet**: The RI Wallet obtains EBSI capabilities be means of an **EBSI plugin.** This plugin is a Kotlin module that will be natively integrated with the wallet codebase, making core VC operations available to the latter. Specifically, it should expose interfaces for verifying received credentials and creating verifiable presentations thereof, being implicitly responsible for the relevant DID resolutions be means on an appropriately configured EBSI DID resolver. The plugin should have direct access to

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

the user's PKI for creating signed VP tokens. The exact toolkit used for this purpose is specified per the evaluation results outlined in Section 3.

- **Verifier**: The RI Verifier obtains EBSI capabilities by means of an **EBSI module**. This is a JavaScript module that will be natively integrated with the verifier codebase (also written in JS) making VP verification available to the latter and being implicitly responsible for the relevant DID resolution. The module builds on top of the same library as the EBSI-agent, essentially a convenience wrapper of the official EBSI SDK (EBSI Hub Libraries).

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

REFERENCES

[1] https://github.com/eu-digital-identity-wallet/eudi-app-android-wallet-ui

[2] https://github.com/eu-digital-identity-wallet/eudi-srv-web-issuing-eudiw-py

[3] https://github.com/eu-digital-identity-wallet/eudi-web-verifier

[4] https://hub.ebsi.eu/tools/libraries

[5] https://docs.walt.id/community-stack/home

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611

APPENDIX A

Anything that is related but not core to the deliverable can go into the appendix.

The DC4EU project is Co-funded by the European Union's Digital Europe Programme under Grant Agreement no. 101102611