




Securing Digital Identities with Blockchain and Smart Contracts

Lucía Muñoz-Solanas , José Álvaro Fernández-Carrasco , Daniel Paredes-García 

Department of Cybersecurity
 Vicomtech, Basque Research and Technology Alliance (BRTA)
 Donostia/San Sebastián, Spain
 e-mail: {lmunoz | jafernandez | dparedes}@vicomtech.org

Abstract—The identity management model based on Self-Sovereign Identities, unlike classic models such as the centralized or federated model, allows users to have full control of their identity, without depending on external entities. A key element in a Self-Sovereign Identities-based system is the Verifiable Data Registry, where proofs and signatures of user credentials are securely recorded. This paper will present a Verifiable Data Registry that has been developed based on blockchain technology and implemented in an identity manager for a Data Space in the agri-food sector. In addition, the Smart Contracts developed to implement the necessary functionalities within the Self-Sovereign Identities context will be explained.

Keywords—Self-Sovereign Identities; Blockchain; Verifiable Data Registry; Smart Contracts.

I. INTRODUCTION

Self-Sovereign Identities (SSI) systems are revolutionizing digital identity management by giving individuals direct control over their credentials, enhancing privacy and security [1]. The agri-food sector, a vital part of the European economy, faces significant challenges in managing sensitive data. In response to these challenges, within the European project DIVINE [2], an SSI-based identity management system is being developed [3] for a Data Space tailored to the agri-food sector, enabling stakeholders to benefit from shared data.

This paper presents an SSI-based system for the agri-food sector, focusing on the Verifiable Data Registry (VDR) [4]. The VDR, developed on a private Ethereum blockchain, enhances trust and transparency by immutably recording all transactions and credential issuance [5]. The use of Smart Contracts (SCs) further automates and enforces credential and permission management rules, reducing human error and increasing operational efficiency.

The structure of this paper is as follows: Section II covers the components and functionalities of the SSI system. Section III examines the VDR configuration and SC customization. Section IV summarizes key insights and suggests future improvements.

II. BACKGROUND

SSI represents a modern approach for managing digital identities, granting individuals complete control over their personal information [1]. Unlike traditional systems reliant on centralized authorities, SSI allows users to own and manage their digital credentials directly. This model enhances privacy by storing data in personal digital wallets (digital applications for managing, storing, and presenting Verifiable Credentials (VCs) securely), rather than on vulnerable central servers,

thereby reducing the risk of breaches and unauthorized access. SSI also simplifies identity verification through cryptographic proofs, enabling secure presentation and validation of credentials.

Interactions within the SSI ecosystem rely on secure protocols and standards for trustless exchanges. VCs [4], which include metadata, claims, and cryptographic proofs, serve as digital equivalents to physical credentials. Metadata provides details about the credential, claims represent specific attributes, and cryptographic proofs ensure integrity and authenticity. Issued by trusted entities, VCs are securely stored and can be validated digitally.

The SSI ecosystem comprises three main actors and a VDR:

- **Holder:** The individual or entity that owns and controls their VCs, stored in a digital wallet.
- **Issuer:** The trusted entity that signs credentials, such as organizations or companies.
- **Verifier:** The party responsible for verifying the presented VCs. The verifier ensures that the VCs are properly signed by a trusted issuer and not revoked.
- **VDR:** A public or private ledger, functioning as a system or database, that stores public keys about issuers and other relevant data. The purpose of this ledger is to verify the authenticity of the VCs, holding the necessary information for verifiers to reliably assess their validity, without having to establish direct communication with the issuer, as illustrated in Figure 1. Often, blockchain technology is used as the VDR to ensure immutability and transparency.

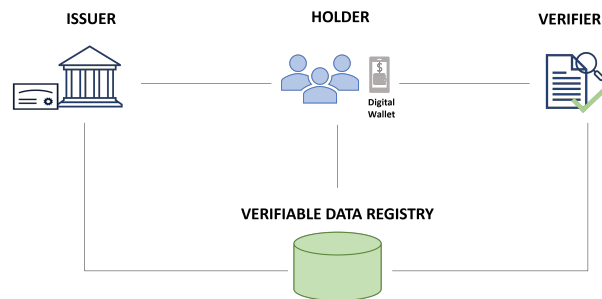


Figure 1. SSI Ecosystem.

This paper covers the development of a VDR based on blockchain technology for an SSI-based identity management system. Using a private Ethereum network with a Proof of Work (PoW) consensus, the VDR features distinct SCs for

different SSI actors to meet the DIVINE project’s needs. Key features of this solution will be outlined below.

III. FUNCTIONAL ANALYSIS

This section explores the **VDR** and its associated **SCs** designed to improve secure and efficient identity management. The VDR, based on the SSI framework, gives users control over their identity data, using blockchain technology for secure and immutable record-keeping.

The VDR operates on a private Ethereum blockchain consisting of three nodes. It utilizes PoW as its consensus mechanism, ensuring secure and immutable management of identity records. Ethereum was chosen for its robust SC capabilities and decentralized nature, while PoW provides network security and resistance to censorship [6].

SCs tailored to each participant in the SSI ecosystem - holders, issuers, and verifiers - have been created in Solidity [7] and implemented within the VDR to automate the management and validation of VCs. These self-executing programs enforce agreements based on predefined conditions, improving efficiency and security by eliminating intermediaries [8]. Developed as part of the DIVINE project, these SCs follow Ethereum’s ERC-735 and ERC-725 standards, enhancing SSI system functionality and security. ERC-735 [9] manages VCs on the blockchain and ERC-725 [10] governs key and permission management associated with these digital identities.

For the **holder’s** SC, based on ERC-735, the development includes:

- **Status Field Addition:** A status field in the Claim structure indicates if a claim is signed, pending, denied, or revoked, improving claim management.
- **Claim Editing Functions:** Functions allow holders to edit claim data and URI. Modifying data revokes the claim, requiring re-submission to the issuer for re-signing if valid.
- **Verifier Management Functions:** A verifier field in the Claim enables holders to manage who can access their claims and control verifier access.
- **Claim Overview Functions:** Functions for viewing all claims and retrieving claim IDs have been added.

The SC for the holder includes the following functions: *getClaimId*, *getClaim*, *getClaimIdsByType*, *addClaim*, *editData*, *editScheme*, *editUri*, *removeClaim*, *addVerifier*, *removeVerifier*, *generateClaimToSign*, *getClaims*, and *editStatusHolder*. The Claim structure for the holder’s SC contains the following fields: *topic*, *scheme*, *issuer*, *signature*, *data*, *uri*, *verifiers* and *status*.

For the **issuer’s** SC, based on ERC-734, the design includes:

- **Key Struct Revision:** The Key structure has been simplified to include *keyType* (e.g., Elliptic Curve Digital Signature Algorithm (ECDSA) or Rivest–Shamir–Adleman (RSA)) and *key* fields.
- **Claim Struct Introduction:** A new Claim structure with fields for *claimId*, *holder*, *signature*, *data*, *timestamp*, and *status* enhances claim management and tracking.

- **Function Enhancements:** New functions for managing and handling claims and keys have been added, including features for authenticating and verifying claims, retrieving specific claims, and updating claim statuses.

This SC includes the following functions: *getKey*, *signClaimToHolder*, *getClaim*, *getClaimSignature*, *addrToKey*, *addHolderClaim*, *getClaimList*, *unsignClaimToHolder*, *editClaimStatus*, *getClaimIssuer*, and *removeClaimIssuer*. The issuer uses the *getClaim* function to view the holder’s claim. If the data provided in the topic field is valid, the issuer will sign the claim. This will modify the *signature* field in the claim, adding a cryptographic proof that contains information of the holder, the data and the topic of the claim.

A specialized SC fulfills the role of **verifier**, including the following features and functionalities:

- **Validate Claims:** Functions to verify if a claim’s topic aligns with the holder’s context.
- **Issuer Managements:** Functions for allowing issuers to sign claims on specific topics, enhancing verification and topic integrity.

For the verifier’s SC, the design includes the following functionalities: *checkClaimPurposes*, *checkClaimByPurpose*, *claimToSign*, *addTopicToIssuer*, *removeTopicFromIssuer*, *getSignatureAddress*, and *checkPurposesByIssuer*. Once the holder’s claim has been signed, the verifier will check the *signature* field in the claim, which contains a cryptographic proof of the *issuer* who signed it. The verifier will then verify that the *status* field in the claim is set to “approved” and that one of the entries in the *verifiers* field corresponds to the verifier performing the check. These enhancements collectively strengthen the system for managing digital identities and claims.

IV. CONCLUSION AND FUTURE WORK

In this work, a VDR for an SSI-based identity management system based on blockchain technology has been developed, which improves security and immutability. As part of the DIVINE project, participants in the identity management system - known as holders - obtain digital identities through VC. These participants register their applications, designed for the agri-food sector, on the platform, with the applications functioning as issuers. The credentials will represent roles within specific applications, and the issuer will sign this credentials. An identity provider will act as the verifier, ensuring the validity of the VCs in each request.

Future improvements will include migrating the credential format to align with the European Blockchain Services Infrastructure (EBSI) [11] for regulatory compliance and better interoperability. Additionally, transitioning from PoW to Proof of Stake (PoS) will be explored to enhance system efficiency and sustainability.

ACKNOWLEDGMENT

This work has been partially supported by the European Union’s Horizon 2020 Research and Innovation Program under the project DIVINE (Grant Agreement No. 101060884).

REFERENCES

- [1] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, “A survey on essential components of a self-sovereign identity,” *Computer Science Review*, vol. 30, pp. 80–86, 2018.
- [2] DIVINE Project Consortium, “Divine - demonstrating value of agri data sharing for boosting data economy in agriculture,” 2024, [Online]. Available: <https://divine-project.eu/>.
- [3] J. Á. Fernández-Carrasco, L. Muñoz-Solanas, L. Seguro-Gil, and D. Paredes-García, “Credssi: Enhancing security and privacy with self-sovereign identities approach,” in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, IEEE, 2024, pp. 745–750.
- [4] M. Sporny, D. Longley, D. Chadwick, and O. Steele, “Verifiable credentials data model v2.0,” Published by World Wide Web Consortium (W3C), 2024, [Online]. Available: <https://www.w3.org/TR/vc-data-model-2.0/>.
- [5] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, “In search of self-sovereign identity leveraging blockchain technology,” *IEEE access*, vol. 7, pp. 103 059–103 079, 2019.
- [6] S. Chandler, “Proof of Stake vs Proof of Work: Key Differences,” Published by Business Insider, 2024, [Online]. Available: <https://www.businessinsider.com/personal-finance/investing/proof-of-stake-vs-proof-of-work#:~:text=The%20main%20strength%20of%20proof,of%20energy%20has%20become%20controversial.>
- [7] S. Team, “Solidity documentation,” Published by Solidity, 2023, [Online]. Available: <https://docs.soliditylang.org/en/v0.8.28/>.
- [8] W. Zou *et al.*, “Smart contract development: Challenges and opportunities,” *IEEE transactions on software engineering*, vol. 47, no. 10, pp. 2084–2106, 2019.
- [9] Ethereum Improvement Proposals, “EIP-735: Self-sovereign identity,” 2018, [Online]. Available: <https://github.com/ethereum/EIPs/issues/735>.
- [10] Ethereum Improvement Proposals, “EIP-725: Ethereum identity,” 2018, [Online]. Available: <https://github.com/ethereum/EIPs/issues/734>.
- [11] European Commission, “European blockchain services infrastructure (EBSI),” Published by European Commission, 2024, [Online]. Available: <https://ec.europa.eu/digital-building-blocks/sites/display/EBSI/Home>.