Monas: Privacy, Data Interoperability, and Self-Sovereignty in Decentralized Personal Data Store v1.0

※ This document is a draft and will be updated.

Abstract

Monas aims to enhance data interoperability and user data sovereignty through a Decentralized Personal Data Store (PDS). This system leverages the encrypted data structure of Cryptree and the authenticity-guaranteeing technology of Blockchain as its core, allowing users to manage their own data while ensuring privacy and interoperability. Unlike traditional "solid-line data links," Monas introduces a new paradigm of data management through "dotted-line data links," directly reflecting the user's intentions. This approach creates a space in cyberspace that protects autonomy and privacy.

1. Introduction

The current issue of data silos and the utilization of data by platforms and AI without compromising privacy is well recognized. Cryptographic technologies such as the Semantic Web, Blockchain, and distributed storage are making strides towards enhancing data interoperability and establishing mechanisms where third parties cannot interfere with the value possessed by users. For instance, Linked data aims to enhance data interoperability by linking different data sources together. Moreover, technologies like Blockchain and IPFS make unauthorized editing or tampering of data by unauthorized third parties difficult, allowing for the verification of actions executed on the Blockchain. This serves a role in providing a trusted, neutral stance. However, the concepts and technologies mentioned above are predicated on public disclosure, which is not suitable for storing personal data, lacking in privacy and autonomy. The Lit Protocol, which utilizes secret sharing on a distributed network, addresses the issue of public nature of data but requires meeting multiple conditions for sharing a single piece of data. In essence, having one key per key and sharing multiple data points is akin to sharing multiple keys, lacking flexibility (for instance, the Lit protocol was used in the development of a data sharing system: Pipele).

Therefore, through the concept of a decentralized Personal Data Store named Monas, we aim to simultaneously realize both data interoperability and privacy. Monas builds an encrypted data structure called Cryptree, allowing users intuitive and flexible access control over their own space according to their will. This encompasses deciding where the data is stored, who can read or write the data, and how long these operations can continue. Additionally, by leveraging Blockchain for PDS state management, we address the consistency issues previously noted in existing PDS and decentralized SNS platforms like Solid, Personium, and Nostr.

Monas's approach signifies a paradigm shift from traditional "solid-line data links" to "dotted-line data links," illustrating the ability to convert dotted lines into solid lines based on user intent. This creates a space controlled solely by the user, realizing both data interoperability and privacy.

2. About Monas

Monas is a Decentralized Personal Data Store (PDS). It allows users to act as Data Controllers, managing their personal data. Monas grants users the power of privacy, enabling their intentions to be reflected upon businesses and services. It is built upon DID, a globally unique identifier independent of third parties, Cryptree for intuitive access control by Data Controllers, and Blockchain for managing state to ensure data authenticity and consistency.

2.1 DID

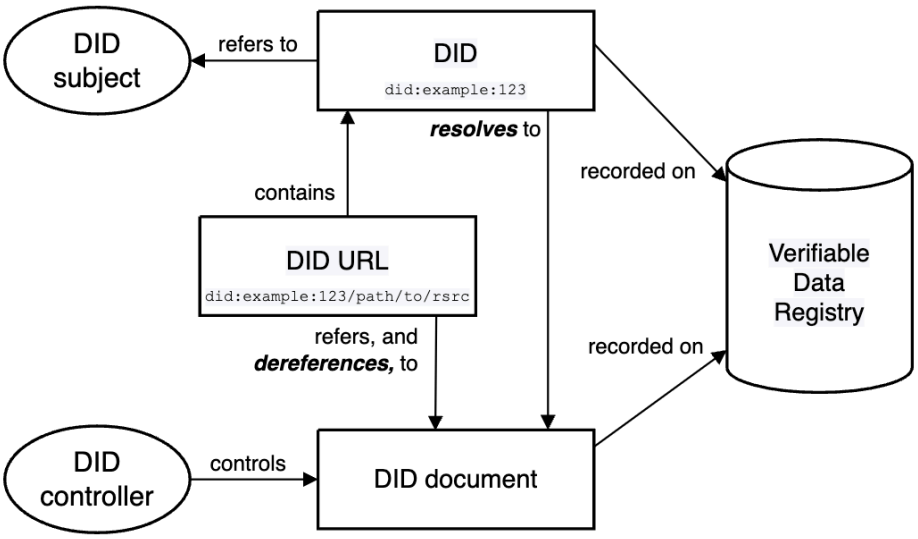

figure.1: DID architecture (Source: W3C, Decentralized Identifiers (DIDs) v1.0, Architecture Overview, available at: https://www.w3.org/TR/did-core/#architecture-overview)

DID stands for Decentralized Identifier, standardized by the World Wide Web Consortium (W3C). Unlike traditional IDs, it is designed to be independent of central registries, ID providers, and certification authorities. A DID controller can manage their identifier without reliance on a third party, and can prove their control over it (https://www.w3.org/TR/did-core/). In Monas, data is signed and stored using keys associated with a DID. This allows for the verification of whether the person who wrote to the PDS is the correct individual, whether the data has been altered or edited since it was written, and most importantly, whether the operations were performed by the DID controller. By adopting DID, we can transfer sovereignty over identifiers to personal data, enabling the realization of Self-sovereign Data.

2.2 Cryptree

Cryptree is an encrypted data structure composed of keys and cryptographic links, which can be visualized as a directed graph with keys as vertices and cryptographic links as edges.

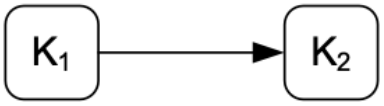

Figure 2: Figure 2: Cryptographic Link: This diagram illustrates that when there exists a key K2 derived from another key K1, everyone possessing K1 can derive K2. (Adapted from: Dominik Grolimund, Luzius Meisser, Stefan Schmid, Roger Wattenhofer, "Cryptree: A Folder Tree Structure for Cryptographic File Systems", presented at the 25th IEEE Symposium on Reliable Distributed Systems (SRDS 06), 2006, ETH Zurich, Computer Engineering and Networks Laboratory (TIK), CH-8092 Zurich, Switzerland.)

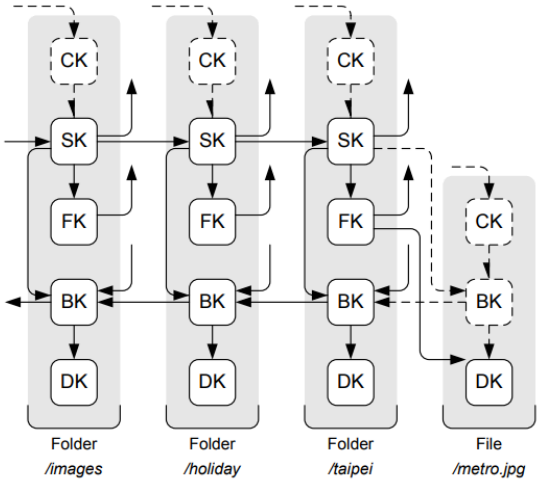

Figure 3: Cryptree (SRDS 06), 2006, ETH Zurich, Computer Engineering and Networks Laboratory (TIK), CH-8092 Zurich, Switzerland.)

In Cryptree, the construction of the encrypted data structure progresses by connecting these cryptographic links from K2 to K1. This means that knowing K1 enables the recursive derivation of its descendant keys, Kn. By building and implementing the Cryptree algorithm as a core function, Monas enables users to intuitively make decisions, allowing for flexible access control.

2.3 State management

In Monas, users can create private spaces on their preferred storage using DID and Cryptree, where the state of the space and the authenticity and consistency of data become critical. Monas must ensure these aspects, utilizing blockchain technology to maintain integrity and address these issues. When data is written within a space, its state along with the space's state is preserved on the blockchain. Administrators of the PDS can share the state of the space and the files with those granted access, who can then use this information to verify if the space's state is current and whether any tampering has occurred. By

managing state on the blockchain, Monas not only controls access to the space but also enables temporal access control.

2.4 Monas Netwrok

Monas aims to move away from central servers by establishing a Peer to Peer (P2P) network. Anyone can join the MonasNetwork, where users, rather than delegating processes to a third party, perform encryption, decryption, and data writing tasks themselves. Furthermore, peers communicate directly with each other using End to End encryption, bypassing central servers. By building a P2P network, we enable users to more profoundly experience sovereignty over their data, empowering them with privacy.

3. Architecture

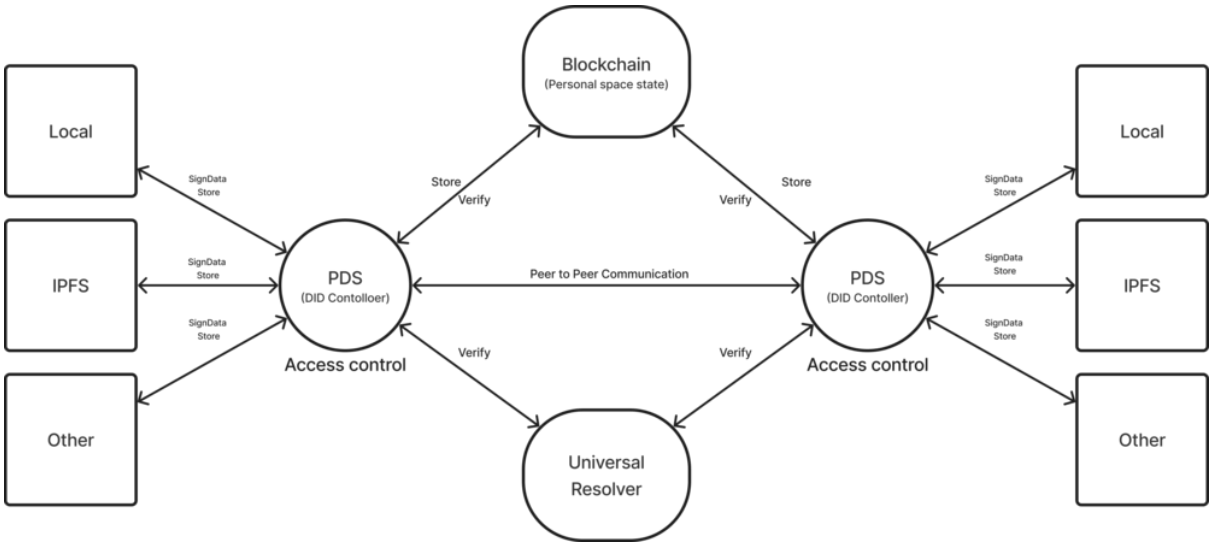

figure 4:
The architecture of Monas is structured as shown in figure 4. Users are issued a DID, which allows them to gain administrative privileges in the Personal Data Store (PDS).

3.1 Storing New Data

This section introduces the process by which users write new data into the PDS. The process described below explains how to handle Cryptree and state management, which are important features of Monas.

3.1.1 Saving State to the Blockchain

As mentioned in the introduction, maintaining consistency (Consistency) in the PDS is a significant challenge. Monas requires a reliable query point to determine the current state when collaborative editing features or application writings occur within the PDS. This query point must be resistant to malicious alterations by third parties. Therefore, Monas believes

that managing the state with blockchain technology can maintain both the consistency of the PDS and the network as a whole. Specifically, by employing hash functions to capture snapshots, which are then stored on the blockchain. Monas constructs a Merkle tree, among other structures, from the state of all layers to manage the PDS's state. The Merkle root represents the overall state of the PDS, allowing authorized individuals to verify if the data has been tampered with or if it is in its most recent state. This process ensures the PDS's consistency. Moreover, this information can be utilized for interoperability with PDS systems outside Monas. It's easy to imagine an increase in PDS services, which, if reliant on specific services, could result in data becoming siloed. Our design aims to address this issue from the outset.

3.1.2 Encryption and Storage of Data

Encryption processes occur within the Cryptree algorithm. Data signed by a DID is encrypted with a key, K, and stored in a storage selected by the user. This key, K1, is linked to a parent node, key K2. Meaning, if one knows K2, they can access all its descendants. The crucial aspect here is that users can choose where their data is stored, moving away from reliance on storage providers, which poses risks of data being seen or tampered with. By implementing Cryptree, Monas enables users to have complete control over their data, effectively abstracting storage.

3.2 Access Control

Access control plays a crucial role in Monas, allowing users to manage who can have access to, or be denied access to, specific data or places. Monas implements two types of access control:

Read Access Control: This controls who can read the data within the PDS. Users can control access to individual or multiple pieces of data with a single key. Implementing Cryptree facilitates intuitive and flexible access control.
Write Access Control: This determines who can write data to the PDS. PDS administrators have constant writing privileges. Write access control is essential for enabling collaborative editing and the development of various applications on the Monas network.

Furthermore, the aspect of access rights underscores the importance of DID. Traditionally, it was impossible for users to alter data managed by platforms or services within databases. However, Monas utilizes DID to sign data upon writing, enabling the proof that data remains trustworthy even if management rights are transferred from the platform to the user. DID incorporates platform trust, operating on the same principle as Verifiable Credentials (VC).

4 Discussion

In the sections above, we introduced the components and roles of Monas. Now, we will discuss the potential societal impact Monas could have, as well as challenges that may arise in its development.

4.1 Blockchain as State Management

Monas has the potential to address the issue of consistency within the PDS using blockchain technology. As mentioned in 3.1.1, managing state on the blockchain allows those with read access (such as sharers, applications, and co-editors) to verify the current state as being up-to-date. In Monas, whenever a user's space is updated, a state hash is generated, a Merkle tree is constructed, and the root hash is saved on the blockchain. This process is similar to rollups where transactions performed on the blockchain are batched together at Layer 1 (L1). Operations are processed off-chain for each account, with the root hash of these processes saved to the blockchain. This suggests that Monas could function as Layer 2 (L2) or Layer 3 (L3) solutions. Moreover, if other PDS services manage state on the blockchain similarly, communication between different PDSs becomes possible.

4.1.1 Timing of Root Hash Storage

Maintaining consistency through blockchain state management presents its challenges. For instance, if collaborative editing prompts updates, deciding when to save the root hash to the blockchain becomes problematic. Saving after every update without constructing a Merkle tree is equivalent to not using the tree at all, which could lead to high operational costs due to gas fees. One solution could be to schedule updates at specific times of the day or at intervals chosen by the user to reduce the frequency of transactions. However, this might not be a definitive solution and requires further consideration.

4.2 Collaborative Editing

As detailed in section 3.3 regarding access control, the design of Write access control is more complex compared to Read access control due to the emergence of collaborative editing. Individuals with write permission can edit permitted spaces, encompassing operations like uploading files into a folder and editing files. In Monas, whenever a write operation occurs, processes for generating signatures and hash values are executed to ensure authenticity and consistency. As mentioned in 4.1.1, each operation incurs computational and gas costs. Furthermore, Monas processes transactions client-side, directly between peers, which introduces challenges in sequencing concurrent operations. To maintain data synchronization and consistency, Monas intends to adopt Conflict-free Replicated Data Types (CRDTs). This approach, enabled by blockchain-based state management, incorporates temporal concepts, allowing for the sequencing of operations.

4.3 The Presence of Keys

The management of keys used on the blockchain is a critical area of focus. In Monas, two types of keys exist: signing keys and encryption keys. The former is used for verifying if a write operation from an authorized individual is legitimate and untampered, while the latter is utilized for encrypting data and metadata within Cryptree. Thus, users must manage two keys, presenting a challenge. One solution is to enable both signing and encryption with a single key, potentially through threshold cryptography or secret sharing implementations like the Lit protocol or Threshold network. This setup could condition authentication on successful signature verification, allowing decryption only upon successful authentication. Furthermore, developing new DID methods could prove essential in asserting that both the signing and public keys originate from the same entity.

4.4 Client-side Processing and Delegation

As discussed in 2.4, Monas is developing a network that facilitates data addition, encryption, and decryption processes to be conducted client-side, allowing direct communication between clients. These processes are designed with the assumption that anyone can execute them client-side. Not everyone uses a personal computer today; some processes need to be executable on smartphones. It necessitates contemplating which processes must be feasible on such devices to maintain self-sovereign data management. The approach of delegating tasks to capable individuals, rather than establishing central servers in Monas, presents a viable business model. Considering delegation on potentially unreliable devices maximizes the boundaries for minimum necessary processing.

4.5 The Concept of a Meta Platform

The issue of businesses not profiting from users storing their personal data has frequently been discussed. Platforms and businesses have traditionally profited by collecting data, essentially knowing more about the user than other platforms. However, this leads to data silos and has its limitations. Data within a single enterprise only holds meaning within that platform. In the frame of the Semantic Web, linking data across different platforms and contexts allows it to maintain various meanings. Data that previously only had relevance within one platform can, through interlinking and interoperability between platforms, achieve connections denoted by the variable N. This transition from a 1-to-1 relationship between users and platforms to a 1-to-N relationship where all platforms have access rights to user data could enhance the value of data through accelerated interoperability among numerous platforms. This aligns with the network effect mentioned in the concept of a Meta platform. Monas seeks to realize a Meta Platform by ensuring authenticity, consistency, and reliability through cryptographic and decentralized technologies.

4.6 Data Voting

Monas' approach represents a paradigm shift from traditional "solid-line data links" to "dotted-line data links," as discussed in the Introduction. This means third parties cannot

interfere with users' data without their consent in Monas. This can be seen as empowering users against rapidly evolving AI and platforms. Users can choose not to provide their data to flawed AI or platforms, effectively reflecting their intentions through their data on platforms and AI. Thus, data can be viewed as a form of voting right, leading to governance over platforms and technologies.

5 Conclusion

Monas proposes a decentralized Personal Data Store (PDS) aimed at simultaneously ensuring data privacy and interoperability. By leveraging blockchain and cryptographic technologies, Monas guarantees data authenticity and consistency, allowing users complete control over their data. Unlike traditional methods that link data, platforms, and contexts with solid lines, Monas uses dotted lines, reflecting users' intentions and potentially solving the existing issues of data silos and privacy breaches.

6 Next Work

We are currently advancing the implementation of our prototype, focusing on Read access control and state management on the blockchain. The prototype employs Cryptree implemented via IPFS to structure data, while consistency is achieved using Tableland and Filecoin for state management.
About prototype:
https://github.com/Monas-project/Filecoin-Data-Economy-Hackathon/blob/main/docs/prototype.md
Following the implementation of our prototype, we will first focus on addressing the following areas:
・Developing the Write access control feature
・Selecting encryption algorithm
・Constructing a Peer to Peer network
・Establishing a key generation algorithm
Github: https://github.com/Monas-project

References

What Are Linked Data and Linked Open Data?:https://www.ontotext.com/knowledgehub/fundamentals/linked-data-linked-open-data/
Learn Concepts: Semantic Web:
https://medium.com/mattr-global/learn-concepts-semantic-web-250784d6a49f
Four things everyone should know about the fair data economy:
https://mydata.org/2023/05/16/four-things-everyone-should-know-about-the-fair-data-economy/
Nostr 2.0: Layer 2 Off-Chain Data Storage. Syncing Nostr Relays and Paying them to Become Full-Nodes:
https://medium.com/@colbyserpa/nostr-2-0-layer-2-off-chain-data-storage-b7d299078c60
Conflict-free Replicated Data Types: https://arxiv.org/pdf/1805.06358.pdf

Decentralized Identifiers (DIDs) v1.0: https://www.w3.org/TR/did-core/
Cryptree: A Folder Tree Structure for Cryptographic File Systems: https://ieeexplore.ieee.org/document/4032481
Decentralized Identity as a Meta-platform: How Cooperation Beats Aggregation: https://nbviewer.org/github/WebOfTrustInfo/rwot9-prague/blob/master/final-documents/CooperationBeatsAggregation.pdf
Meta-Platforms and Cooperative Network-of-Networks Effects: https://medium.com/selfrule/meta-platforms-and-cooperative-network-of-networks-effects-6e61eb15c586
Provably-Secure Time-Bound Hierarchical Key Assignment Schemes: https://eprint.iacr.org/2006/225.pdf
A new key assignment scheme for enforcing complicated access control policies in hierarchy: https://www.sciencedirect.com/science/article/abs/pii/S0167739X02002005
Data Subjectivation - Self-sovereign Identity and Digital Self-Determination: https://link.springer.com/article/10.1007/s44206-023-00048-0#ref-CR19
Linked Data: https://archive.is/syHB3#selection-793.0-931.255
Lit protocol: https://www.litprotocol.com/
Personium: https://personium.io/en/index.html
Solid: https://solidproject.org/
Peergos: https://peergos.org/