IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 July 2023, accepted 21 July 2023, date of publication 26 July 2023, date of current version 1 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299047

RESEARCH ARTICLE

Orchestrating Digital Wallets for On- and Off-Chain Decentralized Identity Management

VID KERSIC, (Member, IEEE), URBAN VIDOVIC, ANDRAZ VRECKO, MARTIN DOMAJNKO, AND MUHAMED TURKANOVIC, (Member, IEEE)
Faculty of Electrical Engineering and Computer Science, University of Maribor, 2000 Maribor, Slovenia

Corresponding author: Vid Kersic (vid.kersic@um.si)

This work was supported by the Slovenian Research Agency (Research Core Funding) under Grant P2-0057.

ABSTRACT Digital identity is becoming one of the core elements during the digitalization age, when more and more processes and interactions are taking place in the digital sphere. Therefore, current identity management approaches will define how these interactions will look in the future, but different fields and communities often approach management with their own solutions and tools, despite their similarities. This includes decentralized digital identities, where the identity is managed with asymmetric cryptographic keys, and no centralized entity oversees the whole identity system. This paper focuses on managing on- and off-chain decentralized digital identities, with the former being used for blockchain networks and the latter for self-sovereignty and privacy. While both types of decentralized identity build on the same cryptographic and identity primitives, there is no single wallet that handles both. Therefore, this paper proposes an orchestration solution for both wallet types, which enables their convergence to a single universal wallet and validates it with a real-life decentralized identity use case.

INDEX TERMS Decentralized identity, self-sovereign identity, wallet, identity, identity management, web3.

ABBREVIATIONS

DLT	Distributed Ledger Technology.
DSL	Domain Specific Language.
SSI	Self-Sovereign Identity.
NFT	Non-Fungible Token.
DAO	Decentralized Autonomous Organization.
dApp	Decentralized Application.
IIW	Internet Identity Workshop.
DID	Decentralized Identifier.
VC	Verifiable Credential.
VP	Verifiable Presentation.
IoT	Internet of Things.
SLR	Systematic Literature Review.
dPKI	Decentralized/distributed Public Key Infrastructure.
EOA	Externally Owned Account.
W3C	World Wide Web Consortium.
DIF	Decentralized Identity Foundation.
URI	Uniform Resource Identifier.
JSON	JavaScript Object Notation.
JSON	LD - JavaScript Object Notation for Linking Data.
JWT	JSON Web Token.
EIP	Ethereum Improvement Proposal.
SD	JWT - Selective Disclosure for JWT.
ZKP	Zero-Knowledge Proof.
OIDC	OpenID Connect.
OIDC4VC	OpenID Connect for Verifiable Credentials.
OIDC4VP	OpenID Connect for Verifiable Presentations.
EU	European Union.
EUDIW	European Union Digital Identity Wallet.
BIP	Bitcoin Improvement Proposal.
EBSI	European Blockchain Services Infrastructure.
TIR	Trusted Issuers Registry.
VDR	Verifiable Data Registry.
EVM	Ethereum Virtual Machine.
IPFS	InterPlanetary File System.
PoC	Proof of Concept.
ECDSA	Elliptic Curve Digital Signature Algorithm.
HSM	Hardware Security Module.

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

I. INTRODUCTION

Digital identity is one of the most essential and core elements of the Internet, as it establishes trust for digital processes, e.g., remote identification, authentication, and authorization. Furthermore, these processes enable different parties to exchange data or use services digitally under the trust umbrella. Throughout the decades, multiple approaches to digital identity management have been proposed, with each optimizing different characteristics, such as ease of use or self-sovereignty.

Nowadays, we can differentiate between the following core digital identity models: centralized, federated, and decentralized [1]. The latter is the newest among the models, which came into focus after the introduction of blockchain technology. The idea of the decentralized identity model is that end-users create their own digital identities and corresponding identifiers and use them for any service or digital interactions they like [2], [3]. However, the decentralized identity idea grew into multiple forms, like those for using blockchain and DLT-based networks, and also self-sovereign identities (SSI) [1], [3]. We can also group these into on-chain (e.g., blockchain-based) and off-chain (e.g., SSI) decentralized identities. Each group is trying to solve specific challenges and address some requirements for different use cases. At the same time, they also build their concepts and technological ecosystems on interoperability and standardized approaches.

There are multiple benefits of each of the decentralized identity models, while the main drawback of both is that each of them is building on their own continuously and aggressively, and often without the idea of converging their approaches. Nevertheless, this discrepancy burdens the end-users, who need to manage multiple on-chain and off-chain digital identities with various digital wallets. This paper focuses on the problem and proposes a solution to link both on and off-chain decentralized identity models while providing the end-users with the best from both worlds.

A. AIM AND CONTRIBUTION

Considering how much user-centric approaches are essential today, the research aimed to analyze in detail both decentralized identity models, i.e., on-chain (blockchain-based) and off-chain (SSI), and address the conceptual and technical gap between both models as much as possible, to ease the usage of both for the end users. Furthermore, the goal was not only to bridge the gap between the models, but also to come to a solution enabling the usage of all the positive features these models provide. As such, the core idea was that we enable the end-users the creation of multiple decentralized identifiers, both on-chain, as well as off-chain, and provide them with the features of the SSI concept, where additional identity data in the form of attestation is managed with the VC/VP data model, and where the user has the complete control to present his identity-related data to the verifier and collect them from the issuer. Furthermore, the end-user should be able to execute all relevant on-chain related tasks as he can now, e.g., collect NFTs and send crypto tokens. The core beneficiaries of our provided solution presented in this paper are all end-users who nowadays use any form of decentralized identity. Furthermore, with our provided solution, we ease the usage of all decentralized technologies, and thus open the gate for additional end-user onboarding. We emphasize that, while decentralized (off-chain) ecosystems include three main actors (i.e., holder, issues, verifier), in this research paper, we focus only on the holder's perspective and the user-centric decentralized digital identities, and not on machine-centric (e.g., IoT-based) etc.

B. METHODOLOGY

Several complementary research methods were employed to fulfill the aim presented above. We first executed a literature review on decentralized identity models, digital wallets, and related concepts. The review enabled the understanding of the current state of the field, identifying gaps, and building a foundation for the proposed solution. We further analyzed and compared different identified models, specifically on-chain (blockchain-based) and off-chain (SSI). This analysis involved studying the characteristics, benefits, and challenges of each model, as well as their respective building blocks and components. Based on the analysis and comparison, we designed a solution to bridge the gap between on-chain and off-chain decentralized identity models. The solution aimed to provide end-users with the best features of both models and enable seamless integration and interoperability between them. The design of the solution involved defining software architecture, components, and interfaces required for orchestration. The proposed solution was implemented and validated on a real-world decentralized identity use case. We used the Web3 wallet MetaMask as the foundation for the solution and performed validation by addressing a plutocracy problem in decentralized autonomous organizations (DAOs). The functionalities and features provided by the on- and off-chain identity models, as well as the orchestration achieved through the proposed solution, were compared and evaluated. This evaluation helped assess how the solution satisfied decentralized identity requirements and addressed the challenges identified in the literature review. Security threat analysis, using the STRIDE modeling framework, was conducted to identify possible attack scenarios and assess the solution's security vulnerabilities.

The rest of the paper is structured as follows. Section II presents the Related works in this field. We then provide the reader with some preliminaries (Section III) on the topic of decentralized identities and digital wallets. Section IV outlines the proposed model, including a solution which addresses the aimed contributions. The validation of the proposed model and solution, based on a real-world use case, is presented in Section V, including a security threat analysis. Lastly, we discuss and conclude the paper with some future work direction in Sections VI and VII.

II. RELATED WORKS

In general, there has been much research in the past several decades on digital identity and means of authenticating users in different systems. However, the focus on digital identity wallets started in the last few years, due to the advancements in decentralized systems and global awareness of data privacy and user-controlled data. This section provides an overview of the current work on digital identity wallets based on decentralized identity systems. While there is some recent research on this topic, many solutions are presented in the gray literature, or other formats freely available on the Internet.

As the core element of decentralized identity management is the self-custodial-based digital wallet, we analyzed possible approaches to address the challenges presented in the Introduction. Furthermore, we focused on such a literature review since our proposed solution is a new digital wallet model and concrete wallet solution.

In [4], the authors performed a Systematic Literature Review (SLR) on digital identity wallets focusing on users' private data. Their results confirm the need for digital identity wallets, with the main reasons being having secure storage for cryptographic keys, ways to manage identifiers and data, and others. The paper also concludes that self-sovereign approaches, such as SSI, have substantial advantages over centralized ones, such as centralized identity providers or federated identity management systems. Several papers also analyzed and compared digital identity wallets based on SSI and blockchain technology [5], [6]. The authors in [7] proposed a framework for password-less interaction with web applications, with user authentication being based on SSI. Paper [8] proposed an architecture to achieve a high level of assurance identity with a mobile phone-based wallet following the SSI principles.

Solutions have also been proposed for using blockchain technology as a decentralized/distributed public key infrastructure (dPKI), without concern about user identity data. Several papers have introduced new architectures for digital wallets for different blockchain networks, such as Bitcoin [9]. The authors in [10] proposed an e-wallet architecture for banks and financial institutions based on distributed ledger technology (DLT). In [11], the authors created a comprehensive review of all cryptocurrency wallets on the market, focusing on different approaches to cryptographic key management, wallet recovery methods, etc. Karantias [12] presented a taxonomy for the cryptocurrency wallets, and [13] conducted a security analysis of different wallets concerning the safety of cryptographic keys.

There is also some existing work and research on having a universal wallet supporting on- and off-chain identities. However, there was no actual implementation and validation of any use cases or examples. The authors in [14] performed the taxonomy of digital wallets, and used the term universal wallet to describe a wallet that can be used for different types of decentralized identity. They did a high-level overview of which use cases the wallet should support, such as cryptocurrency, NFTs, and digital identities, and what core functionalities should be handled by the wallet, interaction with external storage, encryption, digital signing, etc. There is also an ongoing effort by The OpenWallet Foundation[1] to define the wallet and its interfaces, gaining quite a lot of interest from low, mid, and large-sized companies.

III. DECENTRALIZED IDENTITY ECOSYSTEM

This Section presents and explains the necessary preliminaries used throughout the paper.

A. DECENTRALIZED IDENTITY

Decentralized identity is a movement in the digital identity field that emphasizes the self-custodial and self-sovereign management approach of digital identities, representing identifiers and other identity-related data management and control without the need for centralized entities. This is enabled by cryptography, where users hold cryptographic keys in respective digital identity wallets. This decentralized identity model and principles are becoming adopted increasingly in Web3. Many advancements for decentralized identity come from the Web3 space, where the whole authentication mechanism relies on cryptographic keys, since users sign transactions digitally to perform actions on their financial assets.

Since different parties and entities tackled the decentralized identity from different angles and solved various use cases, several models have evolved in the last decade. Alongside the development of distributed and decentralized technologies, such as DLTs and blockchain, an identity meant purely for blockchain developed in different forms. Apart from that, working groups and researchers focused on upgrading existing systems, focusing on user authentication and private data, and the field of SSI was born. However, both models have several common characteristics and can even be used together.

1) ON-CHAIN IDENTITY

The idea of so-called blockchain-based identities (i.e., on-chain) was primarily to enable end-users to be digitally pseudonymous and execute core blockchain-related tasks, i.e., transaction execution. It relies on public key cryptography, where user accounts or identifiers are blockchain addresses derived from public keys (derived from private keys). Thus, the whole identity is controlled with a single private key and managed through digital (crypto) wallets. Nowadays, we know that transactions can be more complex, and used for various other use cases than just cryptocurrency exchange, i.e., buying or swapping tokens, trading non-fungible tokens (NFTs), voting on DAO proposals, etc. These use cases have grown so much in the last few years that a new concept called Web3 was introduced. Web3 can be summarized as any online application, specifically, decentralized applications (dApps), enabling end-users and their blockchain-based digital identities to interact with the aforementioned use cases [15], [16]. The

[1] https://openwallet.foundation/

interaction and management of the decentralized digital identifiers are performed with so-called digital (crypto) wallets (e.g., MetaMask). As such, we also categorize this approach as non-custodial or self-custodial, since users control their digital identities by holding cryptographic keys in their wallets. In the on-chain identity, digital signatures are verified by nodes running the blockchain ledger; thus, the nodes must verify that the users are, in fact, in control of the identity, and are allowed to change the blockchain state. Blockchain identity optimizes the availability of the data related to the identity, since the data are always available for anyone on the blockchain; hence called on-chain. It works well for public and financial data, but is unsuitable for privacy.

2) OFF-CHAIN IDENTITY

Off-chain, or self-sovereign identity (SSI), defines a new way to manage identity and data, with the most significant focus on user-controlled data and privacy.

While the on-chain approach evolved in the blockchain community, the off-chain is rooted in SSI circles and the Internet Identity Workshop[2] (IIW). This decentralized identity model is not coupled tightly with the blockchain itself. Two main building blocks comprise SSI as a whole: Decentralized Identifiers (DIDs) [17] and Verifiable Credentials (VCs) [18]. Both standards are recommendations by the World Wide Consortium (W3C),[3] and are defined by W3C and the Decentralized Identity Foundation (DIF).[4] DIDs are based on standardized globally unique persistent identifiers, similar to Uniform Resource Identifiers (URI). They serve as unique identifiers for the users or other entities (e.g., enterprises, IOT devices). An example of DID is *did:ebsi:zvHWX359A3CvfJnCYaAiAd*, where ebsi is the name of the DID method, and the last part is a unique identifier. DID methods describe the characteristics of DIDs, such as how to get to the underlying metadata of the identifier, which is contained in the DID document. A DID document is a JSON-LD document containing all the available metadata related to the DID, such as its public cryptographic keys, service endpoint, authentication, and authorization cryptographic keys. Translation between DID identifiers and DID documents happens through DID Resolvers, which fetch the data from the DID repository, e.g., a smart contract on the blockchain, or perform conversion directly from the identifier to the DID document. Several DID methods exist, such as *did:key*, *did:ethr*, *did:pkh*, and *did:ebsi*, each focusing on and optimizing different characteristics, while also some are based on-chain and some off-chain.

One of the core ideas of the SSI concept is also the related identity data, which is represented with VCs, digitally signed off-chain attestations issued by entities that usually have some reputation. Any form of attestation, such as passports, achievements, and certificates, can be structured as VCs. VC is a unique digital data representation specification often expressed in JSON-LD format. The most common applications for VC are attestations like passports, documents, and driving licenses. All VCs contain the data and metadata describing the process and conditions in which the data were issued, such as issuance date, expiration date, the issuer's DID, the holder's DID, type of credential, etc. Digital signatures can be applied differently to VCs, such as JSON Linked Data Proofs, JWT, and EIP-712.

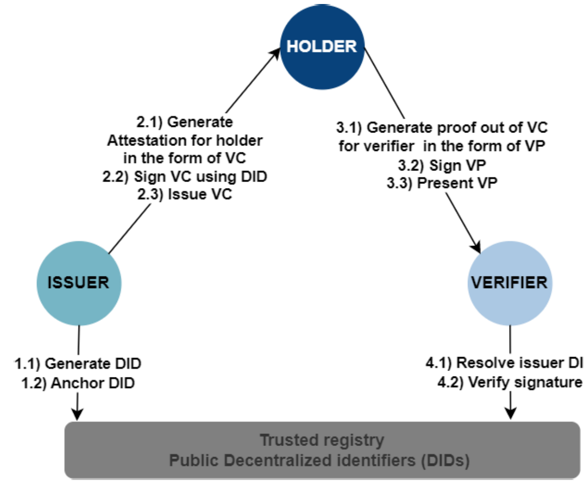


FIGURE 1. SSI trust model [3].

Users share VCs with other entities by creating Verifiable Presentations (VPs), which contain data from one or more VCs. VPs are signed digitally by the user's DID and corresponding cryptographic keys. For privacy reasons, several cryptographic techniques and methods were presented to enable the selective disclosure of data. This ranges from a special type of JWTs, called SD-JWT, or advanced use of cryptography, such as BBS+ [19], [20]. There is also a growing interest in applying Zero Knowledge Proofs (ZKPs) technology to VCs and VPs, enabling the proving of predicates instead of disclosing actual user data values [21], [22].

Trust assumption in SSI is based on the three actors model: issuer, holder, and verifier (Figure 1). Issuers are trusted entities that issue data to the holders via VCs, e.g., government institutions and universities. Holders are users to whom the VCs are bounded. Verifiers are entities to which holders identify and authenticate, and provide proof for specific claims by sending them digitally signed VPs. Both issuers and holders must be identified by their corresponding DIDs. Since the metadata for DIDs is always available online (or resolvable directly through DID identifiers), there is no need for a centralized authority when verifying the authenticity of data through digital signatures. The data and information exchange between all actors is executed based on the soon-to-be standardized protocols, such as OpenID Connect (OIDC)[5] and DIDComm.[6]

[2] https://internetidentityworkshop.com/
[3] https://www.w3.org/
[4] https://identity.foundation/
[5] https://openid.net/specs/openid-connect-4-verifiable-credential-issuance-1_0-05.html
[6] https://identity.foundation/didcomm-messaging/spec/

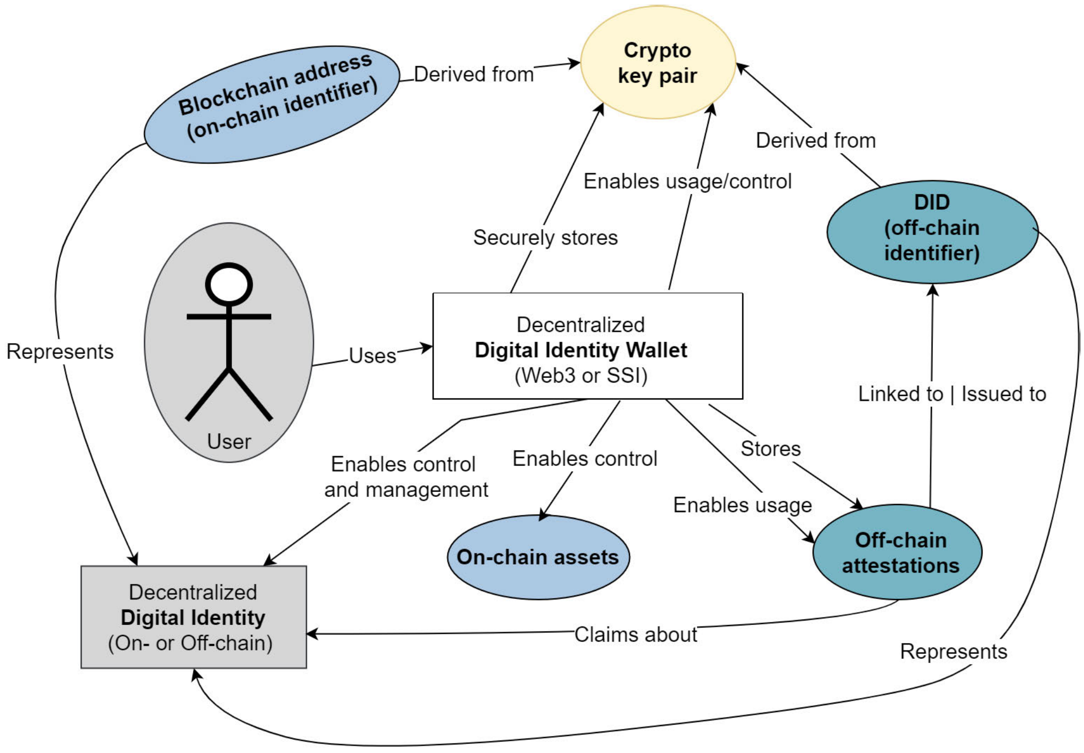


FIGURE 2. Relations between a user, his digital identity, identifier and a digital wallet.

SSI can also be referred to as the off-chain identity model. While, in some DID methods, the metadata of DIDs is managed on-chain, e.g., on the Ethereum blockchain through DID documents, all other related identity data, which are in the form of VCs, are, by default, always off-chain, in the user's wallets, or some other place the user decides. This type of identity is, thus, more suitable for use cases where privacy and security play a significant role. The identity model also satisfies the GDPR requirements of the EU.

B. DIGITAL WALLETS

Digital wallets are software applications allowing users to manage their digital identifiers and related assets securely. There are several implementations of digital wallets, e.g., mobile, desktop, browser, cloud, or hardware-based, whereby each of these implementations could be for multiple types of digital wallets, like the e-banking wallets, EUDIW (European Digital Identity Wallet), Web3 (crypto) wallets and SSI wallets [4], [14]. In this paper, we focus on digital wallets where digital identities are based on cryptography, i.e., public-key cryptography, and users store their identity keys in their wallets. Thus, the proposed approach relates to software-based Web3 and SSI wallets but also works with hardware wallets, which work in tandem with software wallets. Both wallet types allow users to control their assets and data directly, eliminating the need for intermediaries, and thus making them more secure and private than traditional centralized wallets (e.g., e-banking wallets). Figure 2 illustrates in more detail the relations between a digital identity and a digital wallet.

The following two sections contain the architecture and description of on- and off-chain digital wallets. While the definitions of digital wallets are well-known, we identified through research and analysis of existing literature common components and building blocks of the wallets. We also illustrate how they fit and relate to each other in the architecture, which serves as the basis for the proposed model, which combines both types of wallets.

1) ON-CHAIN WALLETS

On-chain wallets are the key entry point for managing the on-chain decentralized identities. Paramount to on-chain wallets are Web3 wallets, which are the crucial aspect of interacting with DLT or blockchain technology. We call these wallets Web3 since various DLT platforms are not based on blockchains but still enable the same features. Their core feature is to provide secure storage for cryptographic keys, which are used to control DLT accounts (e.g., blockchain addresses), sign transactions, encrypt data in some cases, manage coins/tokens, etc. They also allow users to create multiple accounts with their key pairs, which can be switched between easily.

Web3 wallets employ a deterministic method to generate private keys based on a random seed phrase. The aim of utilizing seed phrases rather than numerical values in binary or hexadecimal format is to render the process more user-friendly. This concept was introduced in the Bitcoin Improvement Proposal 39 (BIP39) [23].

As aforementioned, Web3 wallets can be divided into several types based on their implementation architecture (e.g., mobile, browser, cloud). Unlike most, cloud-based wallets are different because the cryptographic keys are managed by a third party, such as a cryptocurrency exchange, rather than in the user's possession.

Figure 3 illustrates the generalized IT architecture concept of the on-chain wallets regardless of their implementation details. The model is based on the C4[7] model for visualizing software architectures. We chose the container view to maintain enough high-level abstraction from different implementations [24]. The details about specific implementation are avoided in the section about the model since this paper proposes a generalized one and is not coupled tightly with a single implementation. The container model focuses on the core applications and data stores needed for an on-chain wallet to work correctly and fully.

As the Figure depicts, an on-chain wallet consists of several vital interfaces. The cryptographic interface is the essential piece, facilitating the generation, management, and usage of cryptographic key pairs for DLT or blockchain accounts/addresses. Generating a keypair is the required first step for all other features of the on-chain identity. However, on-chain wallets usually enable users to create multiple such keypairs, thus accounts, while supporting various types of keypairs to satisfy different DLT/blockchain platforms. Once a cryptographic keypair is established, it is of high importance that the storage of the private key is secured; thus, the on-chain wallets usually come with sensitive cryptographic material storage space. Its feature is that it stores the private key securely, to ensure its confidentiality, integrity, and availability.

Wallets also come with a user authentication interface, which enables identification, authentication, and authorization functionalities for the user, to enable only the rightful owner of the wallet and its cryptographic private keys to be used. It usually supports various identification and authentication mechanisms, like fingerprint scanning if the wallet is on a smartphone, etc. Another important interface of on-chain wallets is the token interface, which enables the wallet to understand and manage various forms of cryptographic coins and tokens, since the majority of the on-chain use cases involve some form of tokens, e.g., ERC-20 or ERC-721. The token interface has to use the cryptographic interface since the private keys determine the ownership of coins and tokens in an on-chain identity.

Also, the network interface plays a crucial role, since all the user-related data are stored on-chain, and the wallets need to communicate with the DLT/blockchain network to view/use/manage the user's assets, since, without it, it is not operational at all. Hence, the token interface uses it to scan and analyze the DLT/blockchain network for coins/tokens belonging to the respective wallet addresses. Even though the core feature of on-chain wallets is the management of user-respective coins and tokens, there are also various other use cases the wallet could be used for, as we discussed in the on-chain identity section (e.g., DAO voting, using NFTs to play games). For this type of user interaction, a dApp is almost necessary. Practically all dApps use Web3 libraries and features to enable interaction with the user's on-chain wallet; hence, a Web3 interface is also one of the core elements of the wallet. It allows the user to connect to dApps and receive RPC calls from those, thus interacting with the business logic of various dApp use cases. Finally, the on-chain wallets have to provide a GUI for the user, and usually also a configuration interface, where the user can configure, manage, and support various DLT/blockchain networks, accounts, RPCs, etc.

Several standardized flows of interacting with applications with decentralized identities are applied, both on-chain and off-chain. Most common flows are either same-device or cross-device interactions. In the first, the wallet and applications (centralized/decentralized) are on the same device, meaning connection with the wallet happens over the same device calls, usually deep links on mobile phones or through RPC methods using browser extensions. In the cross-device flow, the application and wallet are located on different devices, most often the application on a desktop computer and the wallet on a mobile phone. At the same time, the connection is made by scanning a QR code.

MetaMask is the most popular Web3 wallet, and it falls under the category of software-based crypto wallets [25]. It is available as a browser extension and a standalone mobile application. In addition to basic storage and key management functions, it also allows connection to hardware-based crypto wallets, purchasing supported cryptocurrencies with fiat money, exchanging cryptocurrencies, interacting with smart contracts, etc. Additionally, the display of ERC20 and ERC 721 token balances is supported within the wallet. Digital signing is implemented by the EIP712 [26] and EIP191 [27] standards.Â The mobile version of the wallet also supports the WalletConnect protocol,[8] and provides a built-in browser for accessing decentralized applications (dApps). Furthermore, MetaMask supports the new EIP4361 [28] standard, which specifies how to use Ethereum accounts for authentication and session establishment without relying on traditional centralized identity providers.

2) OFF-CHAIN WALLETS

Off-chain wallets are the key entry point for managing off-chain decentralized identities. The key example of off-chain wallets is SSI wallets. SSI wallets have a more general-purpose use case compared to Web3 wallets, which are used mostly for signing blockchain transactions and messages [4]. Additionally, to manage cryptographic keys, they enable users to control decentralized digital identifiers. Those

[7] https://c4model.com/

[8] https://walletconnect.com/

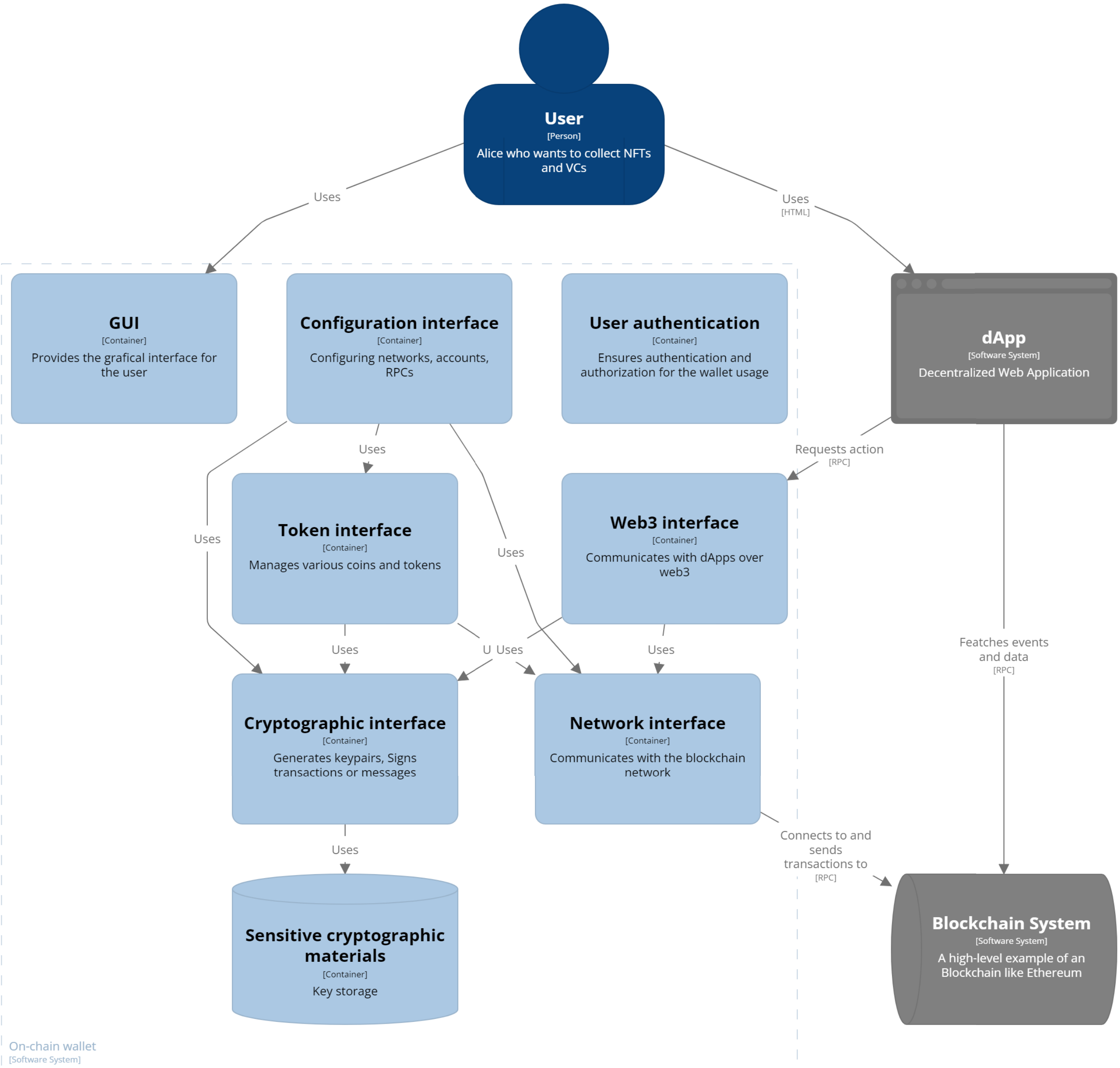

User
[Person]
Alice who wants to collect NFTs and VCs
Uses
Uses
[HTML]
GUI
[Container]
Provides the grafical interface for the user
Configuration interface
[Container]
Configuring networks, accounts, RPCs
User authentication
[Container]
Ensures authentication and authorization for the wallet usage
dApp
[Software System]
Decentralized Web Application
Requests action
[RPC]
Uses
Uses
Uses
Token interface
[Container]
Manages various coins and tokens
Web3 interface
[Container]
Communicates with dApps over web3
Featches events and data
[RPC]
Uses
U Uses
Uses
Cryptographic interface
[Container]
Generates keypairs, Signs transactions or messages
Network interface
[Container]
Communicates with the blockchain network
Connects to and sends transactions to
[RPC]
Uses
Sensitive cryptographic materials
[Container]
Key storage
Blockchain System
[Software System]
A high-level example of an Blockchain like Ethereum
On-chain wallet
[Software System]

FIGURE 3. A C4 model of an on-chain wallet.

can be used for identification, authentication, and digital signing, as well as collecting and managing attestations in the form of VCs [29], [30], [31]. Figure 4 illustrates the generalized IT architecture concept of the off-chain wallets.

As with the on-chain wallet, we chose the container view, to stay on a high enough abstraction from possible implementation versions. As the Figure depicts, an off-chain wallet contains many similar components to an on-chain wallet, such as sensitive cryptographic material, a cryptographic interface, a network interface, a configuration interface, a GUI, user authentication, and a blockchain system. While all of these components have the same role in both types of wallets, some interfaces may support different protocols or algorithms, e.g., the network interface supports DIDComm and OIDC (also OIDC4VC) in the off-chain wallets while sending transactions in the on-chain wallets.

Some new components are specific to off-chain wallets, both internal and external. The three new internal components are storage, a DID interface, and a VC/VP interface. The storage interface's role is to store users' data in the form of VCs, keeping them in the local state. The DID interface handles the DID management - generating DID identifiers, DID documents, and routing the resolution of the identifiers to the correct resolver. There are no constraints on which DID

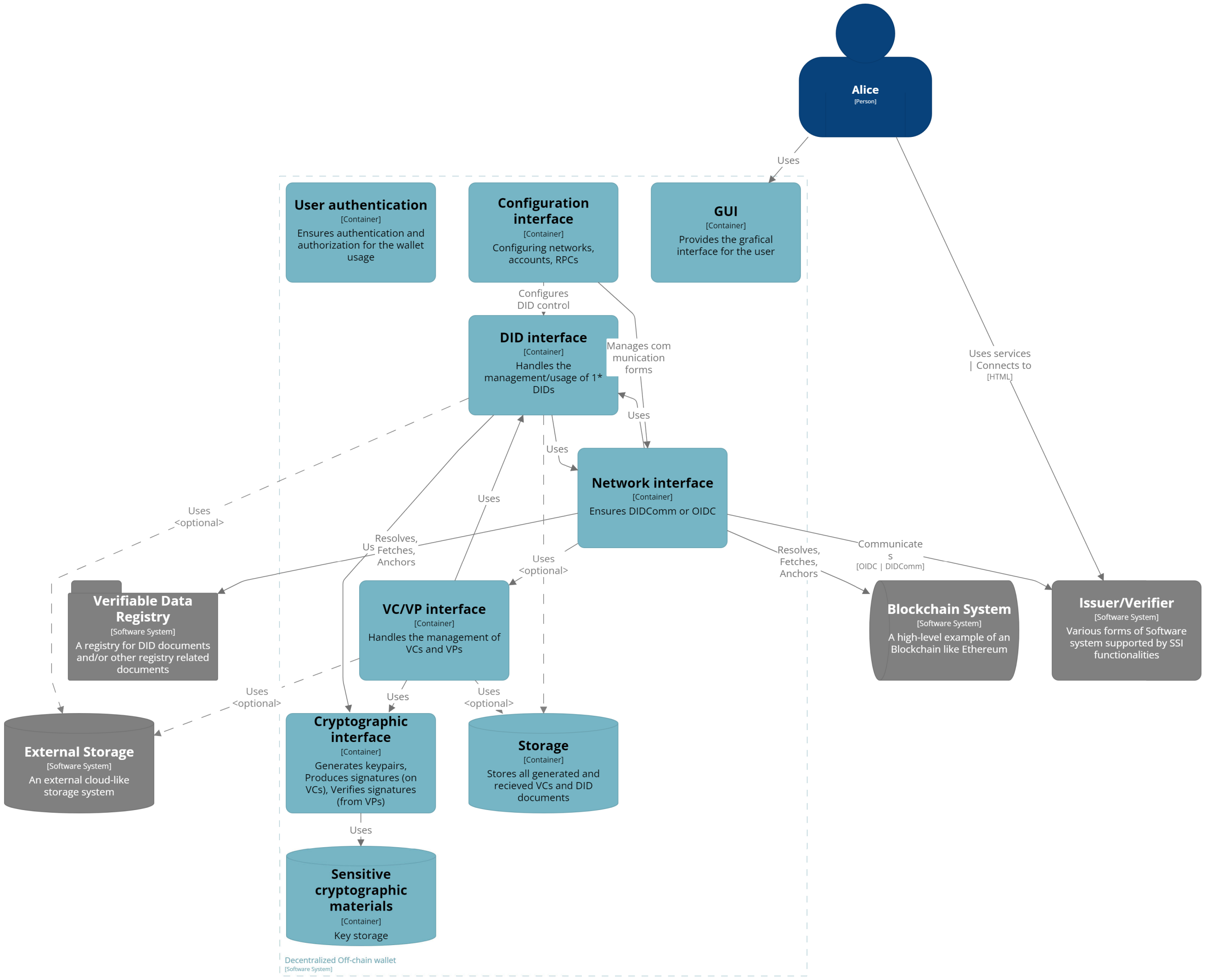

Alice
[Person]
Uses
User authentication
[Container]
Ensures authentication and authorization for the wallet usage
Configuration interface
[Container]
Configuring networks, accounts, RPCs
GUI
[Container]
Provides the grafical interface for the user
Configures DID control
DID interface
[Container]
Handles the management/usage of 1* DIDs
Manages com munication forms
Uses
Uses services | Connects to
[HTML]
Uses
Network interface
[Container]
Ensures DIDComm or OIDC
Uses
Uses <optional>
Resolves, Fetches, Anchors
Us
Uses <optional>
Resolves, Fetches, Anchors
Communicates
[OIDC | DIDComm]
Verifiable Data Registry
[Software System]
A registry for DID documents and/or other registry related documents
VC/VP interface
[Container]
Handles the management of VCs and VPs
Blockchain System
[Software System]
A high-level example of an Blockchain like Ethereum
Issuer/Verifier
[Software System]
Various forms of Software system supported by SSI functionalities
Uses <optional>
Uses
Uses <optional>
External Storage
[Software System]
An external cloud-like storage system
Cryptographic interface
[Container]
Generates keypairs, Produces signatures (on VCs), Verifies signatures (from VPs)
Storage
[Container]
Stores all generated and recieved VCs and DID documents
Uses
Sensitive cryptographic materials
[Container]
Key storage
Decentralized Off-chain wallet
[Software System]

FIGURE 4. A C4 model of an off-chain wallet.

methods can be supported since all DID methods rely on the same base components, such as cryptographic keys and communication interfaces. Thus, public DID methods, such as *did:ebsi* and *did:web*, and special-purpose or permanent, such as *did:peer* and *did:key*, can be supported. The VC/VP interface manages the interaction with different storage providers, and uses the cryptographic interface to generate various VC proof types, such as JWTs and JSON Linked Data Proofs.

There are also external systems that are specific for SSI, such as external storage, verifiable data registries (VDR), and issuer/verifiers. External storages are optional and are similar to internal storage, but they are located on the Internet and not in the wallet itself, and only authenticated users can access the data. Verifiable data registries are non-local public services that enable the storage of credentials and lists of trusted entities, such as verified issuers or certified organizations. Sometimes they also store public DID documents (e.g., *did:github*), which are not on-chain, but off-chain. An example of such a Verifiable Data Registry service are the various registries on the European Blockchain Services Infrastructure (EBSI), e.g., the Trusted Issuer Registry (TIR) [32]. However, EBSI also enables the on-chain DID Registry and the off-chain TIR.

There are already several known off-chain wallets in the market, such as the Trinsic wallet,[9] Lissi wallet[10] and Gataca wallet.[11] Most SSI wallets are currently being developed as mobile applications, compared to Web3 wallets, which often provide browser extensions and mobile applications.

IV. MODEL

A. CHALLENGE

Based on the decentralized identity landscape, focusing on the on- and off-chain wallets, the user-experience complexity increases steadily. Let’s consider a user who onboards on the

[9] https://trinsic.id/trinsic-wallet/
[10] https://www.lissi.id/
[11] https://www.gataca.io/

decentralized identity utilizing an on-chain wallet. The reasons for it can be various, whereby most of those are related to the use cases on DLT/blockchain systems (e.g., cryptographic token, coin, and NFT collection). Should the same user also choose to onboard on the usage of off-chain identity, which is ever more popular due to the support for VC/VP etc., they would need to install an off-chain like SSI Wallet, especially if VCs mustn't be stored on public networks, e.g., the Ceramic Network.[12] Even if we neglect that there are two wallets the user has to use and manage, there is also the fact that the user has two sets of sensitive cryptographic material storage in two different software systems, thus increasing the fear of possible bugs and security hacks.

Moreover, we must consider that the same user will often use centralized and federate identities due to other formal use cases (e.g., eGovernment, e-commerce). There is also a high probability that most centralized and federated-based digital identities will be wallet-driven, as with the European Digital Identity Wallet (EUDIW) [33]. In such a scenario, a user must use and manage three, or even more, wallets and other software systems, as depicted in Figure 5.

In order to simplify the digital identity landscape, we propose orchestrating decentralized identity management in the manner described in the following paragraphs.

B. ORCHESTRATION

In contrast to centralized and federated digital identity models, the key entry points and components for decentralized identity management are the digital identity wallets. After analyzing both on- and off-chain wallets carefully, we can identify several vital interfaces, which are more or less the same in both wallets (Figure 6).

Figure 7 illustrates both architectures, while interfaces, which are the same, are color-coded identically, i.e., blue. The C4 DSL-based source code of the complete architecture is open-sourced and published on GitHub.[13]

The core interface in both cases is the cryptographic interface, which has the same role and functionality, i.e., the generation and usage of cryptographic keypairs and the secure storage of related private keys in the sensitive cryptographic material storage. From the interface abstraction, we can conclude that these are the same. However, on the component level, and based on the implementation, there is a requirement that this interface supports those cryptographic libraries and primitives which are used in both on- and off-chain identity models. However, considering the current wallet implementation, this situation is already diverse, e.g., some on-chain wallets only support elliptic curves for the Ethereum platform, while some off-chain wallets support all possible curves for the *did:key*. However, some off-chain wallets already support keypairs for *did:ethr*, which uses the same cryptographic primitives as those in the on-chain wallets which support the Ethereum platform. Therefore, we can conclude that, on the higher abstraction level, the cryptographic interface could and should support those cryptographic libraries and primitives, which will satisfy the most common use cases for both the on- and off-chain identity models.

Similarly, the network interface is also one of the core interfaces for both wallet types. While the on-chain version of the network interface is to support the connection to and communication with the DLT/blockchain network (i.e., sending transactions, querying the ledger for events, etc.), the role of the off-chain version interface is almost identical. Still, for different reasons, i.e., it sometimes has to connect to a DLT/blockchain to anchor or resolve public DIDs. However, the off-chain version also connects to and communicates with various public registries, which are only sometimes DLT/blockchain-based. Considering the current wallet implementations, we can already find multiple solutions which support various DLT/blockchain networks and registries. The network interface should, thus, support both RPC and HTTP communications. At the same time, the end implementation would need to ensure the support for those networks, which will satisfy the most common use cases for both the on- and off-chain identity models.

As is evident, multiple aspects would require various configuration settings from the user, which could be grouped in one configuration interface that supports both on- (e.g., address, blockchain network) and off-chain (e.g., DIDs, registry) related configurations. Lastly, from the perspective of the standard interface, user authentication and GUI can be merged, and thus encapsulate decentralized identity management. Besides the core and common interfaces mentioned earlier, we should remember the important distinct interfaces present in the on- and off-chain worlds, i.e., the token and Web3 interface for the former and the DID and VC/VP interfaces for the latter. Each should support its main features in connection to the other common interfaces.

C. SOLUTION

In this Section we present a solution for the orchestration of decentralized identity management. We base our proposed solution on the existing implementation of on- and off-chain wallets and their respective supporting tools and frameworks. The presented solution proves that orchestrating decentralized identity management is theoretically possible and executable with the current technological ecosystem. For the foundation of our proposed solution, we chose to use the Web3 wallet Metamask.

MetaMask does not support off-chain functionalities out of the box. To address this issue, we concentrated on Metamask's newly introduced feature, Snaps, which provides the environment to extend the wallet's functionality by creating extensions/plugins [34]. As a Web3 wallet and gateway to blockchain apps, MetaMask provides an easy-to-use interface for users to interact with EVM-based blockchains, sign transactions, and more. Snaps enable extending MetaMask to support other networks, such as Polkadot, Solana, and

[12] https://ceramic.network/

[13] https://github.com/blockchain-lab-um/on_off_chain_wallet_C4

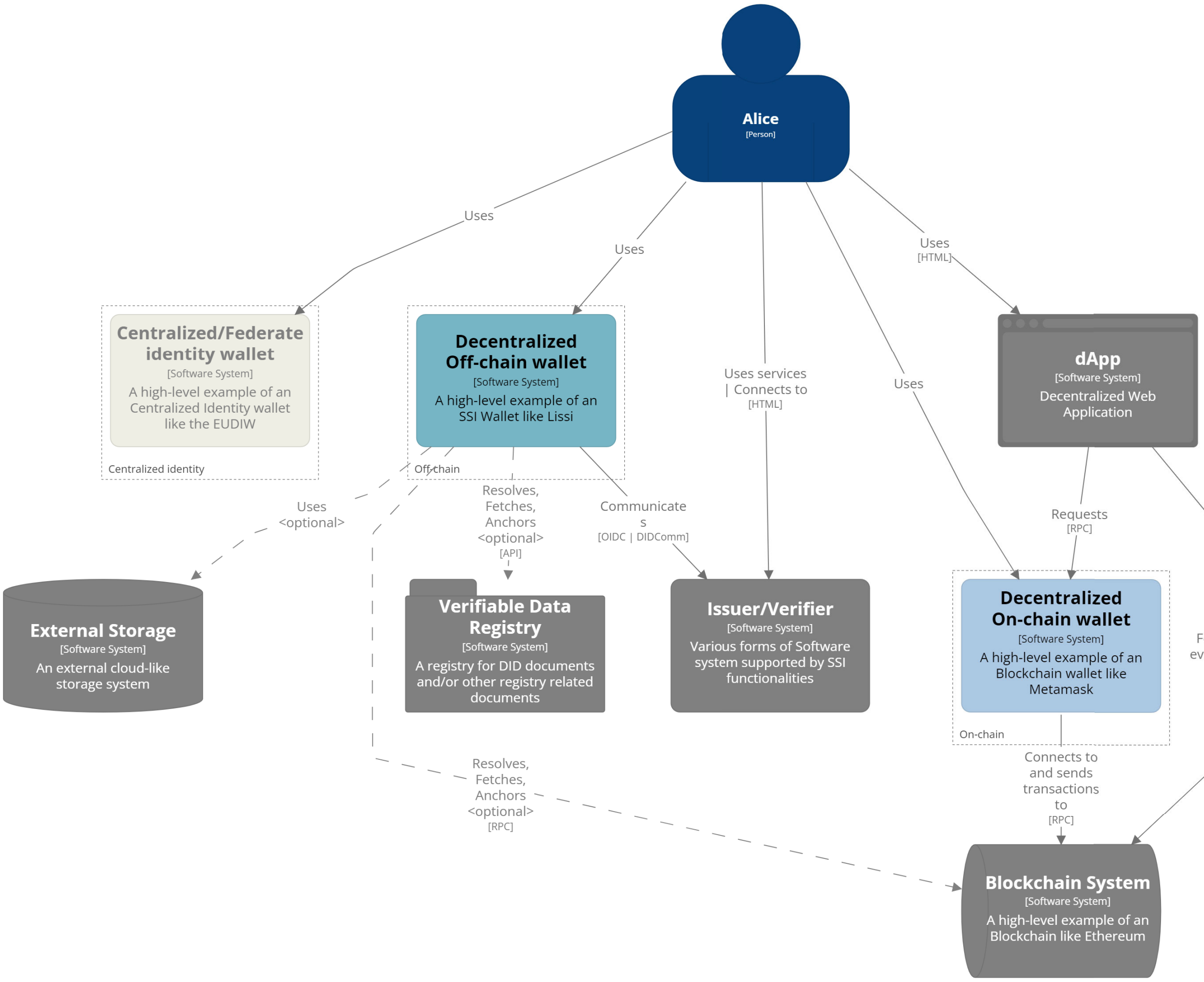

Alice
[Person]
Uses
Uses
Uses services
| Connects to
[HTML]
Uses
Uses
[HTML]
Centralized/Federate identity wallet
[Software System]
A high-level example of an Centralized Identity wallet like the EUDIW
Centralized identity
Decentralized Off-chain wallet
[Software System]
A high-level example of an SSI Wallet like Lissi
Off-chain
dApp
[Software System]
Decentralized Web Application
Uses
<optional>
Resolves,
Fetches,
Anchors
<optional>
[API]
Communicate
s
[OIDC | DIDComm]
Requests
[RPC]
Featches
events and
data
[RPC]
External Storage
[Software System]
An external cloud-like storage system
Verifiable Data Registry
[Software System]
A registry for DID documents and/or other registry related documents
Issuer/Verifier
[Software System]
Various forms of Software system supported by SSI functionalities
Decentralized On-chain wallet
[Software System]
A high-level example of an Blockchain wallet like Metamask
On-chain
Resolves,
Fetches,
Anchors
<optional>
[RPC]
Connects to
and sends
transactions
to
[RPC]
Blockchain System
[Software System]
A high-level example of an Blockchain like Ethereum

FIGURE 5. System context model (C4 model) of the on- and off-chain wallet usage.

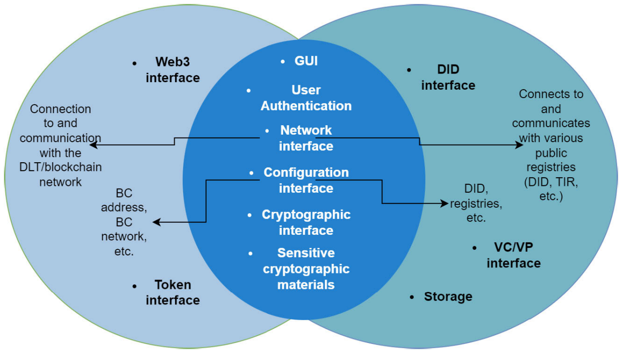

Web3 interface
GUI
DID interface
User Authentication
Connection to and communication with the DLT/blockchain network
Network interface
Connects to and communicates with various public registries (DID, TIR, etc.)
Configuration interface
BC address, BC network, etc.
DID, registries, etc.
Cryptographic interface
Sensitive cryptographic materials
VC/VP interface
Token interface
Storage

FIGURE 6. An overview of equal and specific interfaces of on- and off-chain decentralized wallets.

Bitcoin, as well as any other custom functionalities, such as simulating transactions before submitting them. Developers can add new RPC API methods to the existing ones, which are then used by dApps.

MetaMask, the most used Web3 wallet, supports all functionalities of the on-chain wallets, while we design and develop the needed orchestrating extension of the MetaMask to support of-chain capabilities and functionalities. These functionalities are encapsulated inside the Snap, which runs in an isolated environment, thus unable to access the internal system of the MetaMask directly. Still, they can call dedicated exposed RPC methods, e.g., retrieving and deriving private keys from the MetaMask cryptographic keys interface. By calling these methods we can use on-chain interfaces from the off-chain wallet component.

Off-chain capabilities are implemented using the open-source framework Veramo, which provides functionalities to handle different DID methods, cryptographic signatures, etc. By importing cryptographic material to Veramo, we can reuse the same cryptographic material for both on-chain and off-chain identities, thus having a single source for cryptographic keys, i.e., a mnemonic phrase from MetaMask. By relying on the same cryptographic material, blockchain-based DID methods, such as *did:pkh*, result in the same identifier as the blockchain addresses for on-chain identities.

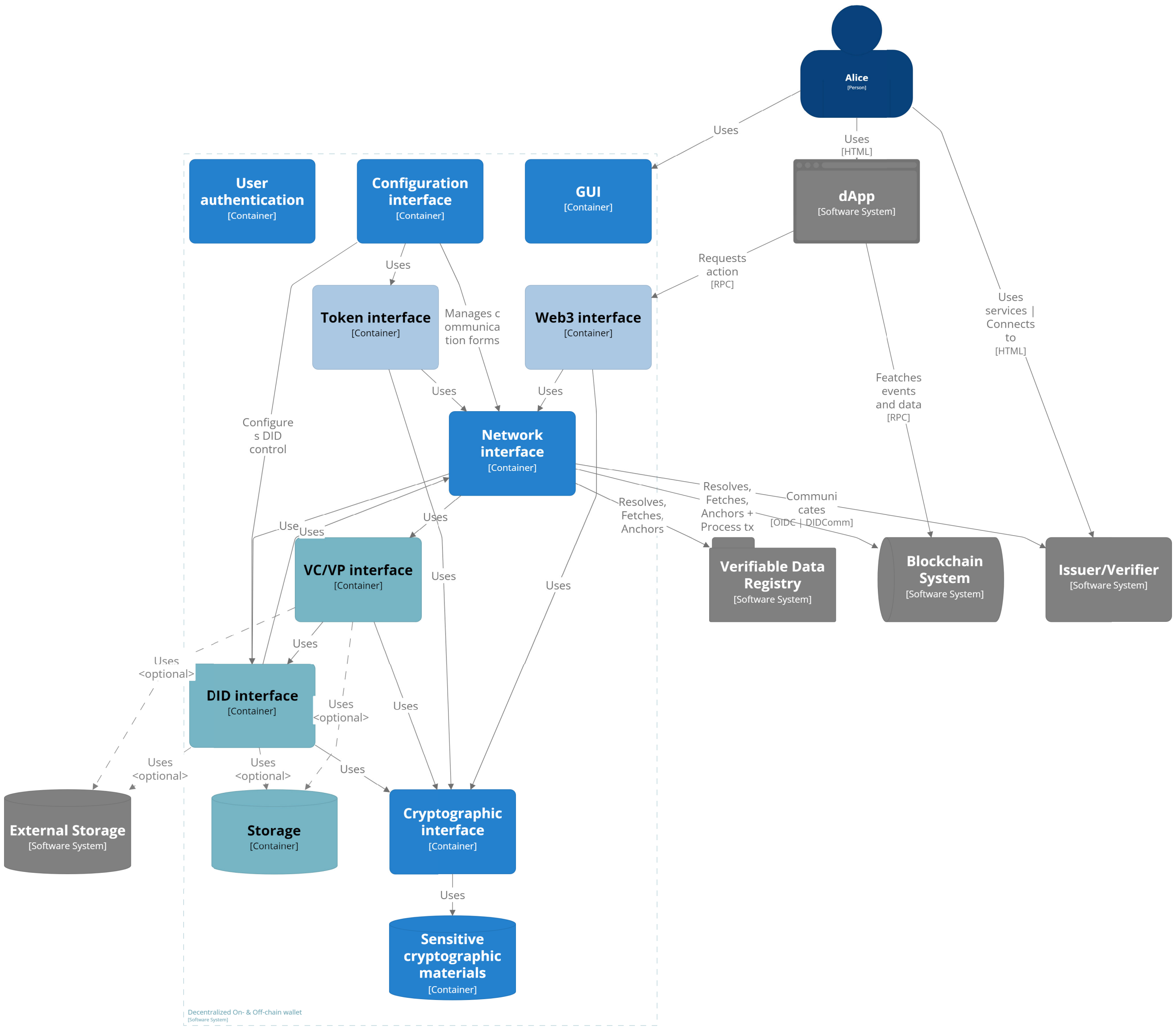


FIGURE 7. A joined architecture view (C4 model) of an on- and off-chain wallet, showcasing equal interfaces.

Additional common components can be reused for both types of decentralized identity, e.g., the configuration and network interfaces. By configuring the blockchain to a particular EVM-based network in MetaMask, we can reuse that information in the off-chain wallet, thus generating *did:ethr* identifiers for the same network that MetaMask is connected to. The network interface can also be reused, meaning transactions and interactions with the smart contract on the EVM-based blockchains can be sent with the same network interface.

Some components must be built from scratch in the off-chain wallet component, due to not being supported in the on-chain wallets. The VC interface handles generating different proof formats, such as JWTs and JSON-LD proofs. It also provides access to various storage providers, such as local Snap state or integration with the Ceramic Network. The network interface must support communication protocols such as DIDComm and OIDC4VC when interacting with issuers and verifiers and resolving and using different trusted registries, such as the ones on the EBSI. The configuration interface must be extended with additional features, such as switch DID methods being similar to switching blockchain networks. Figure 8 shows the final Snap architecture. We used the C4 model to illustrate the functioning orchestrated solution for decentralized identity management. We use the same color coding of the model elements as set out in previous Figures, thus emphasizing the elements/interfaces which are purely on-chain (light blue) or purely off-chain (light green), and the common (blue). Furthermore, we added an icon (right upper corner of specific interfaces) indicating

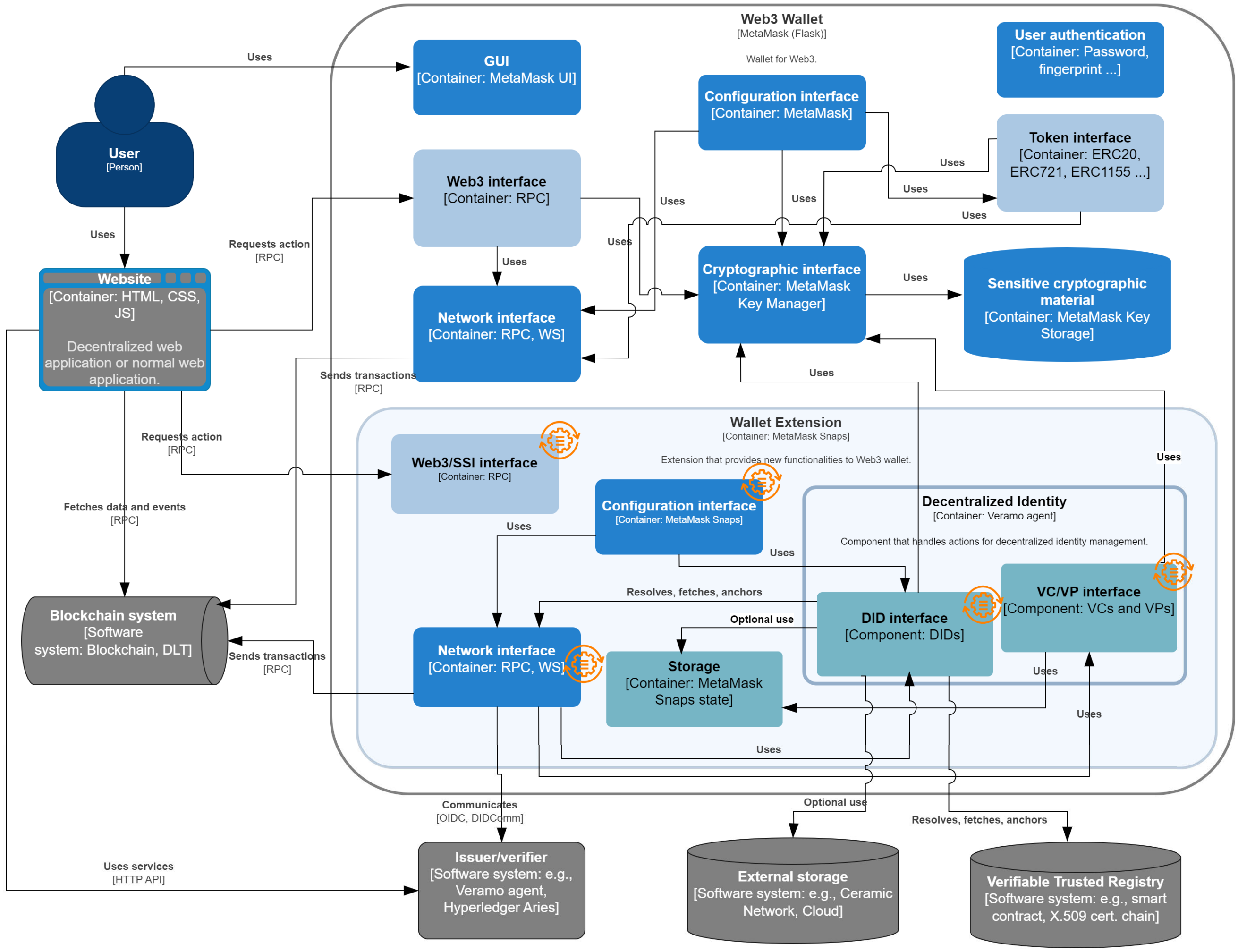

Web3 Wallet
[MetaMask (Flask)]
Wallet for Web3.
User
[Person]
Uses
Uses
GUI
[Container: MetaMask UI]
User authentication
[Container: Password, fingerprint ...]
Configuration interface
[Container: MetaMask]
Token interface
[Container: ERC20, ERC721, ERC1155 ...]
Web3 interface
[Container: RPC]
Requests action
[RPC]
Uses
Uses
Uses
Uses
Uses
Uses
Website
[Container: HTML, CSS, JS]
Decentralized web application or normal web application.
Network interface
[Container: RPC, WS]
Cryptographic interface
[Container: MetaMask Key Manager]
Uses
Sensitive cryptographic material
[Container: MetaMask Key Storage]
Sends transactions
[RPC]
Uses
Wallet Extension
[Container: MetaMask Snaps]
Extension that provides new functionalities to Web3 wallet.
Requests action
[RPC]
Web3/SSI interface
[Container: RPC]
Uses
Configuration interface
[Container: MetaMask Snaps]
Decentralized Identity
[Container: Veramo agent]
Component that handles actions for decentralized identity management.
Fetches data and events
[RPC]
Uses
Uses
Resolves, fetches, anchors
VC/VP interface
[Component: VCs and VPs]
Optional use
DID interface
[Component: DIDs]
Blockchain system
[Software system: Blockchain, DLT]
Sends transactions
[RPC]
Network interface
[Container: RPC, WS]
Storage
[Container: MetaMask Snaps state]
Uses
Uses
Uses
Communicates
[OIDC, DIDComm]
Optional use
Resolves, fetches, anchors
Uses services
[HTTP API]
Issuer/verifier
[Software system: e.g., Veramo agent, Hyperledger Aries]
External storage
[Software system: e.g., Ceramic Network, Cloud]
Verifiable Trusted Registry
[Software system: e.g., smart contract, X.509 cert. chain]

FIGURE 8. A C4 model of the implemented solution for the orchestration of on- and off-chain wallets.

an interface which had to be adapted to fulfill their new feature.

V. VALIDATION AND EVALUATION

Validation of the proposed solution was performed on a real-life problem – a plutocracy problem in decentralized autonomous organizations. DAOs are organizations controlled by a community, not a central entity. They can be run on-chain in the form of smart contracts, or off-chain, by using cryptographically signed messages. In both ways, members of the organizations use their decentralized identity to decide and vote on governance proposals, with voting being secured and verifiable by cryptography. When they are run on-chain, decision-making can be executed automatically when the governance voting ends. Thus the primary objective of DAOs is to provide autonomous, decentralized, and transparent governance, with the latter being achieved by blockchain technology.

Snapshot is one of the most used platforms for decentralized governance in DAOs, providing an intuitive interface for governance voting and supporting several voting mechanisms, i.e., determining how the voting power is distributed. The vote happens off-chain by signing blockchain messages due to not having to pay gas fees, as is necessary with standard blockchain transactions. Thus, this kind of voting is often used as the first step of the governance, the so-called temperature check, with the final voting happening later on-chain. Each vote, i.e., signed blockchain message, is stored on the InterPlanetary File System (IPFS), and can be retrieved and verified by everyone.

The most popular voting mechanism utilized by the majority of the projects is token-based voting, where voting power is based on the number of governance tokens the users hold on their blockchain account. This results in the plutocracy problem, where the wealthy participants have more power, since they can buy more governance tokens. However, most organizations work better and more efficiently if the power is leaning towards meritocracy, where power is determined based on the knowledge and skills of the organization's members. This problem can be solved by having the voting mechanisms based on personal data representing skills, which can be modeled using off-chain identity primitives, such as diplomas or course certificates in the form of VCs.

We developed three components:

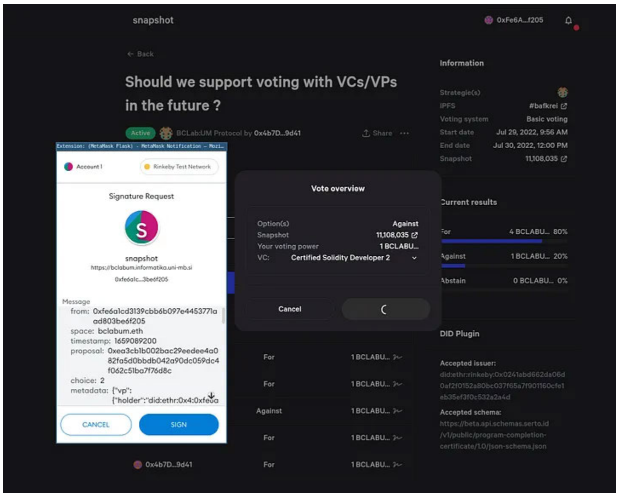
- Demo course platform for receiving VCs.
- Extended Snapshot platform that supports voting based on VCs.
- Connector library for integrating Snap into dApps and other web applications.

The proof-of-concept (PoC) runs on the Ethereum blockchain platform due to several reasons, such as being one of the most popular and developed platforms for smart contract development, the best support for the DID method *did:ethr*, the most decentralized and public permissionless blockchain networks, plenty of already established frameworks and libraries, etc. Source code is open-sourced and published on GitHub.[14]

Only asymmetric cryptography was needed for our specific use case, specifically for identity-related operations, e.g., digital signatures. Digital signatures are required when creating VCs, VPs, and voting by signing blockchain messages. Because we built on top of the Ethereum blockchain, we added support only for cryptographic keys of the elliptic curve *secp256k1* and the *ECDSA* (*ES256K*) signing algorithm. The implementation was made extensible, making it possible to add new cryptographic key types in the future, such as *Ed25519*. While VCs can be in different data formats, we used the *EIP-712* signature format exclusively because the Ethereum community adopted this standard widely.

On the course platform, users can connect with their MetaMask wallet, whereas the connection is handled with a connector library, which abstracts custom RPC methods with JavaScript functions. Users can complete a mockup of Solidity course development on the platform, obtaining a VC that presents their skills in smart contract development. This VC is signed digitally and received from the platform issuer, and is stored inside the MetaMask Snap. Users can also view and generate VPs on the Demo platform.

On the Snapshot platform, we added the mechanism to select the necessary credentials to vote on specific proposals by choosing the VC issuer's DID and credential schema. Thus, only users with particular credentials are able to vote in the governance proposals. For example, this enables programming proposals that tune certain parameters of smart contracts, being voted only by certified Solidity developers (Figure 9). Users must add VPs with the correct VC to the Snapshot vote message when voting. The signed message with the vote and VP (added inside the metadata) are stored on the IPFS, where they can be retrieved, verified, and used for the voting results.



Should we support voting with VCs/VPs in the future ?

Signature Request

snapshot
https://bclabum.informatika.uni-mb.si
0xfe6a1c...3be6f205

Message
from: 0xfe6a1cd3139cbb6b097e4453771a ad803be6f205
space: bclabum.eth
timestamp: 1659089200
proposal: 0xea3cb1b002bac29eedee4a0 82fa5d0bbdb042a90dc059dc4 f062c51ba7f76d8c
choice: 2
metadata: {"vp": {"holder":"did:ethr:0x4:0xfe6a

Vote overview

Option(s)	Against
Snapshot	11,108,035
Your voting power	1 BCLABU...
VC:	Certified Solidity Developer 2

For	1 BCLABU...
For	1 BCLABU...
Against	1 BCLABU...
For	1 BCLABU...
For	1 BCLABU...

Information

Strategie(s)	
IPFS	#bafkrei
Voting system	Basic voting
Start date	Jul 29, 2022, 9:56 AM
End date	Jul 30, 2022, 12:00 PM
Snapshot	11,108,035

Current results

For	4 BCLABU...	80%
Against	1 BCLABU...	20%
Abstain	0 BCLABU...	0%

DID Plugin

Accepted issuer:
did:ethr:rinkeby:0x0241abd662da06d0af2f0152a80bc037f65a7f901160cfe1eb35ef3f0c532a2a4d

Accepted schema:
https://beta.api.schemas.serto.id/v1/public/program-completion-certificate/1.0/json-schema.json

FIGURE 9. Voting on the Snapshot platform using MetaMask Snap and VC.

The validation showcases how the orchestration between on-chain and off-chain identities is feasible. The user can perform all the actions, such as connecting on-chain identity and signing messages, in the same way as before on the dApps (in our case, Snapshot). At the same time, the off-chain wallet component provides the functionalities of off-chain identity, e.g., using and adding proofs in the form of VCs/VPs.

A. SECURITY THREAT ANALYSIS

For the security analysis we chose to apply the STRIDE modeling framework, to validate the proposed solution against the security vulnerability [35]. Based on the modeling methodology, we identified several possible attack scenarios for each threat type, i.e., spoofing, tampering, repudiation, information disclosure, denial of service (DoS), and elevation of privilege. For illustrative purposes, a Data Flow Diagram (DFD) (Figure 10) was created to showcase the trust boundary, trust zones, data flows, and communications over the network. We have also identified the solution's most critical components and parts, highlighted with the red rectangles in the DFD. The majority of critical parts is on the level of communication between the orchestrated solution and all the external entities, while, from the inner trust boundary, those are the data stores, user authentication, the cryptographic interfaces, and the VC/VP and DID interfaces. A detailed overview follows.

1) SPOOFING

Similar to other identity and blockchain solutions that work with cryptography, malicious malware can be injected that steals the cryptographic material of the users (DS-1) and uses it for other spoofing activities toward others. Even though the cryptographic material is encrypted, it presents a problem due to quantum computers' possible future breaking of some cryptographic algorithms. This enables the attackers to use users' identities, either on- or off-chain. Hardware wallets, where cryptographic keys never leave the external physical

[14] https://github.com/blockchain-lab-um/ssi-snap

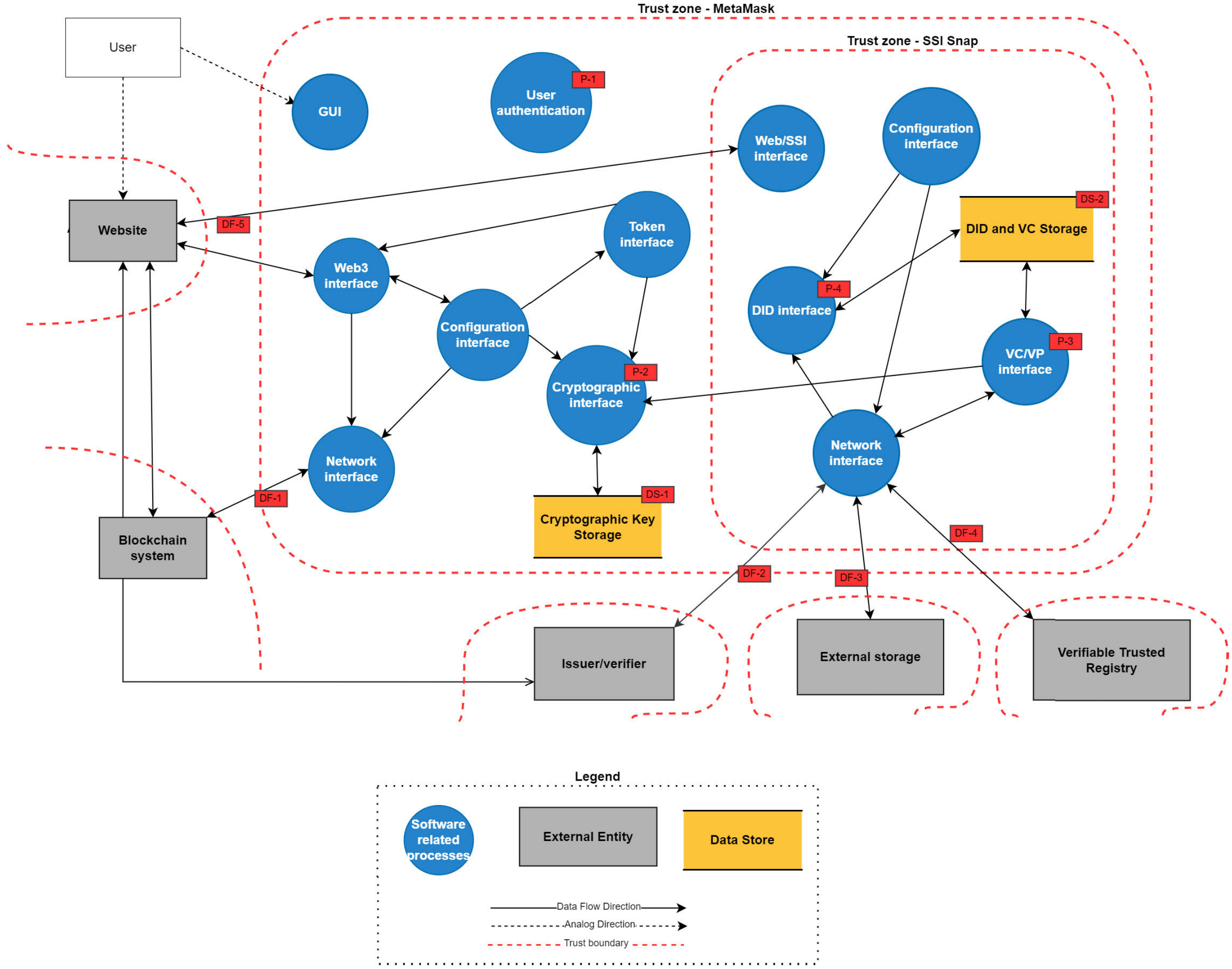


FIGURE 10. DFD of the implemented solution of an on- and off-chain wallet.

device, can prevent such scenarios. The proposed solution is generalized, so that cryptographic key storage can be in different forms. Stealing personal data (DS-2), i.e., a passport as a VC, can also lead to spoofing if issuers and verifiers do not employ well-established and standardized protocols for data exchange, e.g., OIDC and DIDComm, that verify proof of possession of cryptographic keys or VP generation on the fly during the protocol execution (DF-1, DF-2, DF-3, and DF-4). Another critical part is user authentication, which also ensures encryption of cryptographic material, whereas, if the attacker can bypass it, e.g., by stealing the device and password, take the user's identity (P-1).

2) TAMPERING

A payload that needs to be signed by the user's wallet can come from two different sources - issues/verifiers and dApps. If the payload verification bypasses the verification checks, the wallet can sign arbitrary documents or data, or the malicious data passes inside a wallet (P-2). Another unauthorized access to the wallet's components can happen if the vulnerability is found in third-party (open-source) dependencies.

3) REPUDIATION

While this entity is not strictly part of the wallet itself, disabling access to external storages or verifiable trusted registries prevents users from accessing their data or identity data, e.g., DID documents, which are needed to verify digital signatures, since they contain information about public keys (DF-1, DF-3, and DF-4). This can be mitigated by backup data to local storage inside wallets, or using DID methods that don't rely on verifiable trust registries, such as *did:key* and *did:pkh*. For any other classical repudiation attack scenarios, the proposed solution should employ extensive secure logging on all interactions with external entities, thus, on the network interfaces, as well as on the user authentication and cryptographic interfaces. Furthermore, any possible tampering with the VCs or DIDs would be detectable, due to the digital signatures and their integrity constraints, thus mitigating repudiation actions for attackers.

4) INFORMATION DISCLOSURE

Unauthorized disclosure of information can happen if the user is tricked by issuers/verifiers to present more data or not intended personal data during a VC/VP exchange (P-3, P-4, and DF-2). For this, the user is always asked to confirm and react to any VC/VP exchange request. Since the idea for using VCs is also their support for using ZKP, unintended information disclosure could happen if the ZKP process (P-2) would not work correctly. However, support for ZKP with VC is still in its infancy, and will be addressed with other future research. Another possible attack can happen in our validated implementation by malicious dApps, which try to access SSI Snap data by calling RPC methods and requesting different personal VCs (DF-5), which again should be mitigated with the user's active involvement.

5) DENIAL OF SERVICE (DoS)

Since the proposed solution is the wallet which often creates requests to other external services, i.e., servers, there are fewer opportunities for DoS attacks than in the server or cloud-based applications, which receive requests from the public Internet. Nevertheless, there are still some attack possibilities, such as issuers sending a large number of VCs to the wallet, which can result in a large storage footprint or crashing the wallet itself (but not losing data), and a large amount of RPC requests from the connected dApp, which also leads to wallet crashing (DF-2 and DF-6). The threat does not pose a high risk, since no data will be lost, while the mitigation perspective includes application-based firewalls on the network interfaces.

6) ELEVATION OF PRIVILEGE

When using their on- and off-chain identities, users rely on the data received/fetched from external entities, e.g., swapping tokens based on the current exchange rate in some decentralized exchange (DEX) (DF-1). Corrupting the received data can result in users performing actions or transactions in the attacker's interests. Another possible attack is dApp pushing the malicious transaction or payload to a user to sign, resulting in the attacker obtaining users' permissions with their digital signatures (P-2 and DF-1).

VI. DISCUSSION

Based on the analysis of the proposed working solution, we can compare the functionalities satisfied by the on- and off-chain identity models and the orchestration achieved. Table 1 summarizes the comparison, whereby it is evident that, by supporting both on- and off-chain wallets' key different interfaces, one can satisfy all the decentralized identity requirements.

As seen in the Table, both on- and off-chain wallets support only some of the functionalities and use cases of the decentralized identity. Thus, there is a need for a universal wallet, that can act and serve both use cases, on-chain, off-chain, and in combination. Having a wallet that can fit all use cases improves the user experience and interaction with such applications without having several wallets in use.

TABLE 1. Comparison of functionalities satisfied by the proposed solution.

Example	On-chain wallet Web3 (Metamask)	Off-chain wallet SSI (Lissi)	Proposed solution MetaMask & Snap
Functionality/Feature			
Generating multiple key-pairs	Yes	Yes	Yes
Supporting user configurations	Yes	Yes	Yes
Supporting user authentication	Yes	Yes	Yes
Providing GUI	Yes	Yes	Yes
Sign blockchain tx	Yes	No	Yes
Sign blockchain message	Yes	No	Yes
Send signed tx to blockchain network	Yes	No	Yes
Support web3 interactions	Yes	No	Yes
Support for DID methods	No	Yes	Yes
Support for handling VC/VP	No	Yes	Yes
Support for DIDComm and OIDC	No	Yes	Yes

While the Web3 wallets are already well established and used in the decentralized space, off-chain wallets still lack widespread adoption, which prevents the adoption of several features and use cases that could improve the applications and decentralized identity management in general, e.g., data confidentiality and privacy.

Many projects in the Web3 space are trying to integrate DIDs and VCs, either through expanding the application (frontend) code or using cloud/public network solutions for handling VCs, because of the need for more wallet infrastructure. The first approach is missing the shared configuration of the identity across different applications, e.g., a selected VC storage provider and DID method. In contrast, the second defies the whole essence of self-sovereignty of the identities.

By solving the challenges mentioned above, we summarize the main contributions of this paper:

- outlining and defining the decentralized identity model and its various sub-forms (on- and off-chain),
- presenting the building blocks of the on- and off-chain digital identity wallets,
- presenting an approach to linking off- and on-chain decentralized identities;
- orchestration of core features from both off- and on-chain decentralized identity models;
- enabling SSI-based encapsulation of on-chain decentralized identities;
- defining the solution to the challenges in the form of software architecture and components, which enable seamless integration of the two ecosystems;

- providing a working digital wallet solution for all the features mentioned above;
- validating the concept on a real-world decentralized identity use case.

Although the proposed approach achieves a unified system for decentralized identity management, it still has some limitations and constraints. One of the limitations is increased complexity in relation to separate wallet types, as well as a single component for cryptographic material, which means that, in the case of the compromise, an attacker can gain access to both digital identity and assets. As mentioned in the Introduction, we focused only on the user-centric approach, and not on machine-centric, as it would be the case with, e.g., IoT devices. Another constraint of the implemented solution is the tight dependence on the Web3 wallet MetaMask, meaning potential bugs in its codebase affect the security of the whole universal wallet.

VII. CONCLUSION

In this paper, we identified the building blocks and components of on- and off-chain digital wallets and introduced a generalized model for universal wallets. We also highlighted and analyzed the intersection of both types of wallets. While most wallet implementors focus on and develop only one type of wallet, this research shows that the more generalized approach is more suitable to support multiple types of decentralized identity.

To evaluate the proposed method, we extended the on-chain wallet MetaMask to support the SSI while reusing several existing components, such as cryptographic material, network interfaces, etc. The validation was performed on the use case of DAOs, where both types of identity data, on- and off-chain, are necessary to improve the governance process. By having a single wallet for both types of identity, user workflows and experience are better, as well as data sovereignty and privacy. Users do not have to have multiple wallets, and they control their data completely. We also performed STRIDE security threat modeling in the discussion, highlighting possible attack vectors on the proposed solution and system, and explaining how they can be avoided.

A. FUTURE WORK

While the proposed architecture and implemented solution cover many use cases, there are still some open questions and possible upgrades. On the architecture level, additional research should be done on integrating hardware wallets and other hardware security modules (HSM). While many on-chain wallets already support using them as cryptographic key storage, they are not used so often in the SSI community. Another research direction is extending the wallet to be fully compliant according to the EUDIW reference framework. While several standards, e.g., OIDC, are already positioned in the current model, further detailed research should be conducted to analyze all the requirements.

As mentioned in the article, the focus of this paper was to propose a generalized system for the unification of on- and off-chain identity management in the form of digital wallets. Therefore, the C4 container view/diagram was used throughout the paper. For future work, exploring lower-level details of multiple implementations and analyzing the differences and best practices would be interesting.

REFERENCES

[1] O. Avellaneda, A. Bachmann, A. Barbir, J. Brenan, P. Dingle, K. H. Duffy, E. Maler, D. Reed, and M. Sporny, "Decentralized identity: Where did it come from and where is it going?" *IEEE Commun. Standards Mag.*, vol. 3, no. 4, pp. 10–13, Dec. 2019.

[2] O. Jacobovitz, "Blockchain for identity management," The Lynne William Frankel Center Comput. Sci., Dept. Comput. Sci., Ben-Gurion Univ., Beer Sheva, Israel, Tech. Rep. 16-02, Dec. 2016.

[3] A. Preukschat and D. Reed, *Self-Sovereign Identity*. Shelter Island, NY, USA: Manning Publications, 2021.

[4] B. Podgorelec, L. Alber, and T. Zefferer, "What is a (Digital) identity wallet? A systematic literature review," in *Proc. IEEE 46th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jun. 2022, pp. 809–818.

[5] N. Naik and P. Jenkins, "Sovrin network for decentralized digital identity: Analysing a self-sovereign identity system based on distributed ledger technology," in *Proc. IEEE Int. Symp. Syst. Eng. (ISSE)*, Sep. 2021, pp. 1–7.

[6] N. Naik and P. Jenkins, "Self-sovereign identity specifications: Govern your identity through your digital wallet using blockchain technology," in *Proc. 8th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng. (MobileCloud)*, Aug. 2020, pp. 90–95.

[7] M. S. Ferdous, A. Ionita, and W. Prinz, "SSI4Web: A self-sovereign identity (SSI) framework for the web," in *Proc. Int. Congr. Blockchain Appl.* Cham, Switzerland: Springer, 2023, pp. 366–379.

[8] A. Abraham, C. Schinnerl, and S. More, "SSI strong authentication using a mobile-phone based identity wallet reaching a high level of assurance," in *SECRYPT*, 2021, pp. 137–148.

[9] W. Dai, J. Deng, Q. Wang, C. Cui, D. Zou, and H. Jin, "SBLWT: A secure blockchain lightweight wallet based on trustzone," *IEEE Access*, vol. 6, pp. 40638–40648, 2018.

[10] K. Singh, N. Singh, and D. S. Kushwaha, "An interoperable and secure e-wallet architecture based on digital ledger technology using blockchain," in *Proc. Int. Conf. Comput., Power Commun. Technol. (GUCON)*, Sep. 2018, pp. 165–169.

[11] S. Suratkar, M. Shirole, and S. Bhirud, "Cryptocurrency wallet: A review," in *Proc. 4th Int. Conf. Comput., Commun. Signal Process. (ICCCSP)*, Sep. 2020, pp. 1–7.

[12] K. Karantias, "SoK: A taxonomy of cryptocurrency wallets," IOHK (Input Output Hong Kong), Cryptol. ePrint Archive, Hong Kong, White Paper 2020/868, 2020. [Online]. Available: https://eprint.iacr.org/2020/868

[13] I. Eyal, "On cryptocurrency wallet design," in *Proc. 3rd Int. Conf. Blockchain Econ., Secur. Protocols (Tokenomics)*, vol. 97, V. Gramoli, H. Halaburda, and R. Pass, Eds. Dagstuhl, Germany: Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2022, pp. 4:1–4:16. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2022/15901

[14] K. P. Jørgensen and R. Beck, "Universal wallets," *Bus. Inf. Syst. Eng.*, vol. 64, no. 1, pp. 115–125, Feb. 2022.

[15] A. M. Antonopoulos and G. Wood, *Mastering Ethereum: Building Smart Contracts and Dapps*. Newton, MA, USA: O'reilly Media, 2018.

[16] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2017.

[17] *Decentralized Identifiers (DIDs) V1.0*. Accessed: Mar. 2, 2023. [Online]. Available: https://www.w3.org/TR/did-core/

[18] *Verifiable Credentials Data Model V1.1*. Accessed: Mar. 2, 2023. [Online]. Available: https://www.w3.org/TR/vc-data-model/

[19] D. Fett, K. Yasuda, and B. Campbell. (Dec. 2022). *Selective Disclosure for JWTs (SD-JWT)*. Internet Engineering Task Force. Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/02/

[20] *The BBS Signature Scheme*. Accessed: Mar. 2, 2023. [Online]. Available: https://identity.foundation/bbs-signature/draft-irtf-cfrg-bbs-signatures.html

[21] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *J. Cryptol.*, vol. 7, no. 1, pp. 1–32, Dec. 1994.

[22] Š. Čučko, Š. Bećirović, A. Kamišalić, S. Mrdović, and M. Turkanović, "Towards the classification of self-sovereign identity properties," *IEEE Access*, vol. 10, pp. 88306–88329, 2022.

[23] *Mnemonic Code for Generating Deterministic Keys*. Accessed: Mar. 3, 2023. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

[24] A. Vázquez-Ingelmo, A. García-Holgado, and F. J. García-Peñalvo, "C4 model in a software engineering subject to ease the comprehension of UML and the software," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Apr. 2020, pp. 919–924.

[25] (2022). *Ethereum Wallet Metamask Passes 30M Users, Plans DAO and Token—Decrypt*. [Online]. Available: https://decrypt.co/95039/metamask-consensys-30-million-users

[26] *EIP-712: Ethereum Typed Structured Data Hashing and Signing*. Accessed: Mar. 3, 2023. [Online]. Available: https://eips.ethereum.org/EIPS/eip-712

[27] *EIP-191: Signed Data Standard*. Accessed: Mar. 6, 2023. [Online]. Available: https://eips.ethereum.org/EIPS/eip-191

[28] *EIP-4361: Sign-In with Ethereum*. Accessed: Mar. 6, 2023. [Online]. Available: https://eips.ethereum.org/EIPS/eip-4361

[29] M. A. Hassan and Z. Shukur, "Review of digital wallet requirements," in *Proc. Int. Conf. Cybersecurity (ICoCSec)*, Sep. 2019, pp. 43–48.

[30] S. Schwalm, D. Albrecht, and I. Alamillo, "eIDAS 2.0: Challenges, perspectives and proposals to avoid contradictions between eIDAS 2.0 and SSI," in *Open Identity Summit*, H. Roßnagel, C. H. Schunck, and S. Mödersheim, Eds. Bonn, Germany: Gesellschaft für Informatik, 2022, pp. 63–74.

[31] Š. Cucko and M. Turkanovic, "Decentralized and self-sovereign identity: Systematic mapping study," *IEEE Access*, vol. 9, pp. 139009–139027, 2021.

[32] *Issuers Trust Model—Accreditation Of Issuers EBSI Specifications*. Accessed: Mar. 6, 2023. [Online]. Available: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSIDOC/Issuers+trust+model+-+Accreditation+of+Issuers

[33] (2023). *The European Digital Identity Wallet Architecture and Reference Framework | Shaping Europe's Digital Future*. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework

[34] Y. K. Chaturvedi, (Jan. 2022). *A Quick Guide To Metamask Snaps*. [Online]. Available: https://etherworld.co/2022/01/19/a-quick-guide-to-metamask-snaps/

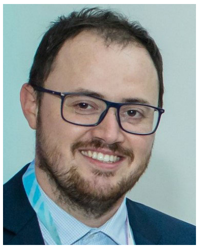
[35] A. Shostack, "Experiences threat modeling at Microsoft," *MODSEC@ MoDELS*, vol. 2008, p. 35, Sep. 2008.

VID KERSIC (Member, IEEE) is currently pursuing the Ph.D. degree with the Faculty of Electrical Engineering and Computer Science, University of Maribor (UM), Maribor, Slovenia. He is a member of the Research and Development Group, Blockchain Lab:UM. He is currently a Young Researcher with the Institute of Informatics. His current research interests include blockchain, DLTs, decentralized identity, and artificial intelligence.

URBAN VIDOVIC received the bachelor's degree in computer science from the Faculty of Electrical Engineering and Computer Science, University of Maribor, in 2020, where he is currently pursuing the master's degree in computer science. He is a Research and Development Engineer with the Blockchain Lab:UM, focusing on the Web3 field, which includes self-sovereign identity, blockchain, smart contracts, and data privacy.

ANDRAZ VRECKO received the bachelor's degree in computer science from the Faculty of Electrical Engineering and Computer Science, University of Maribor (UM), in 2021, where he is currently pursuing the master's degree in information technology and data science. He is a Research and Development Engineer with the Blockchain Lab:UM, focusing on the Web3 field, which includes self-sovereign identity, blockchain, and smart contracts.

MARTIN DOMAJNKO received the degree in computer science from the Faculty of Electrical Engineering and Computer Science, University of Maribor (UM FERI), in 2021, where he is currently pursuing the master's degree in computer science. He is also a full-stack Research and Development Engineer with the Blockchain Lab:UM, where he specializes in working with technologies, such as SSI, blockchain, and smart contracts, developing modern web solutions, and contributing to open-source projects.



MUHAMED TURKANOVIC (Member, IEEE) was the Managing Director and the CTO of an IT company, from 2013 to 2016. He has been an Associate Professor with the Faculty of Electrical Engineering and Computer Science, Institute of Informatics, University of Maribor (UM), Slovenia, since 2016. He is the Deputy Head of the Institute of Informatics and the Head of Research and Development of the Blockchain Lab:UM as well as the Head of the Slovene EDIH DIGI-SI, and a UM's Coordinator of several H2020, HORIZON, and/or DIGITAL projects. He has authored several highly cited research articles, received a patent from EPO, and edited several special issues of scientific journals. His current research interests include advanced digital identities, DLTs, database technologies, and applied cryptography.