



AiCE Undergraduate Research Project Final Report

Spring 2025 Semester

Med-D: Decentralized Medical Application

Team Members

Nunthatinn Veerapaiboon, Poon, nveerap@cmkl.ac.th
Thanawin Pattanaphol, Win, tpattan@cmkl.ac.th
Atchariyapat Sirijirakarnjareon, Beam, asiriji@cmkl.ac.th
Petch Suwapun Diamond, psuwapu@cmkl.ac.th

Advisor

Dr. Charnon Pattiyanon

6th May 2025

Table of Contents

Table of Contents	1
Chapter 1	1
1.1 Abstract	1
1.2 Problem Statement	1
1.3 Project Solution Approach	2
1.4 Project Objectives	3
Chapter 2	5
2.1 Fundamental Theory and Concepts	5
2.1.1 Self-Sovereign Identity	5
2.1.1.1 Verifiable Credentials	5
2.1.1.2 Issuers, holders, and verifiers	7
2.1.1.3 Digital Wallets	7
2.1.1.4 Digital Agents	8
2.1.1.5 Decentralized Identifiers	8
2.1.1.6 Verifiable Data Registries	9
2.2 Technologies	9
2.2.1 Hyperledger Indy	9
2.2.2 Hyperledger Aries	10
2.2.3 Hyperledger AnonCreds	10
2.2.4 Credo	10
2.2.5 Flutter	11
2.2.5.1 Flutter Secure Storage	11
2.3 Related Research	12
2.3.1 Shuaib et al. (2021), "Secure decentralized electronic health records sharing system based on blockchains"	12
2.3.2 Azaria, et al. (2016) "MedRec: Using Blockchain for Medical Data Access and Permission Management,"	12
2.4 Market Analysis	13
Chapter 3	16
3.1 Project Planning and Design Phase	16
3.2 User Interface (UI) Design	17
3.3 Algorithm Design and Core Logic	17
3.4 Programming and Implementation	17
Chapter 4	19
4.1 Med-D Ecosystem Architecture	19
4.2 Mobile Wallet Application	19
4.3 Hospital Website	22
4.4 Med-D Layer	26
4.4.1 Middleware API	26
4.4.1.1 Pre-transfer Handshake	26
4.4.1.2 Scenario 1: Patient get medical records from Hospital (EHR to Wallet)	27

4.4.1.3 Scenario 2: Patient share medical records to Hospital (Wallet to EHR)	28
4.4.2 Agent & Distributed Ledger (Blockchain)	29
Chapter 5	31
5.1 Summary of Accomplishments	31
5.2 Issues and Obstacles	31
5.3 Future Directions	31
5.4 Lessons Learned	31
References	33

Chapter 1

Introduction

1.1 Abstract

Health information management in Thailand is characterized by significant data fragmentation across diverse public and private healthcare providers employing non-interoperable Electronic Health Record (EHR) systems. This systemic lack of integration hinders continuity of care, contributes to diagnostic delays, necessitates redundant investigations, and poses risks associated with incomplete patient histories, such as missing allergy data. This report details the design and prototype implementation of Med-D, a decentralized health record management system proposed to address these challenges within the Thai context. Med-D utilizes W3C Distributed Identifiers (DIDs) for cryptographic identity management of patients and providers, coupled with a simulated blockchain ledger, managed via an Agent service, for storing immutable integrity proofs (SHA-256 hashes) of medical records. A central coordinating API orchestrates key workflows, including DID registration, hash generation during record issuance by simulated EHRs, ledger updates via the Agent, and secure record delivery to a patient-controlled digital Wallet. Furthermore, a verification workflow enables patients to share specific, integrity-verified records with new providers upon consent. The implementation demonstrates the technical feasibility of core Med-D functionalities within a controlled HTTP environment. By promoting data portability, verifiability, and patient control, this architecture offers a potential pathway to mitigate data silos and enhance the efficiency, safety, and patient-centricity of healthcare information exchange in Thailand.

1.2 Problem Statement

Thailand's health sector remains characterized by multiple disconnected electronic record systems and paper-based archives, so that patient data is spread across unlinked hospitals, clinics, and private providers. In practice, each facility often uses its own proprietary health information system and data formats. As one Thai expert observes, "there is neither a centralised system nor initiative to streamline all the healthcare data into a single platform like a National EMR system". Consequently, patient information (e.g., clinical history, medications, test results) is fragmented across silos and even within the same facility, requiring staff to re-enter or reconcile data manually [Jagatheesan et al., 2022; Xu et al., 2012]. This fragmentation persists despite broad EHR adoption: a nationwide survey (2011) found only ~50% of hospitals had even basic EHRs, with comprehensive systems in just ~5% [Palasuwan et al., 2019], and information sharing between institutions remains "very limited."

Thailand's 2016 ten-year e-Health Strategy (e.g., "AI Smart Health Profile") and recent pilots (e.g., Mor Promt personal health record) aim to improve interoperability [MOPH, 2022], but to date no fully integrated, nationwide EHR exists.

Consequences of Fragmentation

1. Clinical decision-making and patient safety:
Disconnected records severely hamper clinical care. When relevant history is split across systems, providers may miss critical data or duplicate tests. For example, missing allergy or test

results stored in a different record can directly harm patients [Joffe, 2012]. In theory, health information exchange (HIE) improves safety by reducing errors and eliminating redundant testing [Sorondo et al., 2020], but in Thailand, fragmented systems prevent these gains. Studies show that duplicate or incomplete records lead to missed abnormal findings and treatment errors; Thai experts similarly report that “fragmented databases and software unavailability limit local health data sharing,” undermining coordinated care [Park et al., 2025]. Without interoperable EHRs, physicians must rely on patient recall or fragmented paper reports, increasing the risk of misdiagnosis and medication errors.

2. Administrative burden and operational inefficiency:

Fragmentation imposes heavy workload on staff. Without shared systems, each facility duplicates data entry, referral paperwork, and manual record transfer. One study notes that duplicate records alone generate significant costs – for example, time spent locating correct charts, repeating tests, and reconciling files [Joffe, 2012]. Conversely, implementing EHRs can streamline workflows: hospitals with mature EHR use have seen shorter patient wait times and higher clinician productivity [Sorondo et al., 2020]. In Thailand, however, the disjointed system means clinicians and administrators spend extra hours managing paperwork and cross-checking data. This inefficiency also limits data capture for research and planning: policymakers lack easy access to integrated datasets, hampering epidemiological surveillance [Park et al., 2025; Xu et al., 2012].

3. Limited patient access and engagement:

Fragmented data also restricts patient empowerment. Thai patients generally cannot view a single consolidated health record; instead they must obtain printed summaries or digital copies from each provider. The Ministry of Public Health’s new “Mor Promt” PHR platform allows patients to retrieve health data from participating providers via an OTP system [MOPH, 2022], but uptake is limited and many facilities are not yet connected. This lack of accessible records undermines preventive care and second opinions. Studies have shown that when patients can easily review their records (e.g., via patient portals), communication improves and medical errors drop [Radell et al., 2022]. In Thailand’s current environment, patients cannot routinely share complete histories between doctors or monitor chronic conditions proactively.

4. Prolonged patient wait times:

Data silos contribute directly to longer waits for care. When records are not electronically shared, front-line staff must spend extra time retrieving or recreating information at each visit or referral. Studies from other settings have found that mature EHR adoption is associated with decreased waiting times for patients [Sorondo et al., 2020]. Thai research similarly notes that implementing a unified digital referral platform “will increase data visibility and accessibility by reducing waiting time” [Kaewla-or et al., 2022]. In practice, fragmented information means longer queues and delays: for example, a referral to a tertiary hospital often involves manual faxed summaries and duplicate testing, lengthening the patient’s journey.

1.3 Project Solution Approach

To address the challenges of data fragmentation, limited interoperability, and lack of patient control inherent in Thailand’s current health information landscape, this project proposes and prototypes Med-D, a decentralized health record management system. Med-D shifts the paradigm from institution-centric data silos towards a patient-centric model, where individuals gain secure ownership and control over their verified medical history.

The core approach leverages two key technologies: Decentralized Identifiers (DIDs) and Blockchain-based Integrity Proofs.

1. **Decentralized Identity:** Med-D utilizes W3C DIDs to provide unique, cryptographically verifiable identifiers for both patients (via their digital Wallet) and healthcare providers (EHR systems). This removes reliance on centralized identity providers and allows for secure, peer-to-peer authentication and authorization within the ecosystem. Public keys associated with these DIDs form the basis for secure data exchange protocols (though full encryption is part of future work).
2. **Verifiable Data Integrity:** Instead of attempting to create a single, centralized database (which faces significant implementation hurdles), Med-D focuses on ensuring the integrity and provenance of records shared between parties. When an EHR issues a record via the Med-D API, a cryptographic hash (specifically, SHA-256 in this prototype) of the standardized record content is generated. This hash, acting as an immutable proof or 'digital fingerprint' of the record at the time of issuance, is stored on a simulated blockchain ledger (managed by the Agent service) and cryptographically linked to the patient's Wallet DID.
3. **Patient Data Custodianship via Digital Wallet:** The actual medical records, now including their associated integrity proofs (hashes), are delivered securely to the patient's digital Wallet application. The patient becomes the custodian of their consolidated records from various providers. Crucially, no medical data is stored on the blockchain ledger itself, only the integrity proofs, thus preserving patient privacy while ensuring data verifiability.
4. **Orchestration and Verification:** A central coordinating API facilitates the key interactions: registering DIDs, generating and recording proofs during issuance, and verifying records during sharing. When a patient consents to share a record from their Wallet with a new provider, the API executes a verification workflow: it confirms the proof (hash) exists on the ledger via the Agent and recalculates the hash of the received record to ensure it matches the stored proof, guaranteeing that the data has not been tampered with since issuance. Only verified records are forwarded to the receiving EHR.
5. **By decoupling record storage (in the Wallet) from identity and integrity verification (using DIDs and the ledger), Med-D aims to provide a practical framework for achieving secure data portability and trustworthiness across Thailand's diverse healthcare network. This prototype focuses on implementing and validating the core technical workflows underpinning this decentralized approach.**

1.4 Project Objectives

The primary goal of this project is to investigate the feasibility and demonstrate the core functionalities of a decentralized health record management system, termed Med-D, designed specifically to address data fragmentation and enhance patient data control within the context of the Thai healthcare system, leveraging the capabilities of the Hyperledger Indy distributed ledger and providing a basic mobile wallet interface for patient interaction.

1. To propose and architect a decentralized health record management system (Med-D) utilizing W3C Decentralized Identifiers (DIDs) anchored on Hyperledger Indy for secure patient (Wallet) and provider (EHR) identity management, mediated by an Agent service.
2. To develop the backend services (API, Agent interfacing with Indy) and a Flutter mobile Wallet to implement key workflows, including: DID registration; medical record issuance by simulated EHRs with ledger-anchored integrity proofs (SHA-256 hashes); patient storage and control of records within the Wallet; and a verification process for sharing integrity-checked records between the Wallet and simulated EHRs via API and Agent/Indy interactions.

3. To evaluate the technical feasibility of the implemented prototype (Wallet, API, Agent, simulated Indy ledger, simulated EHR interactions) in demonstrating verifiable data integrity and enabling patient-controlled medical record portability within the proposed decentralized architecture.

Scope Limitations:

This project focuses on the design and technical implementation of the core backend services (API, Agent), their interactions with Hyperledger Indy, a Flutter Wallet UI, and simulated EHR interactions. The scope does not include:

- Implementation of robust, production-grade security mechanisms beyond DID cryptography (e.g., transport-level encryption [TLS was deferred], secure key management within the Wallet/EHR, advanced access control).
- Full implementation of W3C Verifiable Credentials standards for the medical records themselves within the Wallet (simple hashes are used primarily for integrity proof, though Indy interaction involves VCs for the proof itself).
- Advanced consent management logic within the Wallet beyond selecting records to share.
- Scalability, performance testing under load, or deployment in a clinical setting.
- Setting up and managing a production-grade, multi-organization Hyperledger Indy network; interaction is assumed with an existing development/test network or node.
- Development of a fully functional EHR graphical user interface (GUI) or web portal for hospital use based on the initial design wireframes. EHR interactions for this prototype were simulated using direct API calls (e.g., via Insomnia).
- The objective is to demonstrate the proof-of-concept for the core decentralized data flow, integrity verification, identity management, and basic patient interaction via a mobile wallet, facilitated by the coordinating API and the Agent's interaction with Hyperledger Indy.

Chapter 2

Background

2.1 Fundamental Theory and Concepts

2.1.1 Self-Sovereign Identity

Self-sovereign identity or SSI is a type of decentralized digital identity system that gives users or individuals control and ownership over the information which in turn means that the user gets to control what data can be viewed or accessed and when; eliminating the need for a central authority to act as a middle man to prove the users' identity. This concept mostly handles the entity of a "digital identity" - a user's online identity, similar to the current types of physical identity, such as, national ID cards, passport and passport license; the concept of self-sovereign identity protects such data by keeping the data secure and private.^{1 2 3}

The overall concept of self-sovereign identity consists of the following:

2.1.1.1 Verifiable Credentials

Verifiable credentials or VC are the heart of the SSI concept; they are essentially a digital version of what we know now as set of information that an authority claims to be true about the subject of the credential that cannot be tampered with; this ranges from paper of plastic cards that are in a physical wallet such as: government ID, driving licenses, and employment cards to the other sets of information that cannot be stored in a wallet, such as, birth certificates issued by hospitals, diploma issued by a university, passport issued by a government of a country and others, to information about non-human subjects such as: pet vaccinations records, IoT credentials, and so on.⁴

These credentials all contain three different sections:

- Issuer - A single authority that issues a set of claims about a subject, such as, hospitals, government agencies, and others.
- Holder - The entity, which can be a person, organization or a thing, that keeps the credentials in their digital wallet.
- Claims - The set of information that the issuer claims to be true about the holder such as age, height, relationships, medical benefits, library privileges and others.

In addition to the three parts mentioned above, a credential must also be verifiable. To accomplish this, verifiable credentials use cryptography, the internet and a standard protocol to create a verification process that takes a very short amount of time - the verification process essentially answers these questions.

- Does the credential have the data encoded in a standard format that the verifier wants?
- Does the credential include a digital signature from the issuer?
- Has the credential expired?

¹ Preukschat and Reed, Self-Sovereign Identity.

² "Self-Sovereign Identity (SSI)."

³ "Self-Sovereign Identity."

⁴ Preukschat and Reed, Self-Sovereign Identity; "Verifiable Credentials."

- Does the credential have cryptographic proof that the holder of the credential is the owner of the credential?

These concepts combine to create a verifiable credential, similar to Figure 1, which shows a digital credential of a standard US driver's license which follows the W3C Verifiable Credentials Data Model 1.0 specification.

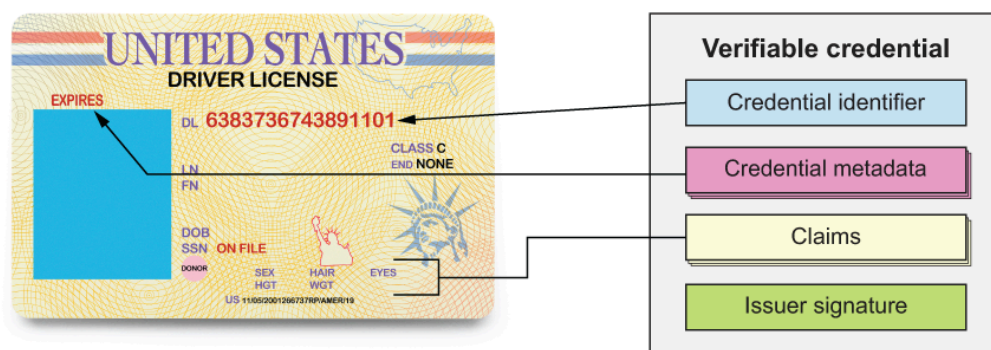


Figure 1: Diagram labeling the different parts of a W3C-standard based verifiable credential. ⁵

In the Med-D project's use case, we are using the W3C standard of verifiable credentials to create a data scheme for a medical record for hospitals to issue to a holder, similar to Figure 2.

```
{
  "holder": "did:key:z6Mkq7XCJQBdsY7FGcpCD3UN18DSq7RA88kHQjgWkas7Scdn",
  "credentials": [
    {
      "id": "<credential-id>",
      "name": "Medical Data",
      "type": [
        "VerifiableCredential",
        "MedicalData",
        "PrettyVerifiableCredential"
      ],
      "issuer": {
        "id": "did:key:z6MksHuVuB9xjbk4kQqa7R69qaWA9vADkhRvs16L51mVHCLD",
        "name": "Lopital",
        "description": "Low cost hospital"
      },
      "@context": [
        "https://www.w3.org/2018/credentials/v1",
      ],
      "issuanceDate": "2024-11-19T07:25:46.528Z",
      "credentialSubject": {
        "height": 170,
        "weight": 74,
        "symptoms": "Coughing",
        "condition": "COVID-19",
        "diagnosis": "COVID-19",
        "labResult": "Real COVID-19",
        "medication": "Paracetamol",
        "bloodPressure": "000000000",
        "appointmentDate": "2025-08-15T05:00:00.000Z"
      }
    }
  ]
}
```

Figure 2: A sample schema of a Med-D verifiable credential

⁵ "Verifiable Credentials Data Model v1.1."

2.1.1.2 Issuers, holders, and verifiers ⁶

There are three main types of entities surrounding the use of verifiable credentials:

- **Issuers:** The source of the data, most of which are government agencies, financial institutions, hospitals and individuals. In Med-D's context, the issuers will be mainly hospitals or health institutions who will be issuing medical records for customers.
- **Holders:** The entity who requests verifiable credentials from issuers and stores or holds them in the holder's digital wallet, and present proofs of claims from one or more credentials when requested by verifiers (with the consent of the holder)
- **Verifiers:** The entity that essentially does the job of verifying the validity of the credentials through the request of proofs from holders of the credentials; if the holder agrees, the holder's agent would respond with a proof that the verifier can use to verify the credential. In this specific step, the DID or Decentralized Identifier will be used to verify the issuer's digital signature of that credential.

They can all be combined to create an ecosystem, collectively called the "Trust Triangle" which is shown in the figure below.

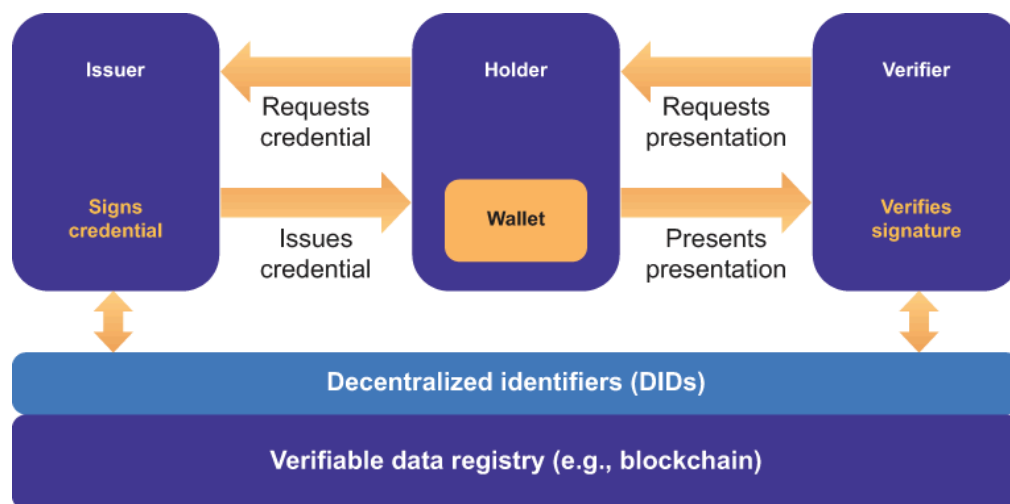


Figure 3: The diagram below shows the role of each entity in the exchange of VCs.

2.1.1.3 Digital Wallets

Digital wallets do the job of storing the holder's credentials, protecting said credentials from theft and being easily accessible and portable across different devices. In Med-D's context, the patients' medical wallet will be following the open standards for portable, self-sovereign verifiable credentials by the W3C Verifiable Credentials Data Model standards along with implementing an encrypted data storage and automated encrypted credential backups to prevent data loss if the the holder's device were to be destroyed or lost.

⁶ Preukschat and Reed, Self-Sovereign Identity.

2.1.1.4 Digital Agents

Digital agents play the role of the protector or the guardian of the wallet, in which, they have the job to secure and protect the data inside the holder's digital wallet and make sure that the only entity that is able to access the verifiable credentials is the holder (the entity that controls and owns the wallet). They also play a role in being the medium to communicate with another agent, form a connection between the two entities and exchange verifiable credentials between each other through various ways, such as, a decentralized and secure messaging protocol such as DIDComm or QR Codes. In Med-D's context, we are considering the use of QR Codes for our overall system architecture as a way to exchange credentials between agents, however, we are also considering the use of a secure messaging protocol as another way for hospitals to be able to request credentials from the holder.

2.1.1.5 Decentralized Identifiers

Decentralized Identifiers or DIDs are essentially a global unique identifier that is similar to an IP address or an ID for each entity (Issuer, Holder and Verifier) which allows secure messaging between digital agents and wallets as well as a way for agents to be able to send cryptographically verifiable proofs of verification credentials to each other. DIDs are designed to work with any modern blockchain technology, distributed ledger technology (DLT) or other decentralized networks using a DID method which is defined as the following actions on any DID:

- DID creation and its accompanying DID document (public keys and other metadata of the DID subject)
- How the DID can be used to look up a DID document
- Updating DID document for a DID
- DID deactivation

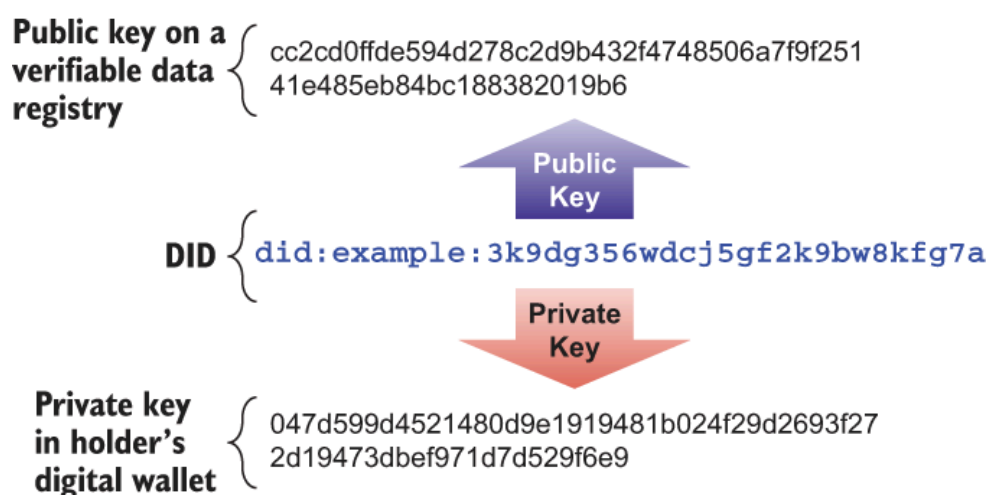


Figure 4: Example of a decentralized identifier - works as the address of a public key on a blockchain or other types of decentralized network.

The current DID Specification Registry is maintained by the W3C DID Working Group which has more than 80 DID method, including several methods for some cryptocurrencies as well as ones that do not need a distributed ledger as they work in a peer-to-peer fashion - similar to how two devices can connect to each other using their IP Addresses in a TCP/IP protocol stack - DIDs can use SSI protocol stack to form a cryptographically secure

connection to transfer data, this usage of DID-DID connections provide several properties to digital relationships or connections.

- The connection between two parties is permanent and will never break unless one of the parties wants it to.
- All communications over the connection are encrypted and digitally signed.
- The connection is end-to-end encrypted.
- The connection supports the transfer of verifiable credentials to establish trust.
- The connection can also be used for other applications that need secure, private, reliable digital communications.

2.1.1.6 Verifiable Data Registries

In SSI, the verifiable data registries are a decentralized network which in Med-D's case is a distributed ledger technology or Blockchain. Public & Private key cryptography is a way of securing data using material algorithms based on cryptographic keys and in identity management, Blockchain is used for decentralized public key infrastructure or DPKI and is essentially a way to exchange public keys between any two peers in a private and secure matter and to store these public keys in which they can be used to verify signatures on verifiable credentials.⁷

Blockchain is preferable for the SSI architecture as it is a highly tamper-resistant transactional distributed database that not a single entity or party controls, thus, it provides an authoritative / trustworthy source of data that many different peers can trust without any single one of them being in complete control over the whole Blockchain network.

To achieve this, the Blockchain does the following:

- Every transaction to the Blockchain is digitally signed; each peer manages its private keys and signs its transactions (writing a record) with the Blockchain.
- The transactions are grouped into blocks that are cryptographically hashed and linked to the previous block, which means that, as the data are linked together, they cannot be modified, thus an immutable chain of ordered transactions.
- Every new block when entered to the Blockchain is cryptographically replicated across all ledger nodes on the network which are run by other peers. This step is done via a consensus protocol which when done, every peer node in the network ends up with the copy of the last block and they all agree on that copy to be the correct information.

2.2 Technologies

2.2.1 Hyperledger Indy⁸

Hyperledger Indy is a decentralized, distributed ledger technology (DLT) designed specifically for creating and managing self-sovereign identities (SSI). It allows individuals to own and control their personal data in a way that is verifiable, secure, and private. In the context of a healthcare system, Indy enables patients to store their medical data in a digital

⁷ Preukschat and Reed.

⁸ "Hyperledger Indy — Hyperledger Indy 1.0 Documentation."

wallet, which can then be shared with healthcare institutions in a secure and privacy-preserving manner using blockchain technology.

In Med-D's context, Hyperledger indy is used as the "Ledger" as well as the "Verifiable Data Registry" for the verification of "Proof Credentials" along with enabling the use of data recovery.

2.2.2 Hyperledger Aries

Hyperledger Aries is an open-source toolkit for developing decentralized identity solutions and digital trust; developed by the Linux Foundation and is licensed under the Apache-2.0 license. It enables the developer to create software that can issue, store, and present verifiable credentials; establish secure communication channels through DIDComm; compatible with government regulations in regards to data security and privacy; and to create digital wallets / agents across different platforms, such as, phones, enterprise systems or the cloud.

Aries is a toolkit that is considered as "platform agnostic" in many ways. This is due to the design of Aries, in which it is designed to be compatible with a variety of protocols, credential standards, ledgers, and registries. Aries also provides various frameworks that can be used to develop in many programming languages, such as Rust, JavaScript and Python.

In Med-D's context, Aries, or its sub-system, "Askar", is used in the agents that run in the hospitals, in which, it is used to issue "Proof Credentials" which are essentially records that contain the "proofs" of the medical records that are used for "proving" the existence of said medical records, and after its issuance, is stored in the hospitals' storage or "wallet" (not to be confused with the patient's medical record wallet).

2.2.3 Hyperledger AnonCreds

Hyperledger AnonCreds is an open-source implementation of W3C's Verifiable Credentials using Zero Knowledge Proofs (ZKPs). It is designed to provide strong cryptographic privacy guarantees while ensuring that individuals can share selective data with third parties without disclosing unnecessary personal information.

AnonCreds enhances the capabilities of Indy by focusing on privacy and anonymity while allowing for data verifiability along with the revocation of credentials, allowing the issuing authority, such as hospitals, to revoke an old credential when needed.

In Med-D's context, AnonCreds is used specifically for the format of Med-D's "Proof Credentials": records that contain the list of "Proofs" which are essentially data that can be used to "prove" if a medical record exists and/or is valid or not.

2.2.4 Credo

Credo is an open-source framework written TypeScript for building decentralized identity solutions that supports multiple identity standards; designed to be "agnostic" to any communication protocols, credential format, signature suite, or did methods. Credo provides features that help developers easily create platforms that are based on decentralized identity concepts, such as, self-sovereign identity (SSI); they are: DIDComm v1, and Aries Interop

Profile, Decentralized Identifiers, OpenID4VC, W3C Verifiable Credential formats and AnonCreds, and more.

In Med-D's context, Credo is used as the basis of implementing the Hyperledger stack for issuing, storing, and verifying "Proof Credentials" as mentioned above. Credo provides a much easier, simpler and faster way to develop the Med-D layer, specifically the agent that handles any communication between the Indy ledger and the middleware API.

2.2.5 Flutter

Flutter is a cross-platform application development framework created by Google that allows developers to build natively compiled applications for mobile, web, and desktop from a single codebase. In this project, Flutter is used to develop the mobile application that patients will use to store and manage their digital wallets containing their medical credentials and self-sovereign identity data. Flutter enables a seamless user experience across various platforms with high performance and flexibility.

Flutter provides various capabilities, such as, cross-platform development through one single codebase for both Android and iOS, high performance thanks to the compilation process that compiles Dart into each platform's respective native machine code, and the various amount of UI components that Flutter provides.

In Med-D's context, we use Flutter as our main UI framework for my user application due to its one-codebase-for-all feature, along with the various UI components that help create a more polished and user-friendly interface for our Med-D client app.

2.2.5.1 Flutter Secure Storage⁹

Purpose:

Flutter Secure Storage is a package for securely storing sensitive data on mobile devices, such as passwords, tokens, and cryptographic keys. In the context of the self-sovereign identity (SSI) system, this package is used to securely store patients' private keys and other sensitive information within the Flutter application. This ensures that only the user can access their medical data and digital credentials.

In Med-D's context, the Flutter Secure Storage is used for storing the patient's personal information, medical records, and any other private and sensitive information in a secure, encrypted, and safe storage environment, along with its support for cross-platform development, this enables Med-D to provide a safe and secure storage for all platforms, such as, iOS and Android.

⁹ "Flutter_secure_storage | Flutter Package."

2.3 Related Research

There are several papers and / or studies that we have discovered over the past months since we have started the Med-D project; these research papers have also inspired some of our development decisions, ideas, and process.

2.3.1 Shuaib et al. (2021), “Secure decentralized electronic health records sharing system based on blockchains”

This paper discusses the use of Blockchain to improve the efficiency, security and privacy of Electronic Health Records (EHR) sharing systems; along with pointing out the flaws of current existing solutions that mostly rely on a centralized database, in which, they are susceptible to security problems, such as, Denial of Service (DOS) attacks and the concept of a single-point-of-failure due to its centralized design.

The paper proposes a permitted Blockchain based healthcare data sharing system that uses Blockchain to address the issues of potential DoS attacks and single-point-of-failure risk, based on the Istanbul Byzantine Fault Tolerant (IBFT) consensus algorithm and Interplanetary File System (IPFS); along with implementing the proposed system on an enterprise Ethereum Blockchain, Hyperledger Besu.

The paper then summarizes its findings that the proposed system performs better than existing Blockchain based systems, as well as, the greater level of security that a decentralized file system provides while maintaining the same level of performance that a centralized database can offer.

2.3.2 Azaria, et al. (2016) "MedRec: Using Blockchain for Medical Data Access and Permission Management,"

This paper proposes a Blockchain-based medical health record sharing, called Medrec, in which it uses Blockchain as its main permission management platform to enable data sharing among different institutions, such as hospitals, insurance companies, pharmacies, and patients. However, the data is not transferred over the Blockchain nor is stored in a certain public location.

Medrec addresses four main challenges of traditional healthcare systems:

- Fragmented data access
- Data interoperability
- Improved data sharing
- Patient Participation

However, there are several issues when it comes to Medrec, such as, the consensus algorithm that is used, the proof-of-work algorithm, which can potentially consume a huge amount of energy along with having high latency and low throughput; privacy risks due to the design that enables providers to be able to submit transactions to the Blockchain using the same Blockchain address, and lastly, Medrec does not allow a more thorough configuration of permissions which leads to an entity having permissions to access data and being able to access the data without any limits or restrictions.

When compared to Med-D, there are certain features that the related research papers have and do not have. Figure __ shows a side-by-side comparison between the three systems.

Paper	Blockchain Type	Consensus Algorithm	Data Location	Permissioned Blockchain	Emergency Data Access
Med-D	Hyperledger Indy	PBFT	Off-Chain	✓	✓
Shuaib et al. (2021)	Hyperledger Besu	IBFT	Off-Chain	✓	✓
Azaria et al. (2016)	Ethereum	PoW	Providers' Database	X	X

Figure 5: Comparison table between Med-D, Shuaib et al., and Medrec.

We can observe that Med-D and Shuaib et al. have similar capabilities, however, there are several differences between them, such as, Shuaib et al. (2021) employs an external IPFS storage for encrypted users' encrypted data while Med-D employs an external PostgreSQL database, however, we are considering migrating to another database system. Other the other hand, Azaria et al.'s Medrec uses the Ethereum Blockchain, in which, uses the Proof-of-Work consensus algorithm, which can lead to potential issues surrounding high latency and low throughput, while the IBFT (Istanbul Byzantine Fault Tolerant) and PBFT (Plenum Byzantine Fault Tolerant) consensus algorithms are more resistant to said issues along with the significantly lower amount of energy consumed for such algorithms to run. Medrec, also lacks the consideration of emergency data access, whilst Med-D provides a solution in which the middleware API will requests each hospitals for medical records corresponding to its proof data in the proof-credentials stored in the Med-Agent & Indy Ledger.

2.4 Market Analysis

The development of Med-D takes place within Thailand's complex healthcare environment, which, as discussed earlier, is challenged by fragmented health data systems and limited communication between hospitals and clinics. To understand how Med-D fits into this picture, it's important to look at the solutions already in place or being developed and see how Med-D offers something new and valuable.

1. To show a clear understanding of current efforts in Thailand to manage and share health information, and
2. Med-D stands out by directly addressing key problems like fragmented data, lack of patient control, and difficulties in verifying whether records have been altered or not.

What Solutions Already Exist?

Thailand already uses several systems to manage health data, but each has limitations that Med-D is designed to overcome. Here's a breakdown of the main types:

1. Hospital EHR Systems

Most hospitals in Thailand use their digital systems, like HOSxP or JHCIS. These help manage patient data within the hospital but don't talk to systems in other hospitals.

- What's missing: These systems are like islands—patients often have to carry paper documents or repeat tests when they go elsewhere. Access is limited, and data can't be easily verified between facilities.

2. Government Platforms (like Mor Prompt)

Mor Prompt started during the COVID-19 pandemic but now includes features for accessing some health records, such as vaccinations and appointments, using an OTP login system.

- What's missing: Not all hospitals are connected, and the system doesn't yet give patients full control or the ability to verify if records have been changed. It's still very centralized.

3. Health Information Exchanges (HIEs)

These are networks where hospitals agree to share patient data, often using a hub or point-to-point connections. Some pilots or regional efforts exist in Thailand.

- What's missing: HIEs can be hard to manage, costly, and usually rely on trust between providers. Patients don't have much say in how their data is shared.

4. Patient Portals

Some hospitals let patients log in to view parts of their medical history, like test results or upcoming appointments.

- What's missing: These portals are limited to that hospital. They don't help patients gather data from multiple places or share it with new doctors.

5. Blockchain/Decentralized Projects

Globally, there are projects that use blockchain and decentralized IDs to help manage health records. Some focus on data access, some on logging patient consent.

- What's missing: Many are still experimental or haven't caught on. They also don't always connect well with existing hospital systems, especially in countries like Thailand.

Side-by-Side Comparison

Feature / Capability	Existing Hospital EHRs	Mor Prompt / National PHR	Traditional HIEs	EHR Patient Portals	Med-D (Prototype)
Primary Data Storage	Hospital Servers	Participating Hospital Servers	Provider Servers /	Hospital Servers	Patient Device (Wallet)

		(accessed via platform)	Central Hub		
Patient Data Control	Low (Indirect Access)	Medium (Viewing/Retrieval)	Low / Varies	Low (Viewing Only)	High (Custody & Consent)
Cross-Provider Portability	Very Limited / Manual	Limited (by platform participation)	Varies (by participation/rules)	None	High (Patient-Mediated Sharing)
Data Integrity Verification	Assumed within Silo	Assumed via Platform	Trust-Based	Assumed within Silo	Cryptographic Hash Proof (Ledger)
Identity Management	Hospital ID / National ID	National ID / OTP	Various / Central	Hospital Login	Decentralized Identifiers (DIDs)
Interoperability Approach	Proprietary / Limited	Central Platform API	Hub/Point-to-Point	Proprietary Portal	API + Agent + Wallet Mediation

Med-D's Unique Value Proposition:

Based on this analysis, Med-D distinguishes itself from existing or alternative approaches within the Thai context through its unique combination of features designed specifically to address the identified problems:

1. Patient Custodianship: Unlike portals or centralized platforms, Med-D places the consolidated records directly under the patient's control within their digital Wallet, empowering them as the primary data custodian.
2. Verifiable Integrity via Ledger: By anchoring cryptographic hashes (proofs) of records to the patient's DID on a distributed ledger (simulated via the Agent), Med-D provides a strong guarantee that shared records have not been tampered with since issuance through the API. This is a significant advantage over systems relying solely on access controls or trust.
3. Decentralized Identity: Utilizing W3C DIDs for both patients and providers removes reliance on potentially fragmented or centralized identity systems and establishes a foundation for secure, peer-to-peer interactions.
4. Pragmatic Integration: Med-D focuses on coordinating data flow between existing systems via its API and Agent layers, rather than requiring an immediate, large-scale replacement of all hospital EHRs. This potentially lowers the barrier to adoption.
5. Focus on Patient-Mediated Sharing: The architecture is designed around the patient explicitly consenting and initiating the sharing of their verified records from their Wallet to a new provider via the API's verification workflow.

Chapter 3

Methodology

This chapter details the methodology employed throughout the Spring 2025 semester for the design, implementation, and internal testing of the Med-D: Decentralized Medical Application prototype. The process involved architectural design based on Self-Sovereign Identity (SSI) principles, iterative software development of backend components and a mobile wallet frontend, simulation of ledger interactions and EHR system behavior, UI/UX design for key interfaces, and functional workflow testing.

3.1 Project Planning and Design Phase

The initial phase focused on understanding the core problem of medical data fragmentation in Thailand (Chapter 1) and defining a suitable decentralized solution approach leveraging W3C DIDs and Verifiable Data Registries, specifically conceptualizing the use of Hyperledger Indy as the target ledger technology.

- **Literature Review & Concept Definition:** Research encompassed SSI concepts, Verifiable Credentials (including Hyperledger AnonCreds for potential future use), DID methods (with Indy's did:indy method as a target), and existing blockchain applications in healthcare (Chapter 2). This solidified the design decision to decouple record storage (patient wallet) from integrity verification (ledger proofs anchored to DIDs).
- **Architecture Design:** A service-oriented architecture was designed (Section 4.1), comprising the Middleware API, Agent (simulating Indy interaction), Mobile Wallet (Flutter), and EHR (conceptualized via Figma, simulated via API calls). Roles, responsibilities, and workflows were defined.
- **Key technologies included:**
 - **Node.js (Express.js) Backend (API, Agent, Mocks):** Chosen for rapid development speed, its large NPM ecosystem (reducing boilerplate), native JSON handling ideal for APIs, and sufficient performance for prototype microservices. Enables quick iteration.
 - **SQLite (API Cache):** Selected for its simplicity (file-based, zero-config) and lightweight nature, suitable for the API's local DID/PK cache without the overhead of a full database server during prototyping. Provides basic transactional integrity.
 - **JSON Files (Mock Storage):** Used for Agent's mock ledger and Wallet/EHR mock storage due to extreme simplicity, allowing focus on inter-service logic rather than complex blockchain/storage implementation at this stage. Ideal for mocking dependencies.
 - **Axios (HTTP Client):** A standard, promise-based library for easily handling inter-service HTTP communication required between the backend components. Simplifies asynchronous network calls.
 - **Node.js crypto (Hashing):** Utilized as a built-in, standard module for generating SHA-256 hashes, providing a simple way to implement the record integrity checks needed for the proof-of-concept.
 - **Flutter (Mobile Wallet):** Picked for its cross-platform capabilities (iOS/Android from one codebase), rich UI toolkit, good native performance, and available crypto/secure storage libraries, making it efficient for developing the user-facing wallet.
 - **Figma (EHR UI Design):** An industry-standard design tool used for its collaborative features and ability to create interactive prototypes, allowing visualization and iteration of the EHR web interface before coding.
- **Workflow Definition:** The two primary use cases (Record Issuance and Record Sharing) were mapped out (Section 4.4.1).

3.2 User Interface (UI) Design

Significant effort was dedicated to designing user-friendly interfaces for both the patient and hospital user experiences:

- **Mobile Wallet UI (Flutter Implementation):**
 - A cross-platform mobile application was developed using Flutter to serve as the patient's primary interface with the Med-D system.
 - Key screens were implemented, including: User Registration/Login, Personal Information Form, Profile Display (showing the patient's Wallet DID), DID Sharing Screen (displaying the Wallet DID and a QR code), Credentials Screen (listing received medical records), and detailed Medical Record View screens
 - The design prioritized ease of use for patients to manage their identity and access their verified medical records securely on their own device. Secure storage mechanisms (e.g., Flutter Secure Storage, Section 2.2.5.1) were incorporated conceptually for handling sensitive data like private keys, although full key management was outside the prototype scope.
- **Hospital Integration Website UI (Figma Prototype):**
 - A user interface prototype for the hospital-side interaction was designed using Figma
 - This design conceptualized how hospital staff (doctors, nurses, administrators) would interact with the Med-D system, potentially integrating with an existing EHR.
 - Key designed workflows included viewing patient information associated with a DID, initiating the process to issue a new medical record through the Med-D API, and receiving/displaying verified records shared by a patient via the API's sharing workflow.
 - While not implemented as a functional web application in this phase, the Figma prototype served as a crucial blueprint for understanding EHR integration requirements and designing the necessary API endpoints.

3.3 Algorithm Design and Core Logic

Key algorithms and logic were designed and implemented for core Med-D functions:

- **DID Generation (Agent):** Deterministic did:med-d: generation based on SHA-256 hash of the public key.
- **Proof Generation (API):** SHA-256 hash generation (generateProof) on canonicalized JSON representation of the medicalRecord.
- **Agent Ledger Update (Agent):** Logic (updateProofArray) to append the record hash string to the proof array for a given DID in mock-blockchain.json.
- **Identity Verification (Agent):** Logic (verifyCredentialIdentity) comparing DID and Public Key against the mock ledger (used for API key retrieval flows).
- **Hash Verification (Agent):** Logic (checkHashExistsForDid) checking for the existence of a specific record hash string within a DID's proof array on the mock ledger (used for the record sharing flow).
- **Record Integrity Verification (API):** The algorithm for /verify-and-share-record comparing the Agent-confirmed hash with a recalculated hash of the received inner medical record.

3.4 Programming and Implementation

The system components were implemented:

- **Backend Services (API, Agent, Mock EHR Target):** Developed using Node.js/Express.js, SQLite, Axios, crypto, fs, as previously described.

- Mobile Wallet Application: Implemented using the Flutter framework and Dart programming language. This involved building the UI components corresponding to the designs , managing local state, and implementing logic to interact with the Middleware API for functions like DID registration (sending public key) and potentially initiating record sharing (sending stored records + target DID - simulated via Insomnia for end-to-end backend tests). Secure storage packages were explored conceptually.

Chapter 4

Results

4.1 Med-D Ecosystem Architecture

We designed a decentralized medical data management ecosystem to enhance the sharing and management of medical information across hospitals. Our work involved setting up the system architecture, using blockchain technology and decentralized identifiers (DIDs) to verify identities and records without storing actual medical data on the blockchain. We developed a mobile wallet application that stays with the patient and stores their medical records securely, giving them full control over how and when to share their data. On the hospital side, we built an EHR web interface that allows staff to display, receive, and send medical records, as well as connect with the hospital's existing EHR system. The system also includes an API layer that handles communication between components for processing record transfers, and an agent that connects to the blockchain to verify proofs. This decentralized setup supports secure, privacy-focused, and patient-centric data exchange between patients and hospitals.

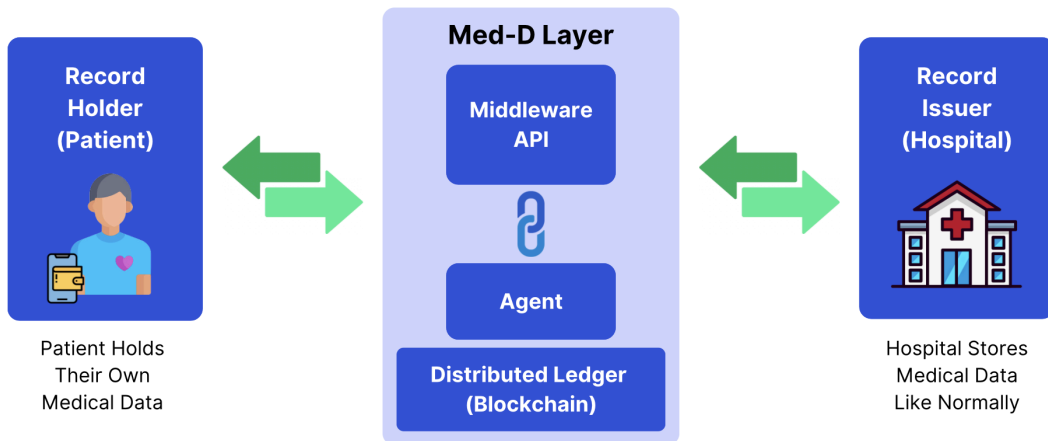


Figure 5: Med-D Ecosystem Infographic

Figure 5 illustrates how each component communicates within the Med-D ecosystem. The following sections provide a detailed explanation of each component and its role in the system.

4.2 Mobile Wallet Application

The mobile wallet app enables patients to store their verified medical records directly on their devices, providing them with full control over their data and eliminating the need for manual record transfers when moving between hospitals. The app incorporates encryption features to secure sensitive information on the user's device and during interactions with ecosystems, ensuring data privacy and security.

The image shows two side-by-side mobile app screens. The left screen is titled 'Register' and features input fields for 'Email Address' and 'Password', a 'Sign Up' button, and a link 'Already have an account? Login'. The right screen is titled 'Sign In' and features input fields for 'Email Address' and 'Password', a 'Sign In' button, and a link 'New User? Create Account'. Both screens have a blue header bar with the title and a status bar at the top showing 'TRUE-H' and '22:40'.

Figure 6: Wallet Authentication Screens

The image shows two side-by-side mobile app screens for the 'Personal Information' form. The left screen displays a list of sections: 'Chronic Conditions', 'Surgery History', 'Insurance', 'Smoking', 'Alcohol Use', and 'Substance Use', each with a text input field. At the bottom is a 'Create Wallet' button. The right screen displays the same sections but with specific input values: 'Full Name' (text), 'National ID' (text), 'Blood Group' (dropdown menu showing 'A+'), 'Birthdate' (calendar icon), 'Email Address' (text: 'abcdef@example.com'), 'Phone Number' (text), and 'Significant Other' (text). Both screens have a blue header bar with the title and a status bar at the top showing 'TRUE-H' and '22:44'.

Figure 7: Wallet Personal Information Form

When a patient creates a new wallet, they are required to register with their email and password, which serve as an authentication layer for added security (Figure 6). After registration, the patient fills out a personal information form to complete their profile (Figure 7). Once this step is completed, the wallet sends a request to the Med-D layer to generate a unique wallet DID, which is then returned and stored in the wallet. When the patient later logs back into the wallet, they use the same registered email and password for authentication. Upon successful login, the wallet retrieves and displays the patient's stored information.

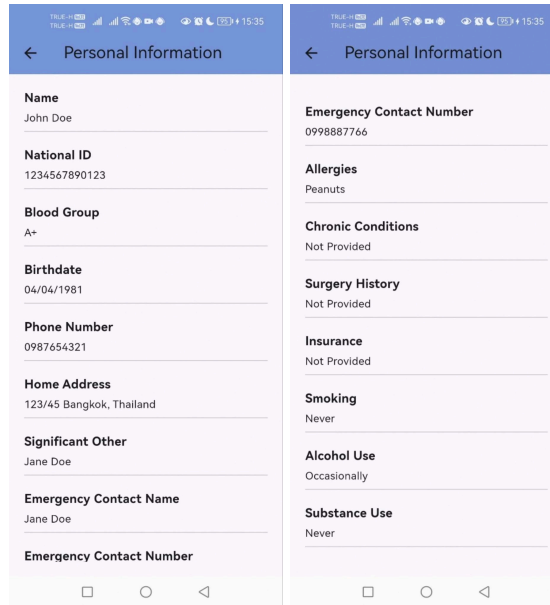


Figure 10: Wallet Display Personal Information Screens

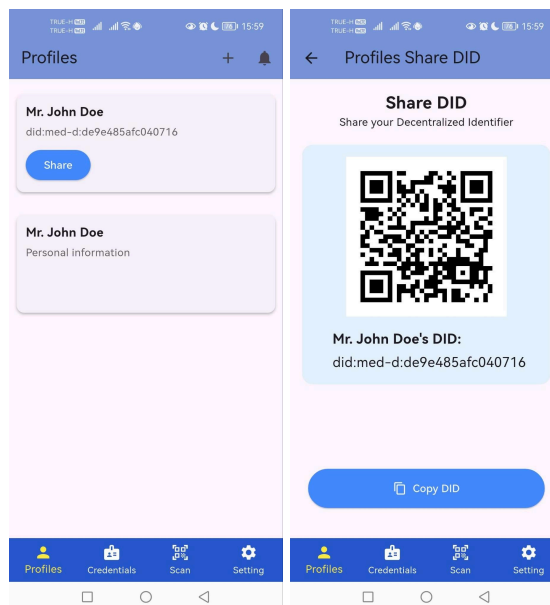


Figure 9: Wallet Share DID Screen (QR code)

On the Profile screen, the patient can view their personal information (Figure 8) and share their wallet DID to receive medical records from hospitals (Figure 9). To share the DID, the wallet generates a QR code containing the wallet's endpoint. A hospital can scan this QR code and send the medical record, along with the wallet endpoint, to the Med-D layer for processing. Once verified, the medical record is returned to the patient's wallet with proof and securely stored.

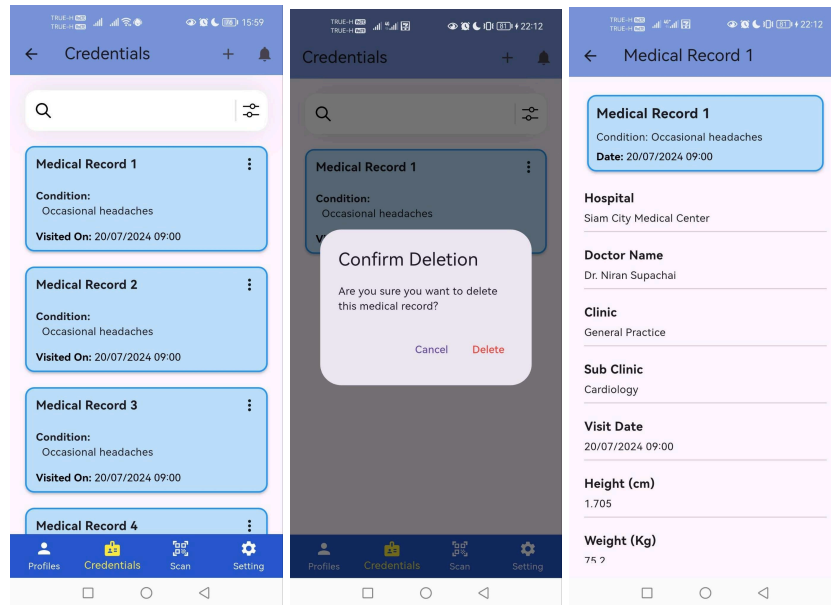


Figure 10 & 11 & 12: Wallet Manage and Display Medical Records Screens

On the Credentials screen, all medical records stored in the wallet are displayed (Figure 10). Patients can manage these records by viewing or deleting them, but they cannot edit the content to ensure the information remains secure and accurate as issued by the hospital. When a record is deleted (Figure 11), it is permanently removed from the wallet; however, the patient can request the same record again from the hospital if needed. Patients can also click on any medical record to view its detailed information (Figure 12).

On the Scan screen, the wallet is designed to include a camera feature for scanning a hospital's QR code and selecting medical records to share. Once a record is selected, the wallet sends it to the Med-D layer, which verifies its authenticity by checking the existing proof on the blockchain. If the validation is successful, the verified medical record is forwarded to the hospital. Due to time constraints, the camera functionality for scanning QR codes has not yet been implemented. However, we currently support sharing records by sending them to a predefined EHR endpoint using hardcoded values.

4.3 Hospital Website

The Med-D platform, built to support the critical work of our hospital nurses and staff, adopts a minimalist design philosophy to sharpen staff focus and optimize daily workflows. By presenting information and actions with utmost clarity and eliminating visual distractions, this approach is specifically intended to reduce cognitive load. Consequently, it significantly minimizes the likelihood of human error, especially during crucial tasks like patient information input, and ensures all essential functions are performed with greater accuracy and efficiency. (All of this was designed in Figma and still being work in progress)

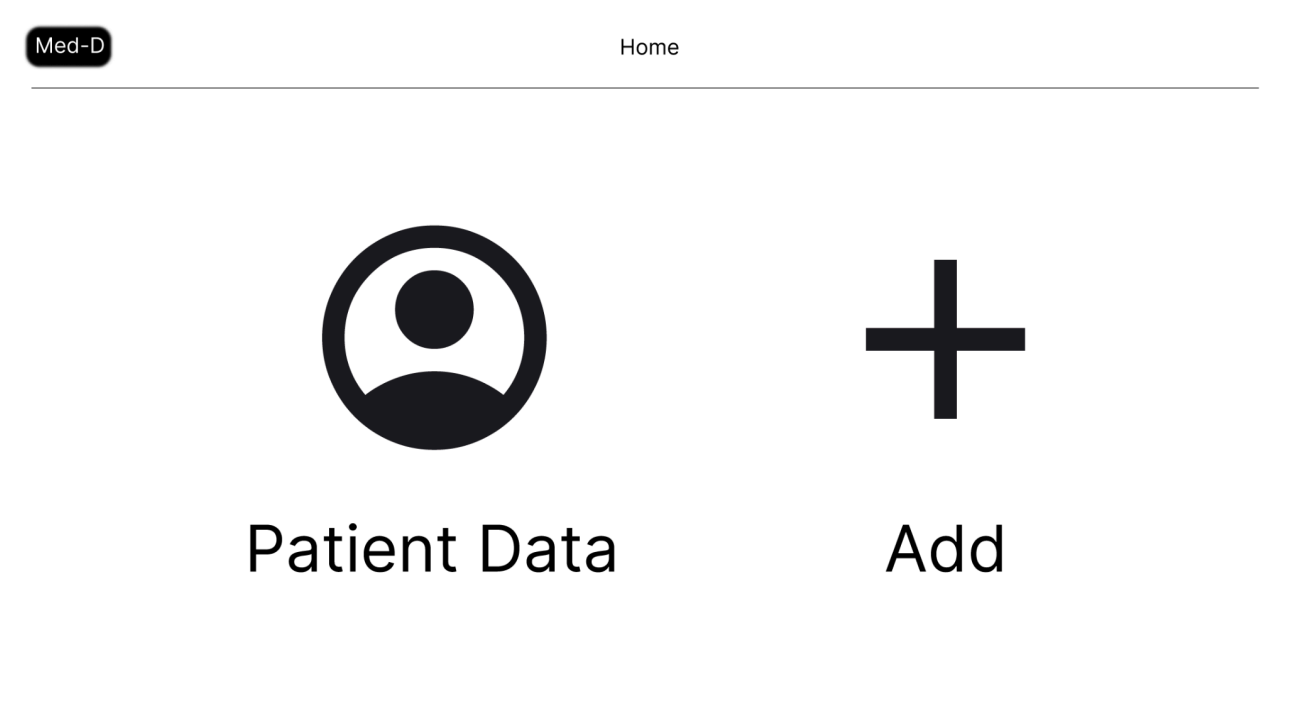


Figure 13: Hospital Website Home Screen

Further reinforcing this commitment to a focused and error-resistant workflow, the Med-D homepage is intentionally streamlined, presenting only two primary action buttons: 'Patient Data' and 'Add.' These buttons are generously sized and clearly delineated, a design choice that significantly minimizes the chance of misclicks or selection errors, guiding staff directly to their intended tasks with greater certainty.

The image shows the "Add Patient" screen in the Med-D application. At the top, there is a header bar with a dark rounded rectangle on the left containing the text "Med-D" in white, followed by a back arrow icon and the text "< Back". On the right side of the header, the text "Add" is followed by a dark plus sign icon. Below the header, the main area contains a form with five labels on the left: "First name", "Last name", "Birth Date", "Location", and "Symptom". To the right of these labels is a large, solid gray rectangular area representing the input field. At the bottom right of the screen, there are two buttons: a "Cancel" button and an "Apply" button, which is a dark rounded rectangle with white text.

Figure 14: Hospital Website Add Patient Screen

The 'Add' page serves as the dedicated area for inputting comprehensive patient data, encompassing vital details and information. This section will also integrate the ability to include a patient photograph, a feature currently under development that will be refined to align precisely with our established data schema.



Figure 15: Hospital Website Patients Data Main Screen

The 'Patient Data' page provides access to a comprehensive list of all entered patient names. We are actively enhancing this page with future quality-of-life improvements, such as a dedicated search bar and other features designed to streamline navigation and data retrieval, all of which are currently in development.



Figure 16: Hospital Website Display Selected Patient Personal Information

When viewing an individual patient's record, this page displays their complete data profile as structured and entered via the 'Add' page. This includes all essential patient information and their photograph. Additionally, positioned in the upper right, you will find three distinct icons which enable further actions or navigation, as demonstrated in the subsequent three images.

Med-D < Back

Modify

First name

Last name

Birth Date

Location

Symptom

Cancel Apply

Figure 17: Hospital Website Modify Patient Data Screen

The 'Modify' page is dedicated to allowing staff to update or correct specific elements within a patient's existing record. This includes the ability to edit various informational data fields as well as change the patient's photograph, ensuring all details can be kept accurate and current.

Med-D < Back

Delete

First name

Last name

Birth Date

Location


Symptom

Cancel Delete

Figure 18: Hospital Website Delete Patient Data Screen

The 'Delete' page provides a final review opportunity, displaying the selected patient's information and photograph before any action is taken. To prevent accidental data loss, clicking the 'Delete' button, located at the bottom right, will trigger a confirmation pop-up. This requires explicit staff verification before proceeding. Once confirmed, the patient's entire record is permanently removed from the system, and their name will consequently disappear from the 'Patient Data' list.

Med-D < Back

Medical Record 

Hospital	Vital Sign	Medical History	Diagnosis & Treatment	Additional note
Hospital: Hospital	Weight: Weight	Chief Complaint: ChiefComplaint	Diagnosis: Diagnosis	Nurse Note: Nurse Note
Doctor Name: Doctor Name	height: height	Present illness: Present illness	Treatment: Treatment	Consultation: Consultation
Clinic: Clinic	Systolic BP: Systolic BP	Physical Examination: Physical Examination	ICD10code: ICD10code	
Sub Clinic: Sub Clinic	Diastolic BP: Diastolic BP	Past History: Past History	ICD9code: ICD9code	
Visit Data: Visit Data	Temperature: Temperature			
	Pulse Rate: Pulse Rate			
	Breath Rate: Breath Rate			

Figure 19: Hospital Website Patient's Medical Record Screen

The 'Medical Record' page provides a comprehensive yet uncluttered view of all pertinent patient medical data, adhering to our minimalist design philosophy. This ensures that while the presentation is streamlined for clarity, all essential clinical information required by staff is thoroughly and accessible displayed.

4.4 Med-D Layer

4.4.1 Middleware API

The Middleware API serves as the central communication layer in the Med-D ecosystem, enabling secure and structured data exchange among the Patient Wallet, Hospital EHR Systems, and the Agent responsible for blockchain operations. Its main responsibilities include handling bidirectional transmission of medical records, generating and verifying cryptographic proofs, storing verified data, and integrating with the Agent to manage blockchain credentials. Additionally, the API enforces a secure communication foundation by performing a DID and public key (PK) exchange before every transfer, allowing the system to evolve into a more robust security architecture in the future.

4.4.1.1 Pre-transfer Handshake

Before each medical record transfer—no matter the direction—a handshake is done between all involved parties: the Wallet, API, and EHR system. During this handshake, they

exchange Decentralized Identifiers (DIDs) and Public Keys (PKs). Because of limited time, we did not implement full TLS (Transport Layer Security). Instead, we use a basic public key exchange to establish trust and prepare for encrypted communication. This setup creates a solid foundation that can be upgraded later to a more secure protocol like TLS.

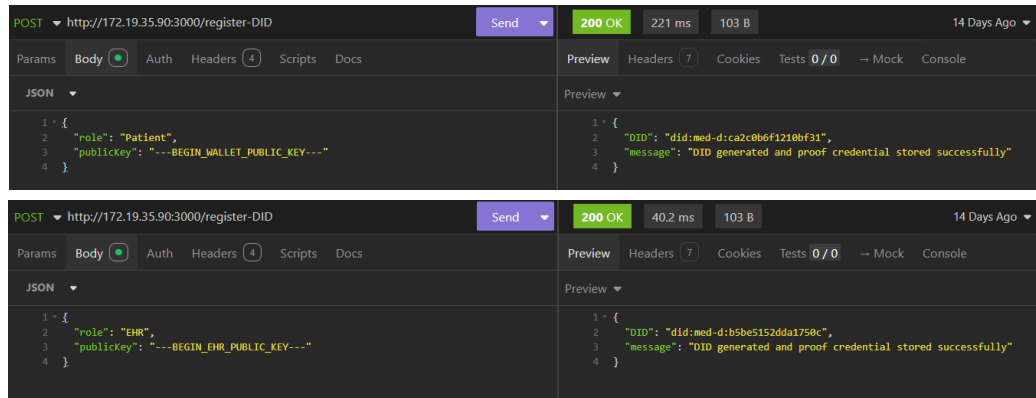


Figure 20: API Request & Response DID Registration

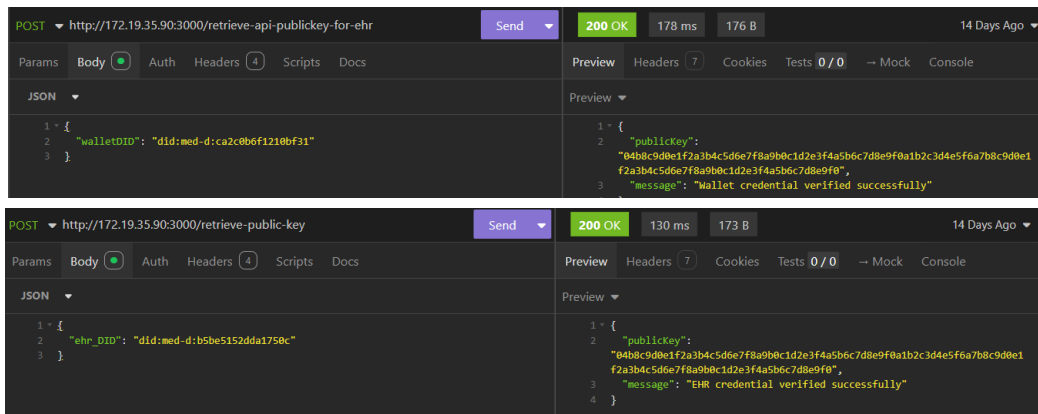


Figure 21: API Request & Response Public Key Retrieval

4.4.1.2 Scenario 1: Patient get medical records from Hospital (EHR to Wallet)

This scenario describes the process when a hospital EHR system sends a medical record to the patient's Wallet via the Middleware API.

Process Flow

1. Medical Record Submission
The Hospital EHR system sends the medical record and the patient's Wallet endpoint to the Middleware API.
2. Proof Generation
The API creates a SHA-256 hash of the medical record. This hash is combined with the patient's unique identifier to generate a tamper-evident proof.
3. Agent Interaction
The API sends the generated proof and the patient's Wallet DID to the Agent. The Agent creates a proof credential and stores it on the blockchain. This credential is then returned to the API.

4. API Data Storage

The API stores both the medical record and the received proof credential in its internal database for verification and auditing purposes.

5. Record Delivery to Wallet

The API forwards the verified medical record along with the proof to the patient's Wallet, where it is securely stored.

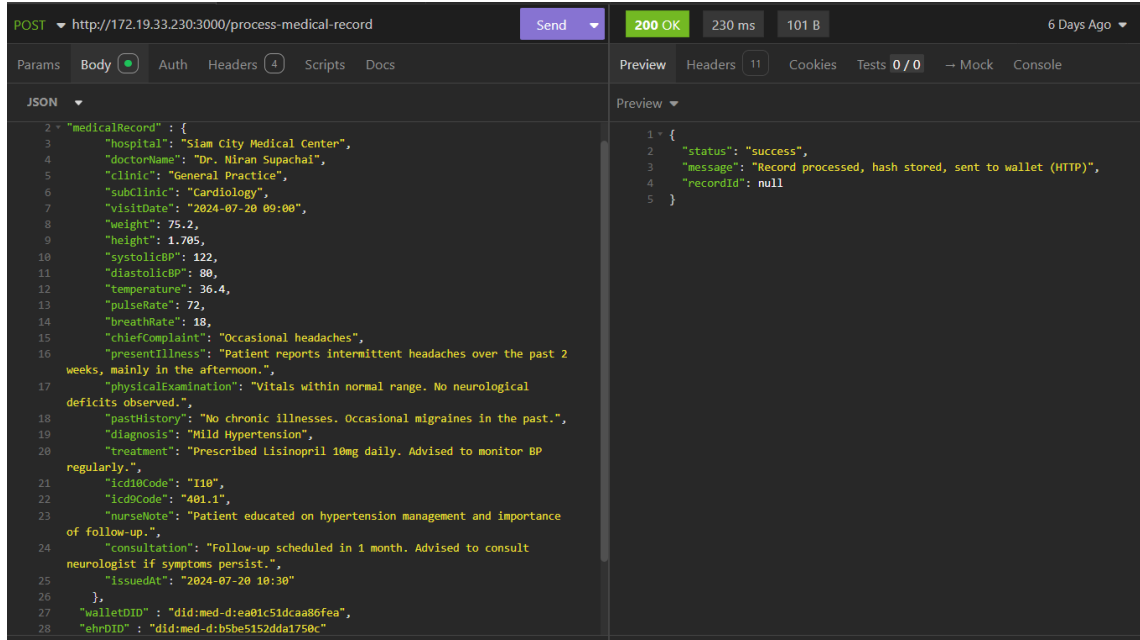


Figure 22: API Request & Response Simulated EHR Medical Record Sending

4.4.1.3 Scenario 2: Patient share medical records to Hospital (Wallet to EHR)

This scenario covers the case where a patient chooses to share a medical record from their Wallet with a hospital EHR system.

Process Flow

1. Medical Record Submission

The patient selects a medical record and shares it with a specific hospital. The Wallet sends the record and the target EHR endpoint to the Middleware API.

2. Proof Credential Retrieval

The API sends the patient's Wallet DID to the Agent and retrieves the associated proof credential.

3. Proof Verification

The API checks the submitted medical record against the proof stored in the proof credential to ensure the record's integrity and authenticity.

4. Record Delivery to EHR

If verification succeeds, the API forwards the validated medical record to the target EHR system.

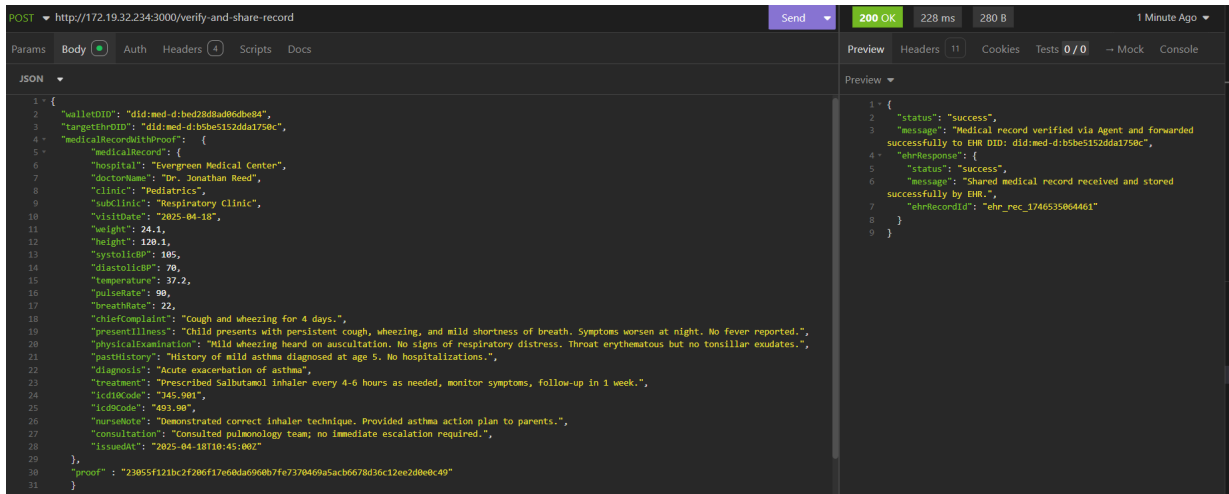


Figure 22: API Request & Response Simulated Wallet Medical Record Sending

4.4.2 Agent & Distributed Ledger (Blockchain)

We created the Med-D Agent, also known as the med-agent, to enable communications between the Middleware API & the Distributed ledger (Blockchain). med-agent mainly handles the creation or the issuance and the storage of “Proof Credentials” through HTTP POST requests from the middleware API. We have managed to work through the issuance, storing, and revocation process of credentials, however, there are some parts that are still needed to be refined and completed, such as, refactoring the code into proper HTTP POST request endpoints, along with bringing compatibility with the OpenAPI standard; enabling more possibilities of open integration with our agent.

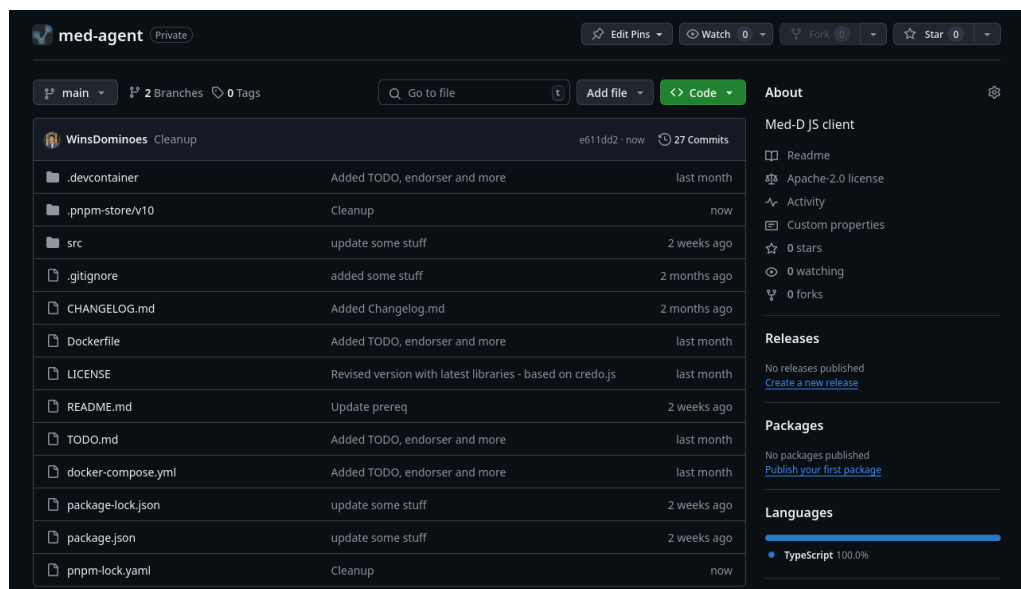


Figure 23: Med-Agent GitHub Page

Alongside our med-agent, we also ran a test setup using the indy-node-container template which comprises Hyperledger Indy-Node (the Ledger) and bcovrin, a library developed by the Government of British Columbia, Canada; that contains a test setup; spawning 4 instances of indy-node, controllers, and the bcovrin dashboard. With this, we

were able to implement our med-agent to be compatible with the indy-node ledger, through the indy-vdr library provided by the Hyperledger Team and Credo.

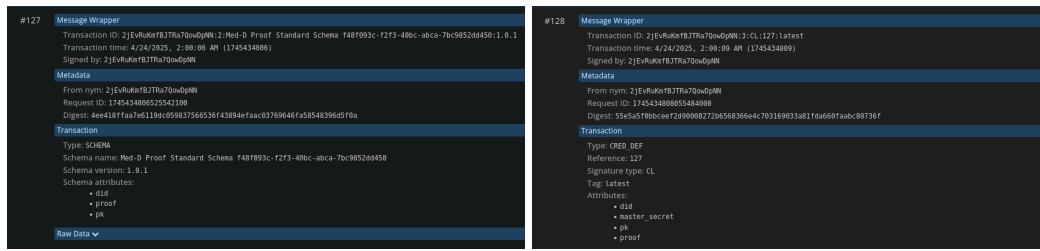


Figure 24: Indy Ledger Transactions showing credential schemas and definitions.

Figure 24 shows one of the usages of the ledger, which is to guarantee the standardization of proof-credentials, as well as their definitions which dictate specifications related to the party that can issue the proof-credential and any type of signatures that are required to issue them.

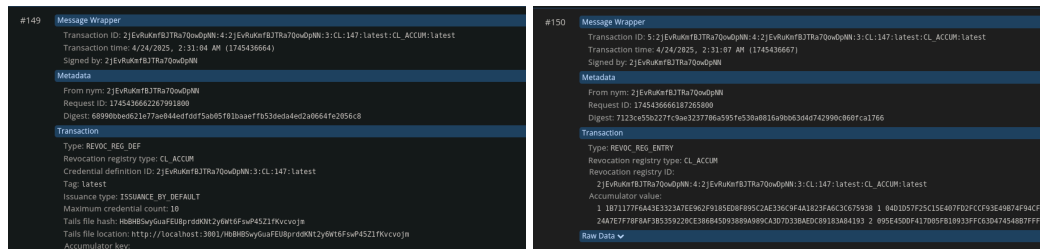


Figure 25: Indy Ledger Transactions showing credential revocation.

Figure 25 displays the second use of the ledger, which is to handle the revocation of proof-credentials. This process is needed for when a patient's medical record gets issued, the proof will be needed to generate from those medical records, which is then published to be put on the ledger through the med-agent; as proof-credentials are used to store proof-data, when new data is added, the old record must be invalidated.

Chapter 5

Conclusion

5.1 Summary of Accomplishments

We've poured our energy into creating something we believe can truly make a difference in people's lives: a system designed with you, the patient, at its very core. It thoughtfully harnesses the robust security and transparent nature of blockchain technology to safeguard your vital medical information. But beyond just secure storage, we've re-imagined how this crucial data travels when you need it most. Imagine needing to share your medical history with a new specialist or another hospital. Instead of the often frustrating delays, misplaced faxes, or cumbersome paperwork, our system empowers you to use your own smartphone as a secure digital wallet for your health records. This approach isn't just about cutting-edge tech; it's about making your healthcare journey smoother, noticeably quicker, and significantly reducing the chances of those small human errors that can occur when information is manually handled multiple times. It's about giving you more control and peace of mind when it comes to your own health information.

5.2 Issues and Obstacles

Building this system was definitely a journey, and like any ambitious project, it threw its share of curveballs our way. There were moments where we realized the path forward required skills or knowledge we hadn't initially anticipated, sending us back to the drawing board, eager to learn and adapt. We definitely hit roadblocks with errors that, at first glance, seemed baffling – those frustrating times where the solution isn't immediately obvious and requires real persistence and collaborative head-scratching to overcome. For instance, integrating with complex, existing healthcare infrastructures meant some of our team members had to dedicate significant extra time and deep focus to truly master the intricacies of specific APIs, like those for (EHR) systems or navigating the specific requirements of the Med-D layer. It wasn't always easy, and it certainly tested our problem-solving skills, but each challenge ultimately strengthened both our team and the final system.

5.3 Future Directions

Our journey with this system is far from over; in fact, we're now embarking on an incredibly exciting and crucial next phase. We're committed to meticulously refining and perfecting what we've built. To do this, we plan to partner with a select group of 2-3 hospitals for an initial, real-world deployment. This isn't just about testing the technology; it's about listening – truly listening – to the experiences of the doctors, nurses, and administrative staff who will interact with it daily. Their on-the-ground feedback will be invaluable, providing us with the insights we need to make meaningful improvements and ensure the system is not just functional, but genuinely helpful and intuitive. Once we've carefully incorporated these learnings and are confident that our system is robust, user-friendly, and truly making a difference, our vision is to roll it out across every hospital in Thailand. And we don't plan to stop there; our long-term aspiration is to expand its reach, bringing the benefits of more secure and streamlined medical data management to other regions, helping to build a more connected and efficient healthcare ecosystem for everyone.

5.4 Lessons Learned

Looking back, the journey certainly wasn't without its share of hurdles. We absolutely encountered plenty of bumps in the road – unexpected problems that sometimes threatened to derail

our progress and moments where we stumbled into areas completely new to us, requiring knowledge we didn't initially possess. These weren't just minor inconveniences; they genuinely challenged our workflow and forced us to pause and re-evaluate. But facing these challenges head-on proved invaluable. It compelled us to think outside the box, embrace trying new approaches even when we weren't sure they'd work, and develop a real skill for troubleshooting and implementing fixes rapidly, often under real pressure. In truth, it's precisely because we navigated those difficulties, learned to adapt on the fly, and collectively pushed through the uncertainty that our system has become the capable solution it is today. It's a direct result of the team's resilience and our shared willingness to learn and evolve throughout the entire process.

References

- Dart packages. "Flutter_secure_storage | Flutter Package." Accessed December 12, 2024. https://pub.dev/packages/flutter_secure_storage.
- "Hyperledger Indy — Hyperledger Indy 1.0 Documentation." Accessed December 12, 2024. <https://hyperledger-indy.readthedocs.io/en/latest/>.
- "Hyperledger/Anoncreds-Rs." Rust. 2022. Reprint, Hyperledger, October 24, 2024. <https://github.com/hyperledger/anoncreds-rs>.
- Insights, Ledger. "Hyperledger Launches New Digital Identity Project, AnonCreds." Ledger Insights - blockchain for enterprise, November 15, 2022. <https://www.ledgerinsights.com/hyperledger-digital-identity-anoncreds-verifiable-credentials-privacy/>.
- Levine, Hallie. "Getting Your Medical Records Is Too Hard, a New Study Finds." Consumer Reports, October 5, 2018. <https://www.consumerreports.org/hospitals/getting-access-to-medical-records-is-too-hard/>.
- Miller, Ronald V. and Jr. "Medical Records Errors Are Killing People." Maryland Medical Malpractice Attorney Blog, December 14, 2018. <https://www.marylandmedicalmalpracticeattorneyblog.com/medical-records-errors-are-killing-people/>.
- Preukschat, Alex, and Drummond Reed. Self-Sovereign Identity, 2021. <https://learning.oreilly.com/library/view/self-sovereign-identity/9781617296598/>.
- "Self-Sovereign Identity (SSI): Autonomous Identity Management | Okta." Accessed December 12, 2024. <https://www.okta.com/identity-101/self-sovereign-identity/>.
- "Self-Sovereign Identity: The Ultimate Guide 2024." Accessed December 12, 2024. <https://www.dock.io/post/self-sovereign-identity>.
- "Thailand Telehealth Market Size And Share | Report, 2030." Accessed December 12, 2024. <https://www.grandviewresearch.com/industry-analysis/thailand-telehealth-market-report>.
- Thomas, Joseph. "Medical Records and Issues in Negligence." Indian Journal of Urology : IJU : Journal of the Urological Society of India 25, no. 3 (2009): 384–88. <https://doi.org/10.4103/0970-1591.56208>.
- "Verifiable Credentials Data Model v1.1." Accessed December 12, 2024. <https://www.w3.org/TR/vc-data-model/>.
- "Verifiable Credentials: The Ultimate Guide 2024." Accessed December 12, 2024. <https://www.dock.io/post/verifiable-credentials>.
- Academic Expert. "Academic Expert." *National Center for Biotechnology Information*. Accessed July 3, 2025. <https://pmc.ncbi.nlm.nih.gov/articles/PMC11836587/>.

Bakar, Nizam Abdul, et al. "Technical Barriers to Health Information Exchange in Developing Countries: A Systematic Review." *Journal of Multidisciplinary Healthcare* 15 (2022): 2383–2394. Accessed July 3, 2025. <https://pmc.ncbi.nlm.nih.gov/articles/PMC9635045/>.

Jirayuthjarus, Suwichaya, et al. "Patient-Centric Health Records in Thailand: Building Interoperability Through Blockchain and Self-Sovereign Identity." *Journal of Medical Internet Research* 27, no. 1 (2025): e58338. Accessed July 3, 2025. <https://www.jmir.org/2025/1/e58338>.

University of Minnesota. *Patient Health Record Systems in Thailand*. Accessed July 3, 2025. <https://conservancy.umn.edu/items/e0deb05a-e2a9-4ae1-9ea3-994c0336b355>.

Ministry of Public Health, Thailand. *How to Use PHR Viewer on MohPrompt*. August 26, 2022. Accessed July 3, 2025. https://mohprompt.moph.go.th/mpc/wp-content/uploads/2022/08/Howto_PHR_Viwer_onMohPrompt-Station_V1.26082022.pdf.

Tilleke & Gibbins. "Digital Health in Thailand, Vietnam, and Indonesia." Accessed July 3, 2025. <https://www.tilleke.com/insights/digital-health-in-thailand-vietnam-and-indonesia/>.

McCoy, Allison B., et al. "Duplicate Patient Records in Electronic Health Records." *Journal of the American Medical Informatics Association* 20, no. 1 (2013): 141–147. Accessed July 3, 2025. <https://pmc.ncbi.nlm.nih.gov/articles/PMC3540536/>.

Wani, Hafeez, et al. "Adoption of Electronic Health Records in Developing Countries." *JMIR Medical Informatics* 9, no. 1 (2021): e19590. Accessed July 3, 2025. <https://pmc.ncbi.nlm.nih.gov/articles/PMC7761950/>.

Atherly, Adam, et al. "Understanding the Role of Health Insurance in Health Care Access in Thailand." *International Journal of Health Policy and Management* 9, no. 4 (2020): 147–157. Accessed July 3, 2025. <https://pmc.ncbi.nlm.nih.gov/articles/PMC7110812/>.