Decentralized Identification and Information Exchange in Distributed, Blockchain-Based Internet Architectures: A Technology Review

Laura Maria Marques da Fonseca, Hamid R. Barzegar and Claus Pahl
Faculty of Engineering, Free University of Bozen-Bolzano, Bolzano, Italy

Keywords: Decentralized Identity, W3C DID Standard, Identity Management, Verifiable Credentials, Blockchain.

Abstract: In many Web and Internet-based systems, sharing Personally Identifiable Information (PII) to identify persons and other entities is common, but centralized systems such as central registries have limitations in terms of control of privacy and identity that a decentralized identity management architecture could address. This study aims to compare the current and potential systems, analyze protocols for decentralized identification and data exchange, propose a protocol selection method, and provide a simple code example. The goal is to assess the feasibility of decentralized processes in software-based business workflows. The methodology involves reviewing protocol materials, including white-papers, articles, and code docs, alongside ontological aspects of identification. Challenges to implementing Decentralized Identifiers (DIDs) include interoperability and the evolving Web/Internet landscape towards more decentralization, openness, and greater user utility.

1 INTRODUCTION

Automation has allowed to execute business tasks more efficiently and error-free than human actors since the 1970s. The great threat for such systems consists in the human misuse of their capacities or information, an issue often addressed by protecting either the access through an identification system, known as access control systems, or other personal identification information (Bowers, 1988). Access control is neither new as a procedure nor exclusive for computerized business environments. With the ubiquity of computer and internet-based tasks, the exchange of Personally Identifiable Information (PII) has immensely expanded in the last 20 years, which has raised awareness about privacy and identity ownership, problems tightly related to the centralized control of and access to the information. Today, decentralized variants of the identification and exchange of PII are proposed such as the W3C DID standard (W3C , 2023) that builds on the URI Uniform Resource Identifier mechanism of the Web, but with a mechanism that is relatively complex, which requires a thorough analysis.

The objectives of this research are to: (i) introduce the concepts of the Decentralized IDentifier (DID) and understand which problems it addresses; (ii) map the state-of-the-art of the protocols implementing the DID; (iii) offer a decision framework for the choice of one protocol for an application given its requirements; (iv) apply the decision framework to one sample decision process and suggest an implementation of the protocol chosen; and (v) provide arguments for the comprehension and discussion about the feasibility of implementing this technology (in its current state) for common identification and PII exchange processes.

The motivation is the concern regarding control and ownership (and their management) of personal identifying data by companies and other organisations in the Web space, which often misuse this data for financial and surveillance reasons. This motivation extends to a personal desire for an identification and data-exchanging processing where the data ownership is held by the data owner (Dodevski and Trajkovik, 2018). Assessing the feasibility of this new identification architecture is the final objective.

We outline our methodology in Section 2. Section 3 explains the decentralized identifier concept, Section 4 describes the construction and application of the decision framework, Section 5 evaluates the decision procedure, code implementation is explained in Section 6, and conclusions are presented in Section 7.

2 METHODOLOGY

We carry out a technology review of suitable protocols (Werth et al., 2023a), following a systematic

Marques da Fonseca, L., Barzegar, H. and Pahl, C.
Decentralized Identification and Information Exchange in Distributed, Blockchain-Based Internet Architectures: A Technology Review.
DOI: 10.5220/0012254900003584
In *Proceedings of the 19th International Conference on Web Information Systems and Technologies (WEBIST 2023)*, pages 535-544
ISBN: 978-989-758-672-9; ISSN: 2184-3252

selection and classification approach (Werth et al., 2023b). We use a sample implementation to discuss the feasibility of protocol implementation for concrete application scenarios. The approach taken involved studying the DID mechanism according to the W3C standard and exploring projects with decentralized ledger-based DID implementations. Here, identification and data exchange processes are closely related, with identification as the initial step followed by data provision (Fukami et al., 2021). We treat identification as distinct from data exchange. The research questions led to a decision framework methodology, incorporating objective criteria (Pahl et al., 2018).

The stages included evaluating protocols through an analysis of white-papers, blogs, social media, and web content to assess their alignment with the chosen criteria. Outcomes were organized into a decision matrix for protocol selection. An identification process was selected and the decision framework applied, leading to an illustrative protocol implementation. Reflections on decentralized app viability and DID challenges form the conclusion. Literature was selected from diverse sources, including blog posts and social media, recognizing that blockchain and other technology discussions often occur outside academia and its publications. The goal is a comprehensive exploration of emerging personally identifiable information (PII) exchange architectures.

- Question 1 concerns *Benefits and Limitations of Architectures for Identification and Data Exchange*. This requires an investigation of the authentication process's nature and its relationship with information systems. This question also addresses drawbacks of centralized identification and data exchange, aiming to identify improved processes to rectify them.
- Question 2 addresses the analysis of *Benefits and Limitations of the DID Concept and its Proposed Standard*. We examined DID privacy compliance, potential drawbacks, and the impact of DID adoption on digital business processes.
- Question 3 concerns the *Feasibility of Decision Frameworks and Protocol Implementations*. This explores the current state of DID implementation and enhancements, protocol selection strategies, feasibility, and strategies to address limitations in existing protocols.

These inquiries constitute an in-depth exploration of decentralized identification and data exchange systems. By systematically investigating these questions, this research aims to contribute to a comprehensive understanding of the implications, challenges, and potential benefits of emerging decentralized information exchange architectures.

3 THE DECENTRALIZED IDENTIFIER

The core of a decentralized identification system is the mechanism of the decentralized identifier (DID), as proposed by the World Wide Web Consortium (W3C). The understanding of the new identification paradigm implies a broader comprehension of the DID internal mechanism and its external effect on digital business processes.

Decentralized identity empowers users by removing centralized data storage, offering control over personal information for authentication and claims exchange. It employs three core elements: the Decentralized Identifier (DID), blockchain, and Verifiable Credential (VC). DID is a unique cryptographic global identifier. W3C standardized this system in late 2021 for decentralized, persistent, and cryptographically verifiable identity creation and resolution (Serto Suite Documentation, 2023),(Shilina, 2022).

3.1 DID Mechanism

The DID serves as a unique identifier, offering access to a specific value from a verifiable data registry -— an essential DID document `DIDstring: document`. This document contains details on the DID controller, cryptographic keys, and potential external references. The controller, acting as the document's key, has sole authority to modify it. Notably, the controller might differ from the DID subject (e.g., a CEO controlling a company's DID). Subjects can encompass individuals, institutions, data models, etc.

The W3C standard does not specify a preferred verifiable data registry, so for example blockchain is not mandated. However, for improved quality and security, an ideal registry should possess the following key attributes: (1) Immutability: prevent unauthorized identification changes. (2) Longevity: Ensure identity preservation. (3) Auditability: record every change. (4) Decentralization: align with decentralized identifier concept, avoiding central data control. (5) Single source of truth: A subject should be in a single ledger for data accuracy and non-repudiation. Blockchains address needs one to four but not the fifth. Blockchain innate properties, addressing points one to four, are not contingent on external data checks prior to processing and storage. Addressing point five forms part of our research focus here.

3.2 Components

The DID-based identification consists of three main interacting components: the controller, the document and the verifiable credentials.

DID Controller. To avoid confusion, the data structure DID will be named DID controller here. The DID is a self-sovereign, portable, verifiable, decentralized identity that lasts a lifetime and is implemented in the form of a string that must follow the following syntax:

```
    URI scheme identifier: identifier for the
DID method: DID method-specific identifier.
```

The URI scheme identifier is always a “did”. The method is a W3C-jargon for the different materializations of the verifiable data storage, which might be a generic one or custom-made. In this verifiable data storage, the DID document can be found. Btcr means the bitcoin blockchain, for example. The two first components of the DID string (uri+method) are fixed, the third part might be redefined by the method to adapt the string for the local information necessities. The sequential structure of the third part might be even changed for a sequence of strings that meet one given standard. Cosmos, for example, developed the DID further in four different versions to adapt it to its interoperable reality; a DID string that carries the zone would be adequate, e.g., for Cosmos:

```
cosmos:version:chainspace(test,mainnet):
namespace:unique-id.
```

The following DID corresponds to a DID document stored in the ”regen” testnet in the NFT “ecocredit” namespace (Andrieu et al., 2022):

```
did:cosmos:1:regentest:ecocredit:
1Kpg3KJPOIarthPWf8HHyy
```

DID Document. The document retrieved from the data registry via DID key is a DID document which contains data related to the DID subject, including cryptographic materials (keys) and possible external references, such as a reference to an alternative identification method, a service reference, amongst others. It is usually created/issued in one standard format (JSON or JSON-LD). The controller creates the DID document and could change it (if it is not stored in an immutable storage), whilst the subject is identified by it. In some cases, the subject and controller might be the same. Whether the DID standard allows changes to be made it is not clear (W3C, 2021). A sample DID document (Serto Suite Documentation, 2023) is:

```
"@context": "https://www.w3.org/ns/did/v1",
"id": "did:example:123456789abcdefghi",
"authentication": [
"id": "did:example:123456789abcdefghi#keys-1",
"type": "Ed25519VerificationKey2018",
"controller": "did:example:123456789abcdefghi",
"publicKeyBase58": "H3C2AVvLMv6gmMNam3u
VAjZpfkcJCwDwnZn6z3wXmqPV" ],
"service": [
// used to retrieve Verifiable Credentials
associated with the DID
"id":"did:example:123456789abcdefghi#vcs",
"type": "VerifiableCredentialService",
"serviceEndpoint": "https://example.com/vc/"]
```

Verifiable Credentials. The DID string is a mechanism for indexing a user or issuer public key in the blockchain, while the VC is a claim or attestation related to the DID subject. The DID is the means and the VC is the finality of the process: a cryptographically signed claim, whose legitimacy can be proven by a reverse cryptographic process using a public key stored in the blockchain. A VC can be a diploma, a birth certificate, the identification name of a user or any piece of information related to one digital/physical subject.

3.3 Drawbacks of the W3C Standard

According to our analysis, the most significant problem of the DID mechanism as designed by the W3C is the fact that it is not interoperable. The reasons for this mechanism design can be assumed as follows:

1. It was designed in 2020/2021 for the then heterogeneous and fragmented state of blockchains.
2. Interoperability was not a focus during the development, especially because the year before the release (2020) saw an explosion in market capitalization of tokens, and it seemed back then there would be enough users for the biggest networks.
3. The developers visualized the interoperability in DIDs would or should be better implemented through another mechanism in the blockchain software stack, for example with an API, a local logic of each application or a layer 2 or 3 solution.

3.4 W3C Standard and Blockchain

The DID standard as defined by the W3C can be implemented on the Web 2.0. The W3C documentation does not affirm that the key storage must be made in the blockchain. Nonetheless, it the DID implementation and expansion in number of protocols can be expected in a Web 3.0 iteration because:

1. Storing the keys outside an immutable ledger (blockchain) would threaten the non-repudiation of the communication that happens in name of that DID (alias the VCs).

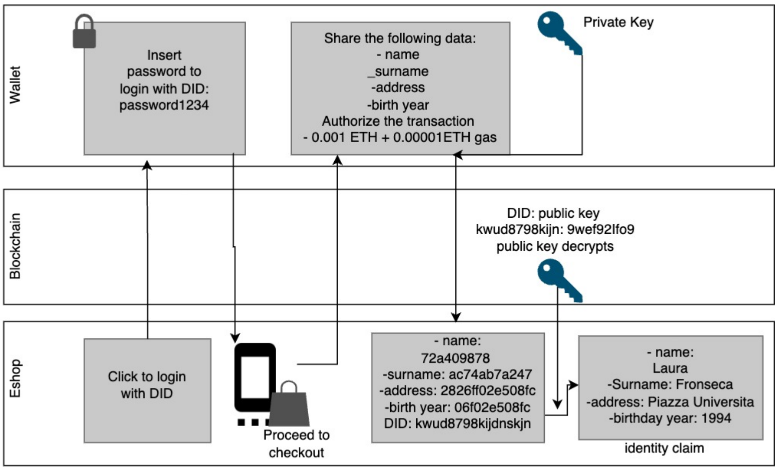


Figure 1: An E-commerce example.

2. According to our research, there is no implementation of the DID in the Web 2.0 currently in use.

3.5 Process for the User

Fig. 1 illustrates the on-boarding and checkout procedures implementing the DID and the VC. Firstly, the customer logs into the platform with the DID stored on the wallet. The customer proceeds to checkout and the website sends a request for the wallet of personal data such as name, age, and the payment gateway authorisation. The wallet formulates the request as a push-up notification for the customer who must accept or deny it. After the customer approval, the hash of the customer's personal data is sent in the form of VC together with the issuer's (the state's) DID string. With help of the DID string, the e-commerce application locates the issuer's public key in the blockchain and uses it to decrypt the hash of the customer's personal data, confirming his or her identity.

3.6 Process for Business/Software Actor

Fig. 2 shows a generic business process that implements the DID and the VC. Firstly, it is necessary to clarify some concepts implemented in the diagram.

The process is triggered with an identification request sent to the DID protocol. The DID protocol – most likely the wallet part – will formulate two requests: one for the login with password and another one for the authorization to transfer the data. If the user types the password wrongly, the password request is repeated, and if the user denies the transferring of data, the process is aborted at the application level. If both password and data transferring requests are successful, the process continues with the encrypting of the user's data in the level of the DID protocol with the private key of the issuer. In this same task, the data is encapsulated with the DID of the issuer, and after that it is sent to the application layer. The application layer receives this package with encrypted data and DID and searches in the blockchain for the key-value pair `DID:public key`. If the key-pair is not identified, the process terminates instantaneously, else the decryption of the data is executed next (the decryption is included in the sub process App Business Process). With the decrypted data, the decentralized application business process can be executed, and the process terminates in its successful path.

4 DECISION FRAMEWORK

A objective is the development of a decision framework, helping developers to select a protocol and blockchain platform for an identity management solution. This is based on a review and classification of protocols. After an initial state-of-the-art assessment, 17 protocols implementing the DID system as standardized by the W3C were found. We used a two-phased procedure for creating the decision framework together with a group of criteria and of questions for the first and the second phase, respectively. Furthermore, a systematic review of the official documentation and white papers related to each protocol was made in order to answer the questions and ascertain the previously established criteria. The decision framework contains a pre-defined number of steps that can assist the choice of one protocol whenever the business and identification requirements of the decentralized application are clearly defined. It is visually represented by the decision tree. Furthermore, for validation and illustration, a decentralized application with a prototypical identification need is

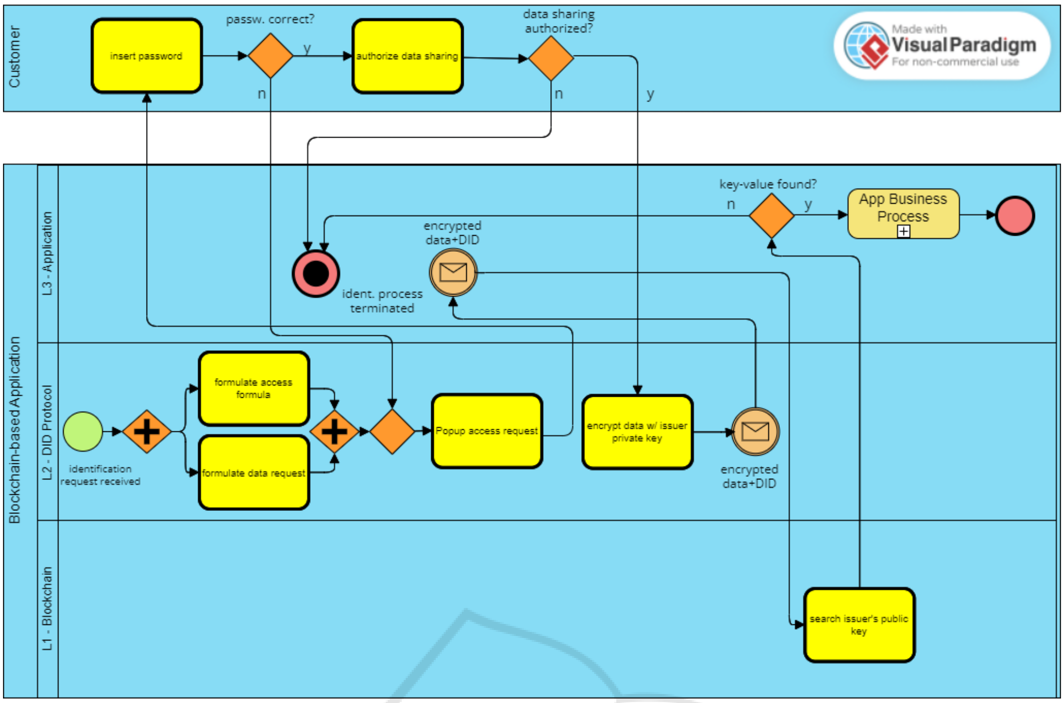


Figure 2: A decentralized identification in a generic business process.

Table 1: An excerpt from the original decision matrix.

Protocol/Criterion	Updated GitHub	Updated Website	Part Conferences	Wallet works
Serto	no	no	no	n.a.
civic.com	yes	yes	yes	no
dock.io	yes	yes	no	yes

specified, used as input for the decision framework and as a result one protocol is chosen.

4.1 DID-Implementing Protocol

A DID-implementing protocol consists of an API containing classes, objects and methods for procedures related to the DID identification and PII-exchanging mechanism, such as:

- Creating the DID string and the set of public and private keys.
- Storing the key-pair value DIDstring:public key in the blockchain in an indexed way.
- Indexing the same key-pair value at user wallet.
- Signing one VC and passing the combo [DIDstring of the issuer + hash/VC + claimed message] to another user
- Receiving the combo [DIDstring of the issuer + hash/VC + claimed message] and verifying the authenticity of the message

The latter two might be executed at application level, but the DID protocol must at least pass the combo `[ DIDstring of the issuer + hash/VC + claimed message ]` to the application, where it will be transmitted and/or received and decrypted.

4.2 Methodology and Construction

After an initial review of the selected projects' documentation, it became evident that some protocols needed exclusion due to obsolescence or other reasons. This led us to a two-phase approach: exclusion and inclusion. In phase one, unsuitable protocols were eliminated, while phase two categorized remaining ones based on their characteristics and relevance. Both phases involved systematic reviews of the available protocol materials, including white papers and official communications. Phase one aimed to address questions as follows: (i) criteria for a protocol's ineligibility in (D)app integration and (ii) ideal features when choosing a DID-implementing protocol. These questions yielded yes-no criteria, such as assessing wallet-based protocols' ease of DID creation and obsolescence. Phase two focused on protocol distinctiveness and relevance to business needs, using the following questions: (i) blockchain platform and its position in the scalability-security-decentralization trilemma and (ii) protocol interoperability.

The inclusion criteria were defined after phase one, taking into account the protocols left. Phase two aimed to align each protocol with specific business process needs or (D)app requirements, often necessitating sub-criteria for differentiation. Ultimately, this process facilitated the mapping of each protocol to unique business necessities or (D)app requirements.

The construction of the decision framework follows the two phases defined above.

4.3 Selection Criteria

We consider three exclusion criteria:

1. The Obsolescence of the Project: Given the recent W3C DID standard publication and the dynamic nature of the darea, constant protocol evolution is imperative. Software maintenance must consider factors like security enhancements and blockchain ecosystem adaptations. Project activity indicators include factors such as GitHub activity, roadmaps, team engagement in workshops, and social media updates. Ongoing observation clarified which protocols met this criterion.
2. Proprietary Software: Integrating proprietary solutions into open-source platforms poses licensing challenges. To avoid compliance issues, both app and DID framework should be from the same vendor. Framework integration assumes license compatibility (often lacking in proprietary software).
3. Testing of Wallet-Based Protocols: Some protocols integrate at the app layer, while others closely collaborate with wallets, acting as identity/password managers. Wallet-based protocols underwent guided tests for creating DIDs.

A first protocol division based on Ethereum compatibilty as one of the key platforms resulted in two exclusive groups: (1) Ethereum and Ethereum Virtual Machine (EVM) compatible; (2) interoperable and aggregating protocols.

Ethereum or EVM Compatible: Running on EVM-compatible blockchains widens user reach due to Ethereum's market share. EVM executes smart contracts, and Ethereum's popularity promises greater user interaction. Arguments for EVM-based protocols: (1) Dapp users engage with Ethereum's ecosystem. (2) Potential network effect favors protocols on widely-used blockchains. However, Ethereum lacks interoperability and faces scalability and price issues.

Subdivision criteria for the Ethereum category are the following: (1) Protocol is Ethereum-based. (2) Protocol is EVM-compatible with higher scalability: Binance Smart Chain-based protocol.

Interoperable and Aggregating Protocols: Original DID and blockchain designs lacked interoperability, causing issues like identity fragmentation. To address this, two imperfect approaches emerged: interoperability and aggregation. Interoperable protocols operate within an interoperable ecosystem, while aggregation handles data communication between non-compatible networks.

- Interoperable protocols solve logic transformation across networks, but they're limited to specific ecosystems like Polkadot and Cosmos. Here, there are no sub-criteria for interoperability protocols due to their ecosystem similarity.
- Aggregation protocols bridge data communication gaps between non-compatible networks but require manual logic translation. Sub-criteria for aggregation protocols are as follows: (1) Protocol has a blockchain of its own. (2) Protocol doesn't have its own blockchain.

4.4 Application of the Decision Criteria

In this part, the above defined decision process with a series of steps will be applied. For the first part, the exclusion criteria, it is not necessary to define a specific context. On the other hand, the second part with inclusion criteria requires the definition of a specific context, more concretely said, a contextualized decentralized application.

The initial list of DID-implementing protocols found was: Litentry, civic.com, kilt.io, ontology, selfkey, XSL, veramo.io, metadium, dock.io, Ceramic, Microsoft ION, blockpass, remme.io, validity.tec, synaps, IBM Blockchain Identity, serto

4.4.1 Exclusion Criteria

Obsoleteness and proprietary software were decided based on the documentation and resources available. The wallet decison required testing, which had the following results. Kilt.io successfully passed, offering clear instructions and easy DID access. Selfkey, however, caused problems as exhaustive research and app exploration failed to yield DID creation.

Thus, the initial protocol list was reduced by applying the exclusion criteria one-by-one:

1. After excluding the obsolete projects, the list reduces to: Litentry, civic.com, kilt.io, selfkey, XSL, metadium, dock.io, ceramic, IBM Blockchain Identity.

2. After excluding the proprietary software-based: Litentry, civic.com, kilt.io, selfkey, XSL, metadium, dock.io, Ceramic.
3. After excluding the non-functioning wallet-based tested protocols: Litentry, kilt.io, XSL, metadium, dock.io, Ceramic.

4.4.2 Inclusion Criteria

The classification of protocols following the interoperability criteria is presented in Table 2. Table 3 shows a discussion of interoperability concerns.

To allow a technical comparison, we also provide an analysis of the blockchain throughput in Table 4.

We already said, that the for second phase, the application context plays a role (Dodevski and Trajkovik, 2018; Guggenberger et al., 2020). In this phase, application requirements guide our choice of an **e-voting Dapp**, serving as an example of a web 3.0 service needing an identification system. Such criteria extend to other web 3.0 services reliant on authentication, including decentralized exchanges, DeFi lending, DAOs, and voting systems. The DID utility extends to future DeFi services like credit rating creation for borrowers. The e-voting app targets specific contexts, such as city council representative elections.

Interoperability and aggregation concerns for the chosen app context are: (1) Ethereum + EVM compatibility or Aggregation + Interoperability: Interoperability (2) Interoperability or Aggregation: Given Polkadot-centric interoperability, aggregation is preferred. (3) Aggregation Protocol with or without its own blockchain: Litentry's own blockchain suits exclusive DID management or Dapp hosting.

In the e-voting context, the DID serves a utilitarian role, authenticating users and validating votes. The selected protocol should interact with diverse blockchains for adaptable e-voting implementations, making an aggregation protocol without a dedicated blockchain, like Ceramic, a fitting choice.

The integrity of the decision can be validated either by proving the quality of the decision procedure or through illustration by implementing the chosen protocol (Ceramic for the given context) as a baseline identification architecture in one production-case of the exemplary business process (e-voting) and testing. Given that a creation, implementation, and testing of do not lie within the scope of this project, greater diligence will be put into the evaluation of the correctness and completion of the decision protocol. Nevertheless, the implementation of one illustrative DID as identification baseline is provided later with the aim of clarifying the achievability of this implementation.

4.5 Decision Tree

A decision tree is a representation of the steps of the decision process. It acts here as a summary of the framework created after the definition of the decision criteria. The aim was to differentiate the protocols as much as possible, such that the decision framework would fit the largest spectrum of decentralized projects (and their requirements). The consequence of this work is the fact that the tree has leaf nodes with only one protocol, except for kilt.io and dock.io. A visualization of the tree can be found in Fig. 3.

5 DECISION PROCEDURE EVALUATION

The protocol's feasibility was assessed in the first half of 2023, considering digital identification systems, business processes, and current DID-implementing protocols. Characteristics were evaluated uniformly, without arbitrary prioritization due to undefined application domains. A specific concern is the up-to-dateness and extensibility of the decision framework. The extensible decision framework, representing accumulated knowledge, suits any cognitive identification process requiring PII exchange in digital business. The protocol remains viable unless: (1) new DID protocol enters production; (2) a listed protocol exits the market; (3) major disruptions affect base elements; (4) the blockchain coins' market capitalization is disrupted; (5) key digital economy actors implement protocols. Adapting the framework for item 1 or 2 is simple, while other scenarios demand a full decision procedure reformulation.

Currently, a few functional DID-implementing protocols exist after the 2022 Crypto crisis, which is reflected in the current list. Larger ecosystems like Polygon and Cosmos lack DID protocols. Ceramic stands out, sharing DID across blockchains as an application layer solution.

6 DID SYSTEM IMPLEMENTATION

The scope of this implementation consists of creating a functional decentralized application for the creation of a DID, which string ought to be posted in the blockchain as an index to the public key. The objective is providing an implementation and deployment in order to investigate viability, feasibility and ease of integration into any digital business process.

Table 2: Protocols divided following defined interoperability criteria and sub-criteria.

	Interoperable or Aggregating	Interoperable or Aggregating
Ethereum + EVM Compatible	Interoperable	Aggregating
Ethereum: Metadium more scalable: XSL (Binance smart chain)	Polkadot: kilt.io, dock.io	With a blockchain of its own: Litentry
		Without a blockchain of its own: Ceramic API

Table 3: Pros and Cons of Interoperability-based and Aggregation-based protocols.

	Interoperability	Aggregation
pros	Prefabricated logic, one DID enough	Network effect, increase scalability
cons	Protocols are incipient, no real interoperability within big chains	Protocols are middlemen APIs. No token of its own (exception Litentry). Potential translation issues

Table 4: The Throughput of the blockchains that have active DID protocols, measured in TPS (transactions per second).

Blockchain	Maximum TPS	Average TPS
Ethereum	20 (Blockchair Blockchain Services, 2023)	13-15 (Blockchair Blockchain Services, 2023)
Binance Smart Chain	300 (BSC Binance Smart Chain News, 2023)	40 (BNB Smart Chain Explorer,)
Polkadot	1,000 (Coinbase Documentation, 2023)	Information not found

The initial intention was to provide an interoperable implementation with Ceramic, based on the decision process and documented experience with a previous implementation provided on GitHub (Dabit, 2023). Nevertheless, due to continuously changing APIs, an Ethereum-based implementation was provided to answer these questions: (1) What is the feasibility of the implementation of protocol for managing DIDs? (2) What are the drawbacks of the DID-implementing protocols? (How) Could they be improved? We report on problems that arose during the implementation that would limit feasibility and/or ease of implementation.

6.1 Ceramic Protocol

The Ceramic Network is a group of open-source APIs for storing, updating and retrieving data from decentralized networks (Ceramic, 2023), that has three main usability branches, amongst which is the support to the creation of apps with interoperable DID storage and communication. The way in which interoperability is implemented and operated in Ceramic can be seen in Fig. 4. One of Ceramic's products is named Decentralized Identifier and provides the client with methods for creating and retrieving DIDs, which is interoperable with seven different blockchains, amongst which Cosmos, Ethereum, Filecoin and Polkadot (DID Toolkit, 2023). This interoperability can be easily implemented in code with a switch-case structure at the moment of creating a data profile that accumulates the DID data and interacts with the blockchain.

Furthermore, Ceramic can interact with already existing wallets – of which most importantly Metamask – which avoids the situation in which the user must download a wallet and manage its credentials for each DID created (Ceramic Developers, 2023). Another relevant reason for implementing APIs from Ceramic is that the JavaScript DID is coded in TypeScript, which makes its implementation and integration into other code structures less challenging compared to other blockchain-related APIs, where interaction happens mostly in Rust, a recently developed and not so largely diffused language.

6.2 Initial Implementation

The objective is to build an interface for Ceramic for interacting with Ethereum to read or create a DID reference. Using TypeScript, the main Ceramic protocol functions like 3id-did-resolver (Ceramic Network, 2023) influenced a React app design. The app is divided into front-end (React App, JavaScript) and back-end (Ceramic APIs, blockchain). Initially, code was based on a tutorial, but libraries 3ID and IDX were replaced by '@didtools/pkh-ethereum' and 'composeDB' (ComposeDB on Ceramic, 2023). ComposeDBonly works in Linux IDEs. Issues arose with the DID class of the 'dids' package, preventing successful implementation. This class creates a skeleton DID structure, authenticates sessions, and connects to Ceramic's endpoint client. The skeleton can be replaced with actual data from the IDX library.

6.3 Second Implementation

The aim is creating an application that writes a DID on the blockchain. Given that the data con-

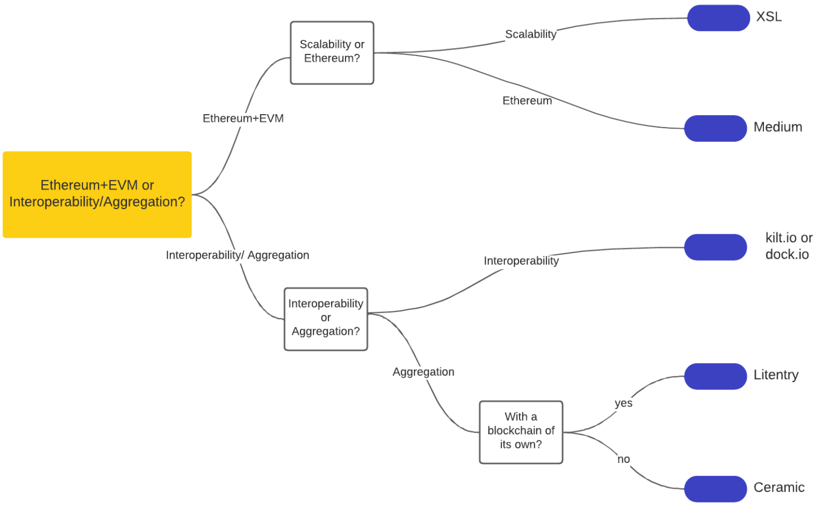
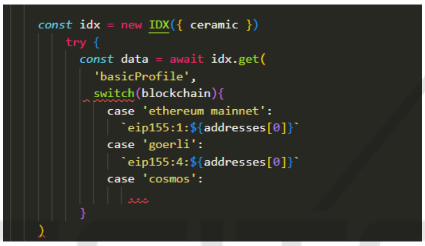
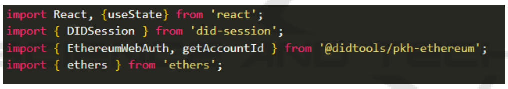

Ethereum+EVM or Interoperability/Aggregation?
Ethereum+EVM
Scalability or Ethereum?
Scalability
XSL
Ethereum
Medium
Interoperability/ Aggregation
Interoperability or Aggregation?
Interoperability
kilt.io or dock.io
Aggregation
With a blockchain of its own?
yes
Litentry
no
Ceramic

Figure 3: The Decision Tree for Protocol Selection for E-Voting.



```
const idx = new IDX({ ceramic })
  try {
    const data = await idx.get(
      'basicProfile',
      switch(blockchain){
        case 'ethereum mainnet':
          `eip155:1:${addresses[0]}`
        case 'goerli':
          `eip155:4:${addresses[0]}`
        case 'cosmos':
            ...
    }
)
```

Figure 4: Example of Interoperability in Ceramic.

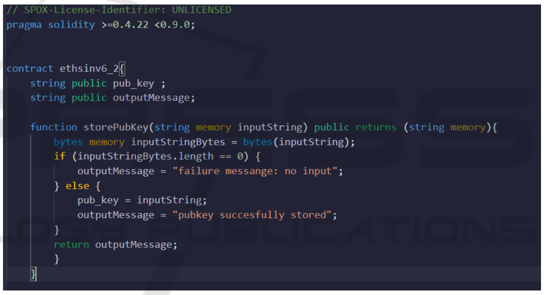


```
import React, {useState} from 'react';
import { DIDSession } from 'did-session';
import { EthereumWebAuth, getAccountId } from '@didtools/pkh-ethereum';
import { ethers } from 'ethers';
```

Figure 5: The Libraries Implemented for the DID-Dapp.

cept of DID is not provided by any JavaScript Ethereum library, it must be created within the application. Fig. 5 shows the libraries used for this implementation. The DID is a mechanism for the indexing of the public key in the blockchain, and that the DID string is this index. The most efficient option for indexing information on Ethereum lies in the transaction hash, which will concatenated with the method to create a DID string such as `did:ethr:0x988d147855e7d02081631d0c0416e960d8139b4e6a657c45822a1a5718bec36d`. The public key stored in the blockchain is the wallet address.

The React Native baseline was maintained, because the JavaScript library 'ethers' (Ethers Ethereum Library, 2023) allows interaction with Ethereum mainchain and testchain (Goerli). Fig. 6 displays the smart contract with which method the application interacted. Whether the interaction is made with the mainchain or in the testchain depends on the user's configuration of the wallet and on the location of the smart contract to interact with. Two functioning Ceramic libraries are implemented ('`did-session`' and '`@didtools/pkh-ethereum`') to collect the user's authorization for the session and for the access to wallet funds in the form of a wallet pop-up.



```
// SPDX-License-Identifier: UNLICENSED
pragma solidity >=0.4.22 <0.9.0;


contract ethsinv6_2{
    string public pub_key ;
    string public outputMessage;

    function storePubKey(string memory inputString) public returns (string memory){
        bytes memory inputStringBytes = bytes(inputString);
        if (inputStringBytes.length == 0) {
            outputMessage = "failure messange: no input";
        } else {
            pub_key = inputString;
            outputMessage = "pubkey succesfully stored";
        }
        return outputMessage;
        }
    }
```

Figure 6: The Smart Contract in Solidity.

A React Native application is created, followed by a smart contract in Solidity (here in the Remix online IDE) to receive a string input (public key) and write it to the blockchain. This contract must be deployed only once, and must be copied into the JavaScript code. Based on the ABI and contract address, the JavaScript code can access the method `storePubKey()` for posting the public key and passed the previously collected wallet address as parameter.

7 CONCLUSIONS

The feasibility of decentralized data exchange architectures raises concerns beyond technological issues. While this research focused primarily on the technical aspects, several crucial factors require attention.

In evaluating the potential for decentralized identification to consolidate, factors beyond privacy compliance need an analysis. Economic, legal, and sociological influences hold considerable sway. The combination of high entry costs and network effects creates a capital risk that could discourage participation from both identity providers and business actors, potentially leading to an oligopolistic market lacking user choice. This underscores the necessity of well-designed technical elements to prevent data misuse.

Market-driven forces further complicate matters. Sustainable business models for providers and incentives for user adoption remain currently unanswered. The transition from centralized routines to decentralized ones relies on user willingness to migrate from familiar systems. Moreover, third-party actors need compelling reasons to embrace these changes.

Human behavior and perceptions play a vital role. Encouraging users to trust information to decentralized identifiers, while motivating them to transition from established systems, presents challenges (Le et al., 2022). Additionally, addressing user awareness and engagement remains vital for success of the DID concept.

REFERENCES

Andrieu, J., Spies, L., and Conway, S. (2022). did:cosmos method specification. https://github.com/EarthProgram/did-cosmos/blob/main/README.md.

Blockchair Blockchain Services (2023). Ethereum transactions per second chart. https://blockchair.com/ethereum/charts/transactions-per-second. [accessed Sept 05, 2023].

BNB Smart Chain Explorer. Binance smart chain log and statistics (average tps). https://bscscan.com/. [accessed Sept 05, 2023].

Bowers, D. M. (1988). Principles of access control and personal identification systems. In Bowers, D. M., editor, *Access Control and Personal Identification Systems*. Butterworth-Heinemann.

BSC Binance Smart Chain News (2023). Binance, bnb chain transaction rates (tps). https://bsc.news/post/bnb-chain-is-the-go-to-blockchain-for-developers-why-is-it-special. [accessed Sept 05, 2023].

Ceramic (2023). Ceramic – the composable data network. https://ceramic.network/. [accessed Sept 05, 2023].

Ceramic Developers (2023). Why ceramic? https://developers.ceramic.network/learn/features/. [accessed Sept 05, 2023].

Ceramic Network (2023). Ceramic github project. https://github.com/ceramicnetwork/js-ceramic/blob/develop/packages/3id-did-resolver/src/index.ts. [accessed Sept 05, 2023].

Coinbase Documentation (2023). What is polkadot. https://www.coinbase.com/learn/crypto-basics/what-is-polkadot. [accessed Sept 05, 2023].

ComposeDB on Ceramic (2023). Composedb docs. https://composedb.js.org/docs/0.4.x/introduction. [accessed Sept 05, 2023].

Dabit, N. (2023). Sign in with ethereum & decentralized identity with ceramic, idx, react, and 3id connect, github repository. https://gist.github.com/dabit3/ba326e47e4882073bd6342dc1998fd16. [accessed Sept 05, 2023].

DID Toolkit (2023). Add support for a new blockchain. https://did.js.org/docs/guides/add-chain-support/. [accessed Sept 05, 2023].

Dodevski, Z. and Trajkovik, V. (2018). Decentralizing the health information exchange systems-a blockchain based approach. In *15th Intl Conf on Informatics and Information Technologies*, pages 43–49.

Ethers Ethereum Library (2023). Ethers v6.7.1 documentation. https://docs.ethers.org/v6/. [accessed Sept 05, 2023].

Fukami, Y., Shimizu, T., and Matsushima, H. (2021). The impact of decentralized identity architecture on data exchange. In *Big Data*, pages 3461–3465.

Guggenberger, T., Schweizer, A., and Urbach, N. (2020). Improving interorganizational information sharing for vendor managed inventory: Toward a decentralized information hub using blockchain technology. *IEEE Trans on Engineering Mgnt*, 67(4):1074–1085.

Le, V. T., Ioini, N. E., Barzegar, H. R., and Pahl, C. (2022). Trust management for service migration in multi-access edge computing environments. *Comput. Commun.*, 194:167–179.

Pahl, C., Ioini, N. E., and Helmer, S. (2018). A decision framework for blockchain platforms for iot and edge computing. In *3rd Intl Conf on Internet of Things, Big Data and Security, IoTBDS*, pages 105–113.

Serto Suite Documentation (2023). What is a did? https://docs.serto.id/docs/main-concepts/dids/. [accessed Sept 05, 2023].

Shilina, S. (2022). What is decentralized identity in blockchain? *Coin Telegraph*. [accessed Sept 05, 2023].

W3C (2023). World wide web consortium: Decentralized identifiers (dids) v1.0 - core architecture, data model, and representations. https://www.w3.org/TR/did-core/. [accessed Sept 05, 2023].

W3C (2021). World wide web consortium - meeting minutes 2021-03-30. https://www.w3.org/2019/did-wg/Meetings/Minutes/2021-03-30-did#section5-8. [accessed Sept 05, 2023].

Werth, J., Berenjestanaki, M. H., Barzegar, H. R., Ioini, N. E., and Pahl, C. (2023a). A review of blockchain platforms based on the scalability, security and decentralization trilemma. In *25th Intl Conf on Enterprise Information Systems, ICEIS*, pages 146–155.

Werth, J., Ioini, N. E., Berenjestanaki, M. H., Barzegar, H. R., and Pahl, C. (2023b). A platform selection framework for blockchain-based software systems based on the blockchain trilemma. In *18th Intl Conf on Evaluation of Novel Approaches to Software Engineering, ENASE*, pages 362–371.