2025 7th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS) | 979-8-3315-5982-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/BRAINS67003.2025.11302904

Demo: Gryphon Digital Identity Management System with Hyperledger Fabric, Compatible with Decentralized Identifiers and Verifiable Credentials*

Alessandro Neri, Iulian Onea, Matei Pânzariu, Mihai Frânculescu, Roland Kromes and Zekeriya Erkin
Delft University of Technology Departments of Intelligent Systems and Software Technology
Delft, The Netherlands
{a.neri, i.m.onea, m.a.panzariu, m.f.franculescu}@student.tudelft.nl, {r.g.kromes, z.erkin}@tudelft.nl

***Abstract*—Employing online identity management technologies and the use of blockchain capabilities, Gryphon is aimed at providing a decentralized Digital Identity Management System that securely handles user data by using Hyperledger Fabric. By introducing *Trustchain*, the system enables the verification of user credentials through modular components, facilitating streamlined and privacy-preserving communication between parties that require mutual data exchange. Gryphon is among the first platforms to implement this form of identity communication using the Hyperledger Fabric framework, demonstrating its viability as a foundation for decentralized identity management.**

***Index Terms*—Blockchain, Hyperledger Fabric, digital identity, verifiable credentials, Trustchain.**

I. Introduction

With recent technological advancements, essential services such as healthcare, finance, and government have become increasingly digital. However, many institutions still rely on centralized systems to manage sensitive data, introducing two main problems.

First, users often lack control over their personal data. Central authorities may use, share, or sell information without explicit consent or transparency. Despite regulations, enforcement can be weak, as illustrated by TikTok's recent EU fine for illegal data transfers to China [1], which put users at risk of being spied on.

Second, centralization creates a single point of failure. If the system is compromised by cyberattacks, malfunctions, or human error, all critical data are at risk. SQ Magazine reports that in the first half of 2025 alone, over 8,200 data breaches occurred worldwide, with the average cost reaching 4.96 million dollars [2].

To address these issues, we propose *Gryphon*, a decentralized identity management framework based on Decentralized Identifiers (DIDs) [3] and Verifiable Credentials (VCs) [4], implemented using Hyperledger Fabric [5]. Gryphon uses the *Trustchain*, an abstract structure that ensures the validity of shared information. Our implementation follows W3C standards and includes the test network provided by Hyperledger Fabric, though it has current limitations, such as default DID types (did:hlf) and the absence of key rotation which reduces security.

The rest of this paper is organized as follows: Section II describes the system architecture, Section III evaluates its performance, and Section IV provides additional materials.

* This work is supported by the European Union's Horizon Europe research and innovation programme under grant agreement No. 101094901 Septon and 101168490, Recitals Projects.

II. Features of Gryphon System

A. The Digital Identity Framework

Consider a university: Koen has a student card that identifies him as a student. The university issue him different credentials during his college years, among which is his diploma. Koen can then use these credentials when talking to companies and rely on the credibility of TU Delft, which is entrusted to issue diplomas by the Ministry of Education. In the digital space, student cards are represented by DIDs, and Koen's diploma and other credentials are represented by VCs. Therefore, to facilitate this example and others, the core of Gryphon will be the Digital Identity Framework, consisting of DIDs, VCs, and the systems in place for interacting with them.

DIDs are a standard that defines unique strings of characters that serve as decentralized identifiers. The standard also describes a resolution method that, given a DID, returns a DID document, a JSON document holding public information about the subject of the DID, such as the public key. These DID - DID document mappings are stored in a VDR (Verifiable Data Registry). There are many different resolution methods, and the one Gryphon uses is a custom one, "did:hlf", based on Hyperledger Fabric. Finally, while the standard for DIDs mentions many fields that can be part of the JSON document, one field took special consideration, namely the service field, which Gryphon uses to store the URLs to public registries.

Verifiable Credentials are a set of claims, packaged together with information about the issuer and the issuee, and a cryptographic signature that can be verified to prove the document is authentic. Therefore, these VCs serve to prove that user A made claims about user B. To prove that user A is trustworthy, they need to provide a public registry that provides the credentials that accredit user A as trustworthy. These public registries need to be hosted by each user who

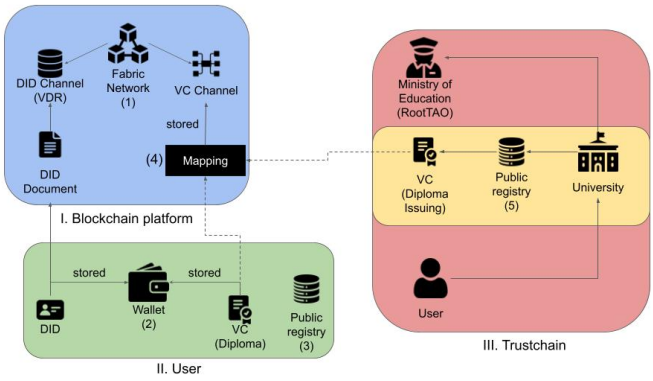

Fig. 1. Diagram showing the Architecture of the System

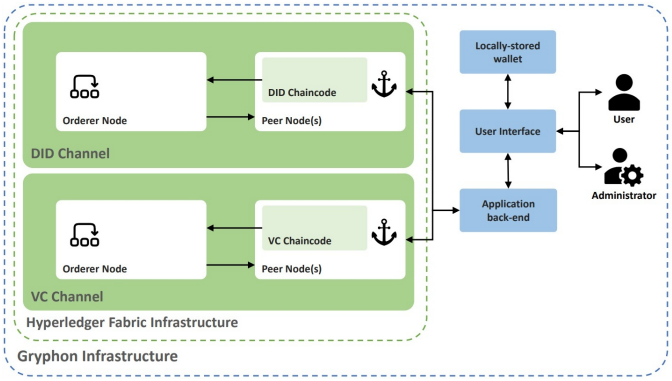

Fig. 2. Diagram showing the Implemented Components of the System

wants to issue VCs, and they provide the URL to that registry in their DID service field.

B. Architecture

Gryphon consists of two main components, the Hyperledger Fabric blockchain Network and the wallet, which is stored on the user's device. In the following example, the user is Koen, a student at TU Delft. Note that we refer to the Hyperledger Fabric blockchain network as the "Fabric Network" for the rest of the paper.

Fabric Network (Figure 1 1) consists of two channels: the DID channel and the VC channel. Channels in Hyperledger Fabric blockchain can be seen as separate ledgers with dedicated policies to store data. The DID channel acts as the VDR that maps a DID to its DID Document. The VC channel keeps track of the authorization a user would need to issue certain VCs. In the example, since TU Delft wants to issue a diploma to Koen, the university will require a "Diploma Issuing" VC. On the VC channel, the mapping "Diploma:Diploma Issuing" will be stored (note: there is no template for the type name, the names are decided by the admin of the application).

The User has access to two components: the wallet (Figure 1 2), and optionally a public registry (Figure 1 3). It is important to note that, all the parties involved in the application (a student, a university etc.) are treated as users, meaning that they all have their own wallet and public registry. The wallet stores DIDs, which can be used to verify the user's identity, e.g., student identity. The wallet also contains VCs corresponding to certifications, such as a diploma. The VCs are shared to prove certain authorizations, for example, only computer science students can access high-performance computing units.

The public registry is a necessary component for issuing and verifying VCs. If an entity, such as the University, but also a user (student) wants to issue Verifiable Credentials, they would need to set up a public registry, which displays the VCs that authorize the user to issue other VCs. In the example, TU Delft would have to store the Diploma Issuing VC in its public registry. This VC will allow issuing diplomas for students. To be more explicit, when a VC is issued to certify the diploma, the VC will contain a field referring to the issuer's DID and the type of the VC. Now, a verifier entity will first have to check if the VC was signed by the issuer and then, if the issuer corresponding to the DID has the necessary permission to issue diplomas. To this end, the verifier must check if the DID issuer holds a VC allowing the diploma issuance, which is stored in the public registry of the issuer. To check that the type of VC is the required one, the verifier queries the blockchain to find out that a VC of type Diploma requires a valid VC of type Diploma Issuing, which the university should hold in this example (Figure 1 4). To check this, the public registry of the university is addressed. If the necessary VC is not present in the public registry, the Diploma is considered invalid. Otherwise, the Diploma Issuing VC needs to be verified, by following the same process of checking the signature and addressing the public registry. For the VC of the university, the issuer is the Ministry of Education, which, in this case, is the RootTAO, an inherently trusted authority. If the Diploma Issuing VC is valid, and issued by the Ministry of education, the process is concluded, and the Diploma of the user is considered valid.

The above mentioned, verification structure forms a chain of trust, which is entitled **Trustchain**.

C. Implementation

The implementation of Gryphon is illustrated in Figure 2. This Digital Identity Management system uses a permissioned blockchain network, built with Hyperledger Fabric. For this Proof-of-Concept, the 'Fabric test network', provided by the "LF Decentralized Trust", has been used. This network consists of two peer nodes and an orderer node. The Hyperledger Fabric channel architecture enables transactions and communications between participants within isolated channels, each channel has its own underlying blockchain, maintained by participating peers.

On top of the given test network, two channels have been created: the DID channel and the VC channel. The DID channel's blockchain is used as VDR for DID related operations, while the VC channel is where administrators define mappings from VC types to required authorizations, which are then queried by users to verify the validity of an existing VC.

Transactions that can be executed on each channel are defined by smart contracts installed on participating peers (packaged as "DID chaincode" and "VC chaincode"). The "DID chaincode" allows transactions such as: storing a mapping from DID to DID document on the blockchain, retrieving the DID document of an existing DID from the blockchain, updating values of a DID document, and revoking the validity of a DID. The "VC chaincode" allows storing mappings from a VC type to the required authorization and querying the authorization of a VC type; for example, a valid mapping could look like this: "Bachelor Diploma" maps to "Diploma Issuer", which means that a valid VC of type "Bachelor Diploma", must be issued by an entity (identified by their DID) that owns a valid VC of type "Diploma Issuer".

The application can be accessed by registered and authenticated users as well as administrators. Administrators are responsible for defining VC mappings and establishing the Root TAO of the Trustchain, which in our example is the Ministry of Education.

Users can register on the system to obtain access to their personal wallet, from which users can generate their own personal DIDs and subsequently have VCs issued to their DIDs by other authorized users or organizations. Registered users can also request verification of a given VC, to make sure that an entity holds the credentials they claim to own.

The entire system is built with JavaScript-based languages such as TypeScript, Vue.js, and Node.js. More specifically, the user interface has been built with Vue.js, except for the wallet component, which required more type safety to handle cryptographic operations such as encrypting and decrypting the locally stored wallet, and, therefore, it is built with TypeScript. The front-end communicates with the Node.js backend, which, in turn, handles communication with the Hyperledger Fabric Network and maintains the user sessions using JSON Web Tokens (JWT). Lastly, the chaincode installed on the network peers is written in TypeScript.

Cryptography-wise, wallet encryption and decryption are ensured by the AES-GCM 256 algorithm. For the generation of public and private key pairs and their usage for signing and verifying VCs, ECDSA is used.

III. Maturity of Gryphon

The following experiment is designed to demonstrate the largest latency of the Gryphon system. As such, it will focus on the most time and resource intensive part of the application, which is the VC Verification. To perform the experiment, each component of the application including the Hyperledger Fabric blockchain infrastructure was performed locally, using Docker, on a 13th Gen Intel Core i7-13700H CPU with 14 cores at 2.4 GHz frequency, and with access to 16 GB of RAM. It must be noted, that the VDR to store the DID documents and the public registry of each user were stored in a local data base. In addition, the blockchain infrastructure contained two peers with default settings.

In this evaluation, the Trustchain consisted of a root user and 6 regular users. The root issued a VC to user 1, who then issued a VC to user 2, and this process was repeated for all users, ensuring that there is a VC at each level of the Trustchain. As we mention in Section II-B, VC Verification is performed by first checking the cryptographic validity of the provided VC, then finding the VC of the issuer and checking its' validity, and repeating this process until the root is reached. Therefore, the process of verifying a VC involves traversing up the Trustchain abstract data structure and performing the validity check operation for the VC at each level, so it stands to reason that the duration of verification would scale with the Trustchain level of the verified VC.

TABLE I
Duration of verifying a VC at different levels of the Trustchain

VC Level	Average Duration(ms)
0 (root)	5.0433
1	34.9012
2	59.6411
3	83.2173
4	99.1482
5	114.6152
6	134.194

Table I shows the latency for verifying VCs. As expected, the latency starts small and is roughly linear with the level of the VC in the Trustchain, which means that the system likely performs fast enough for users to have a good experience.

IV. Additional Materials

This article presents the Gryphona digital identity management system using the Hyperledger Fabric blockchain. Our system enables the identification of entities and individuals via their digital identities (DIDs) while respecting their privacy. The privacy is guaranteed since the DIDs do not contain any privacy sensitive information. In addition, the issuance of identifiers by entities is highly reliable, as verifiable credentials (VCs) and DIDs use digital signatures that follow a hierarchical order for issuing VCs and DIDs. A demo video illustrating the main features and workflow of Gryphon is available at https://www.youtube.com/watch?v=_MIMSX_yqOI. In addition, the source code, and documentation are hosted at https://github.com/Iron-Trust/Gryphon.

References

[1] K. Chan, "Tiktok fined $600 million for china data transfers that broke eu privacy rules," 2025, last accessed 16 September 2025. [Online]. Available: https://apnews.com/article/tiktok-ireland-european-union-data-privacy-regulation-d386ec74becc716905d7f686d6a448e2

[2] R. Namase, "Data breach statistics 2025: Key trends, costs & risks revealed," 2025, last accessed 16 September 2025. [Online]. Available: https://sqmagazine.co.uk/data-breach-statistics/

[3] World Wide Web Consortium (W3C), "Decentralized Identifiers (DIDs) v1.0," 2025, last accessed 16 September 2025. [Online]. Available: https://www.w3.org/TR/did-1.0/

[4] World Wide Web Consortium(W3C), "Verifiable Credentials Data Model v1.1," 2025, last accessed 16 September 2025. [Online]. Available: https://www.w3.org/TR/vc-data-model/

[5] Hyperledger Fabric Contributors, "Hyperledger fabric documentation," 2025, last accessed 16 September 2025. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/