Purely: Blockchain-based decentralized platform for patient centric healthcare data access.

Poojan Sheth 2024H1030081P, Priyank Shah 2024H1030092P
Department of Computer Science, BITS Pilani

Abstract

The digital health landscape is still fragmented, with centralized electronic health record (EHR) systems restricting patient agency, interoperability, and secure data exchange. This article suggests Purely, a blockchain-based decentralized healthcare platform that gives patients power via Decentralized Identifiers (DIDs) and verifiable credentials. By combining Ethereum smart contracts, IPFS for off-chain encrypted data storage and a React-based frontend combined with MetaMask authentication, the system allows hospitals to give patients control over access to their medical records and monitor interactions in a verifiable way. Hospitals provide encrypted, digitally signed treatment credentials, and insurers validate claims exclusively based on verified DIDs—preserving privacy while not losing trust. The architecture provides end-to-end encryption and zero-trust principles to enable complete auditability, revocability, and privacy compliance. The system is tested on the Ethereum Sepolia Testnet and proves real-time, permissioned sharing of patient health records and insurance claim business processes. Results verify that a DID-first architecture provides a privacy-preserving and scalable alternative to conventional health information exchanges (HIEs).

Index Terms

Blockchain, Decentralized Identity, Ethereum, Electronic Health Records (EHR), IPFS, Smart Contracts, DID, Verifiable Credentials, Patient-Centric Healthcare, Self-Sovereign Identity, Health Information Exchange (HIE), MetaMask, Secure Data Sharing, Privacy-Preserving Claims.

I. INTRODUCTION

Over the past few years, the digitization of medical records has transformed how patient information is obtained, kept, and retrieved. Yet, current Electronic Health Record (EHR) systems are still mostly centralized, proprietary, and institution-centric, resulting in enormous interoperability, data siloing, privacy violations, and patient disenfranchisement from decision-making. Notwithstanding international initiatives towards standardization, present EHR architectures still do not have mechanisms which preserve patient sovereignty, medical interaction verifiability, and secure cross-organizational record sharing.

Technologies like blockchain, Decentralized Identifiers (DIDs), and verifiable credentials hold the promise of a paradigm shift to patient-centric health data systems. Cryptographic primitives and decentralized identity models allow patients to possess, control, and share medical information without depending on a single authority. Additionally, smart contracts on blockchain platforms allow programmable, auditable enforcement of access control, insurance claims, and consent management.

This work presents Purely, a decentralized Electronic Health Record management system that addresses the shortcomings of traditional architectures. The system features:

- A DID-based identity layer for user-managed identity control;
- IPFS-based encrypted off-chain storage of patient records;
- Ethereum smart contracts for handling identity registration, policy issuance, and insurance claims in a transparent, tamper-proofing way;
- A React-based frontend with MetaMask and ethics.js integration for key generation, encryption, digital signatures, and safe authentication.

In Purely, patients are issued a DID at registration, which they then use to ask hospitals for treatment and insurers for claims. Everything that is personal—medical history and diagnosis—is end-to-end encrypted using the public key of the recipient, which is obtained from their DID. Physicians can issue verifiable treatment credentials that are cryptographically signed and stored on-chain without disclosing private or medical data to third parties. Insurers are confirmed with the patient and doctor DIDs only, assuring privacy-preserving communication.

This paper proves the conceptual feasibility of a self-sovereign, secure, and privacy-preserving healthcare data management system by deploying and testing the full-stack solution on the Ethereum Sepolia Testnet. Through extensive evaluation, we exemplify how decentralized identity and smart contract infrastructure can enable privacy-compliant healthcare workflows, secure multi-party communication, and trustless coordination of patients, hospitals, and insurers.

II. LITERATURE REVIEW

Over the last few years, noteworthy progress has been made towards decentralized and patient-directed healthcare data management, with a focus on blockchain technology and decentralized identifiers (DIDs). Conventional electronic health record (EHR) systems tend to function in silos, which results in fragmented delivery of healthcare and restricts data interoperability. This results in data silos that prevent timely and comprehensive access to patient records, especially between institutions [79†ooaa073.pdf].

Khurshid et al. presented MediLinker, a blockchain-powered identity and consent management system, to solve patient-centered issues in healthcare. MediLinker uses Hyperledger Indy and Aries to provide decentralized identity (DID) functionality, allowing patients to control verifiable credentials (VCs) and exchange them securely between health systems [80†MediLinker_a_blockchain-based_decentralized_health.pdf]. Their contribution highlighted the usefulness of self-sovereign identity (SSI) solutions in supporting fine-grained control of data, consent-driven sharing, and interoperability in accordance with the 21st Century Cures Act.

Another critical viewpoint is stressed by Khurshid et al. in their work published in 2021, where they developed and implemented MediLinker in modeled healthcare settings. The results emphasized that blockchain can facilitate patients to hold possession of their health data, dynamically authorize consent, and engage securely with multiple stakeholders like physicians and research clinics without successive identification checks. Their research formally proved dominant criteria for identity management—Autonomy, Authority, Availability, Approval, Confidentiality, and Interoperability—in terms of Bouras et al. [79†ooaa073.pdf].

To complement these papers, current literature also invariably finds blockchain's inherent strengths in the healthcare setting, such as tamper-evident audit trails, distributed control of data, and cryptographic privacy [80†MediLinker_a_blockchain-based_decentralized_health.pdf]. They do, however, also identify current challenges such as regulatory compliance, user interface, and adoption in real-world clinical settings.

In addition, privacy technologies (PETs) like encryption and digital signatures—typically combined through cryptographic key pairs (e.g., Ed25519 signing and X25519 encryption)—are becoming increasingly prevalent in secure EHR transfers. These form the basis of DIDs, where patients and providers communicate pseudonymously yet maintain verifiability and privacy, such as in the DID-generating modules used in our present research infrastructure.

In general, the literature reviewed above points towards the imperative necessity of systems incorporating blockchain and DID to make secure, patient-controlled, and interoperable health ecosystems. Our work draws on those foundations by creating and integrating a DID-based platform specific to real-time encrypted communication among patients, physicians, and payers in a manner that guarantees selective disclosure and regulatory compliance in a self-sovereign system.

III. METHODOLOGY

This study takes a decentralized approach in addressing the weaknesses of conventional electronic health record (EHR) systems through the use of blockchain, decentralized identifiers (DIDs), and verifiable credentials. Every user—whether patient, physician, or insurer—starts by creating a cryptographic key pair based on Ed25519 for digital signatures and X25519 for encryption. These keys serve as a basis to derive a DID via the did:key method according to W3C standards. The DID is a user's worldwide, verifiable identity and is saved alongside its public keys on a backend server, with private keys safely stored on the user's device.

After identity creation, individuals sign up for the blockchain through Ethereum smart contracts. The contracts impose identity type constraints and link each address with its appropriate DID. Insurers are empowered to construct healthcare policies on-chain, taking along metadata like premium amount, coverage cap, and term. Patients are able to search and buy these policies using MetaMask transactions, which are stored immutably on the Ethereum Sepolia Testnet.

To facilitate private data exchange, all the health records are client-side encrypted using the recipient's public encryption key prior to being uploaded onto the InterPlanetary File System (IPFS). In this way, only the doctor or insurer for whom the record is intended can decrypt it, and the IPFS hash (content identifier) is stored on-chain for tamper-proof referencing. The patients have complete control of who can access their records, issuing or revoking permissions by way of smart contract calls.

Physicians access and decrypt the patient's encrypted file with their private key, then provide a diagnosis encrypted by the patient's public key for safe return. A claimable insurance claim can be started, citing only the patient's DID, the IPFS hash of the diagnosis file, and the applicable policy ID. Such claims are settled on-chain, the doctor and patient's DIDs being verified by the DID registry by the insurers. When the policy is in force and conditions of claim are fulfilled, payment is made through the smart contract to the healthcare provider or doctor, with transparency and accountability without revealing confidential patient details.

Along this process, all the cryptography tasks like key generation, encryption, and signing take place on the client-side with JavaScript libraries coupled with the ethics package. MetaMask is used to sign transactions and manage Ethereum identities in an effortless user interface. The method provides end-to-end privacy, audibility, and decentralized control of medical records, essentially relocating trust from central custodians to a direct position in the hands of users.

IV. SYSTEM ARCHITECTURE

The system architecture of the suggested system includes a layered stack integrating decentralized identity, encrypted data storage, automated smart contract, and user interface to obtain a secure and interoperable EHR system.

Identity management is actually achieved at the core level through decentralized identifiers. Each DID is cryptographically derived from a key pair and is a globally unique reference to a user without needing a centralized authority. These DIDs are resolved and verified against public keys enrolled in a backend JSON data structure and are called out of Ethereum smart contracts. Authentication and authorization are done in a decentralized cryptographically verifiable way due to the utilization of DIDs.

Neither patient data nor doctor-issued diagnoses are ever stored on the blockchain. Rather, they are encrypted client-side with the recipient's public key and stored on IPFS, a peer-to-peer content-addressable storage system. The resulting content identifier (CID) is unalterable and serves as an impervious pointer to the file, which is used in smart contracts without revealing the contents of the file. This off-chain storage system keeps gas fees drastically lower and in regulatory compliance with laws like HIPAA and GDPR.

Smart contracts on the Ethereum Sepolia Testnet manage all on-chain interactions such as user registration, policy creation, insurance purchase, access grants, and claims processing. These contracts output logs and events to enable transparency and enable future audit of interactions. Role-based access is implemented by mappings within the contract, and only users having valid DID-linked roles can perform certain actions.

It's developed using React.js and is completely integrated with MetaMask to make it possible to interact with Ethereum hassle-free. Once registered, the interface facilitates users to create keys, derive DIDs, and send transactions. The entire sensitive information, such as files and messages, goes through client-side encryption libraries. Visual output such as user dashboards, file upload notifications, and claim status monitoring are integrated into the system to help users navigate the system with ease. Screenshots of the application, such as the registration screen (home.png) and file upload page (logo-1.png), illustrate the user-driven design and usability of the system.

By combining DID-based identity, verifiable off-chain storage, and smart contract logic in a decentralized stack, the system architecture ensures privacy, trust, and transparency—foundational concepts for any future-gen healthcare data platform.

V. PROPOSED SOLUTION

The system proposed here presents a Decentralized Identity (DID)-based Electronic Health Record (EHR) platform that relies on blockchain and smart contracts to ensure secure, patient-held health data exchange, and IPFS (InterPlanetary File System) to store encrypted off-chain. The solution provides a solution for the root challenges in contemporary healthcare—interoperability, privacy, consent control, and real-time verifiability—by leveraging Ethereum smart contracts, DID-based identity management, and public-key cryptography.

A. System Overview

The architecture has three major players: patients, hospitals, and insurance companies, all of which have a cryptographically created DID. Patients create a DID by verifying their identity and obtaining a signed credential from a known clinic. These DIDs are onboarded on-chain through smart contracts, which guarantees global uniqueness and verifyability without relying on centralized identity providers.

Hospitals, when they diagnose a patient, encrypt the diagnosis with the patient's public encryption key and sign the content with their own private signing key. Signed, encrypted data is stored in IPFS, and just the IPFS content identifier (CID) is maintained on-chain. Data privacy, integrity, and provenance are guaranteed in this manner.

B. Credential Issuance and Sharing

Credential issuance is in a verifiable credential model: when the patient registers at a hospital, the hospital signs a treatment credential with its private key. The credential, after being encrypted with the public key of the patient, is distributed off-chain to IPFS. Only the patient is able to decrypt it with their private key. The CID of the credential along with some associated metadata (e.g., doctor DID, patient DID, timestamp, and policy ID) are kept on-chain for purposes of maintaining auditability.

Hospitals and insurers authenticate with W3C-compliant DID resolution and cryptographic signature verification. This ensures that even when the hospital does not have the patient's actual-world data, trust is established through verifiable DID credential.

C. Access Control and Revocation

The patient is still the exclusive owner of their data. Medical record access is granted or taken away through modification of access control lists held in smart contracts. When revoked, hospitals and insurers will not be able to confirm the current state of the credential using the modified registry, thus implementing denial-of-access.

For instance, when a patient withdraws access to their records from a hospital or insurer, the smart contract invalidates the DID relation. Subsequently, any claim derived from that information becomes non-verifiable, enforcing zero-trust architecture and real-time consent.

D. Insurance Claim Lifecycle

Once treated, the hospital can submit a claim to the insurer using the patient's active policy ID and signed diagnostic report. The smart contract forwards the claim to the insurer, who decrypts the claim if authorized and verifies it with the issuing hospital's DID signature. If the claim is compliant with policy terms, the insurer makes a payment through the smart contract. If denied, the patient can be obliged to pay the amount, similarly monitored transparently on-chain.

This policy modularity in enforcement and traceability is realized through Solidity-based smart contracts on the Ethereum testnet (e.g., Sepolia) such that all transactions—be they a claim, a purchase of a policy, or a revocation—are immutably recorded and available for compliance and auditing.

E. Data Privacy and Security

All treatment and diagnosis records are encrypted with asymmetric encryption (X25519) wherein the hospital (sender) encrypts the data with the patient's public key. This guarantees end-to-end confidentiality. Additionally, digital signatures (Ed25519) from the issuing party ensure data authenticity and non-repudiation.

No personally identifiable information (PII) is ever stored on-chain; rather, only DIDs and encrypted data pointers (CIDs) are stored. This method reconciles the openness of blockchain with the privacy requirements of sensitive health information.

F. Implementation Details

The frontend is developed in React.js, paired with MetaMask for interactions based on DID and digital signatures. Smart contracts are written on the Ethereum Sepolia Testnet, while decentralized file storage is via IPFS. Creation of DIDs, generation of signatures, and encryption are managed via JavaScript libraries (e.g., ethr-did, did-jwt, eth-sig-util). There is a server-side Node.js component that manages secure key storage and DID life-cycle operations.

VI. EXPERIMENTATION DETAILS

The implementation of the proposed DID-based EHR system was carried out using a combination of decentralized technologies: Ethereum smart contracts for on-chain logic, IPFS for encrypted off-chain storage, and a React-based frontend integrated with MetaMask and cryptographic libraries for decentralized identity (DID) operations.

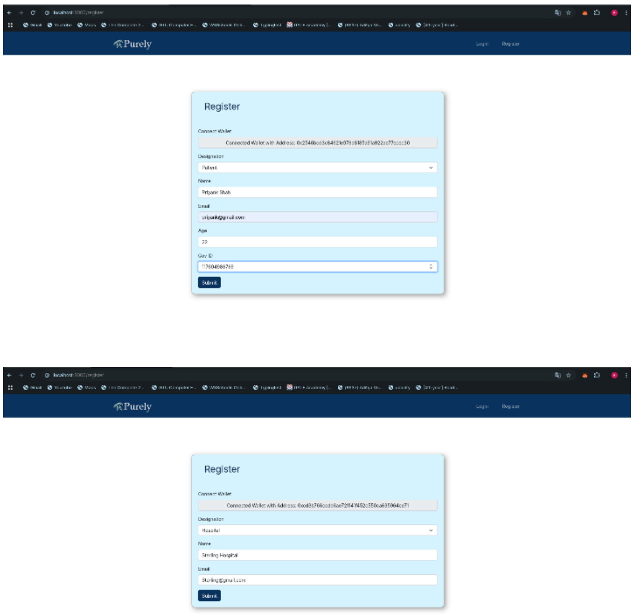
A. Development Stack

Smart Contracts: Written in Solidity and deployed using the Hardhat framework.
Frontend: Developed in React.js, interfaced with Ethereum using ethers.js.
Wallet Integration: MetaMask for identity and transaction signing.
DID & Cryptography: JavaScript-based key generation and DID derivation using ethics library (Ed25519 for signing, X25519 for encryption).
Backend: Node.js server for storing DIDs and providing key management endpoints.
Storage Layer: IPFS for storing encrypted files and diagnostic documents.

B. Modules Implemented

User Registration and DID Generation:

Patients, doctors, and insurers register with names, emails, and generate a DID via ethics on the frontend.



Register

Connect Wallet: Connected Wallet with Address: 0x2546bcd3c84621e976d8185a91a922ae77ecec30

Designation: Patient

Name: Priyank Shah

Email: priyank@gmail.com

Age: 22

Gov ID: 117604086769

Register

Connect Wallet: Connected Wallet with Address: 0xcd3b766ccdd6ae721f141452c550ca635964ce71

Designation: Hospital

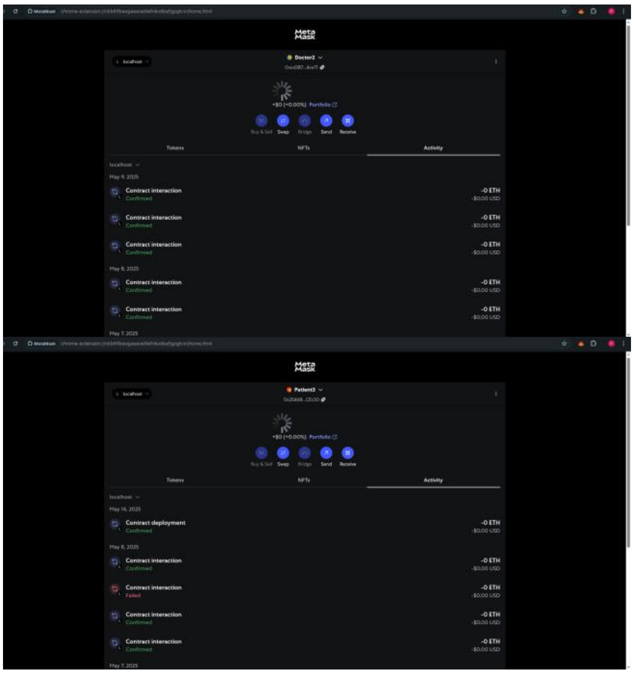
Name: Sterling Hospital

Email: Sterling@gmail.com

Login/Registration Page UI

Policy Creation and Purchase:

- Insurers create policies on-chain.



May 9, 2025

Contract interaction Confirmed -0 ETH -$0.00 USD

Contract interaction Confirmed -0 ETH -$0.00 USD

Contract interaction Confirmed -0 ETH -$0.00 USD

May 8, 2025

Contract interaction Confirmed -0 ETH -$0.00 USD

Contract interaction Confirmed -0 ETH -$0.00 USD

May 7, 2025

May 14, 2025

Contract deployment Confirmed -0 ETH -$0.00 USD

May 8, 2025

Contract interaction Confirmed -0 ETH -$0.00 USD

Contract interaction Failed -0 ETH -$0.00 USD

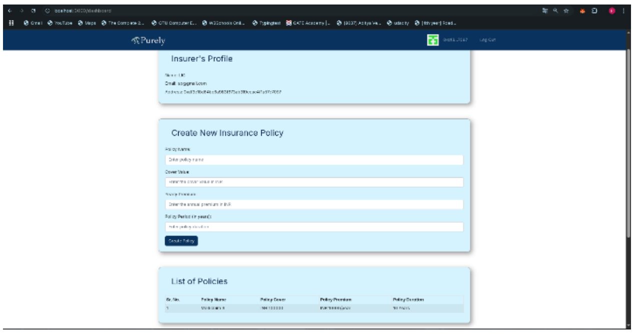
Contract interaction Confirmed -0 ETH -$0.00 USD

Contract interaction Confirmed -0 ETH -$0.00 USD

May 7, 2025

Metamask Image

- Patients can view and purchase policies using ETH from MetaMask.



Insurer's Profile

Name: LIC

Email: lic@gmail.com

Create New Insurance Policy

Policy Name: Enter policy name

Cover Value: Enter the cover value in INR

Yearly Premium: Enter the annual premium in INR

Policy Period (in years): Enter policy duration

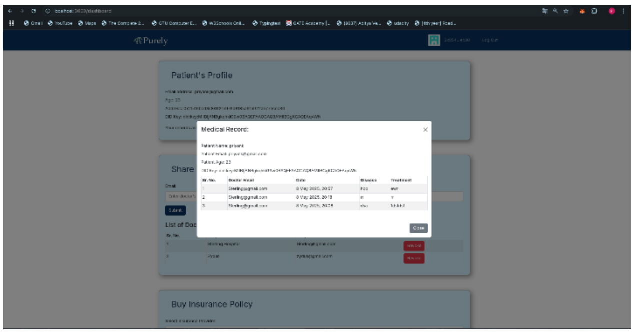
List of Policies

Sr. No.	Policy Name	Policy Cover	Policy Premium	Policy Duration
1	[illegible]	[illegible]	[illegible]	[illegible]

INSURANCE PORTAL

Medical Record Upload:

- Patient encrypts the file using the doctor’s public encryption key (DID-based).
- File is uploaded to IPFS.



Patient's Profile

Age: 22

Medical Record:

Patient name: priyank

Patient Age: 22

Sr. No.	Doctor Email	Date	Disease	Treatment
1	Sterling@gmail.com	9 May 2025, 20:57	[illegible]	[illegible]
2	Sterling@gmail.com	9 May 2025, 20:18	[illegible]	[illegible]
3	Sterling@gmail.com	8 May 2025, 20:28	[illegible]	[illegible]

Buy Insurance Policy

Select Insurance Provider:

VIEW RECORD IN PATIENT

Diagnosis and Claim Submission:

- Doctor decrypts the shared file and submits encrypted diagnosis.
- Claim request is made to the insurer with IPFS CID and DID metadata.

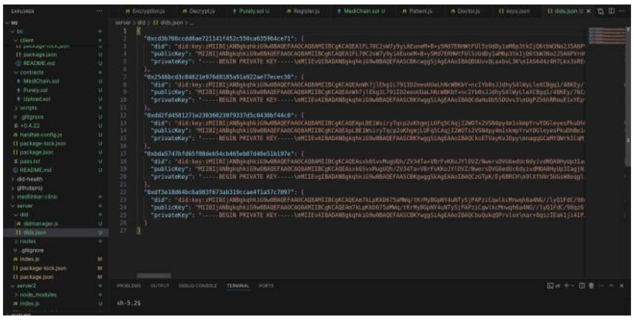
Claim Approval/Rejection:

- Insurer uses doctor’s and patient’s DID to verify authenticity before payment.
- Transactions are recorded on Ethereum Sepolia Testnet.

C. DID Lifecycle

Each DID follows a lifecycle from creation to revocation:

- DID = did:key:z<publicKeyBase58>
- Stored in dids.json on backend server.
- Used to derive public keys for encryption and signing.



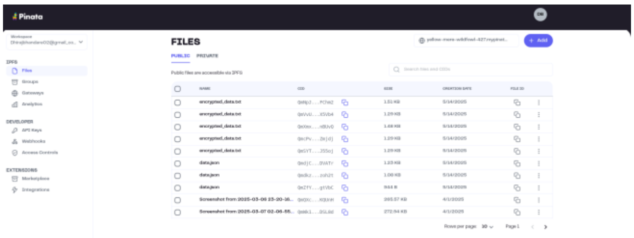
```
{
  "0xcd3b766ccdd6ae721141f452c550ca635964ce71": {
    "did": "did:key:zMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1FL70C2sW7y9yiAEuneM+B+y5Md7ERMWtFUl5zUdDy1aM6p3tkIjQ6tbW3No2J5A6P
    "publicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1FL70C2sW7y9yiAEuneM+B+y5Md7ERMWtFUl5zUdDy1aM6p3tkIjQ6tbW3No2J5A6PYrm
    "privateKey": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDUUvvDLaxbvL3K\nIA56d4z4H7Lkx3sRE
  },
  "0x2546bcd3c84621e976d8185a91a922ae77ecec30": {
    "did": "did:key:zMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnWh7jlEkg1L791ID2eeoUUaLhNcWBKbY+ncIYb0sJJdhyS4lWyLleXCBgq1/48KEy/
    "publicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnWh7jlEkg1L791ID2eeoUUaLhNcWBKbY+ncIYb0sJJdhyS4lWyLleXCBgq1/48KEy/76C
    "privateKey": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCdaMuDU5SDUvv3\nUgP256hRRouE1xYEp
  },
  "0xdd2fd4581271e230360230f9337d5c0430bf44c0": {
    "did": "did:key:zMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApLBEIWsiryTqcp2oKhgmjLUFq5CAqjI2WOTs2VSN4py4mlskmpYrwYDGleyesPkuDh
    "publicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApLBEIWsiryTqcp2oKhgmjLUFq5CAqjI2WOTs2VSN4py4mlskmpYrwYDGleyesPkuDhBe1
    "privateKey": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCksETVayKvJ0py\nnagqGCaMtDWrkICqM
  },
  "0xbda5747bfd65f08deb54cb465eb87d40e51b197e": {
    "did": "did:key:zMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsxk65vxMugUOh/ZV34Ta+VBrFvKKoJYl0VZ/NwersDVG6edUc6dy1vdMQA0HyUp3Ia
    "publicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsxk65vxMugUOh/ZV34Ta+VBrFvKKoJYl0VZ/NwersDVG6edUc6dy1vdMQA0HyUp3IagjN.
    "privateKey": "-----BEGIN PRIVATE KEY-----\nMIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQCzGTpK/Ey6BRCH\n9lXfhNr5UGsW8oqgl.
  },
  "0xdf3e18d64bc6a983f673ab319ccae4f1a57c7097": {
    "did": "did:key:zMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAm7kLpKkD675aMWq/tKrMyBGpNY4uNTy5jPAPziCqwlkcMnwqh6a4NG//lyQ1FdC/98
    "publicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAm7kLpKkD675aMWq/tKrMyBGpNY4uNTy5jPAPziCqwlkcMnwqh6a4NG//lyQ1FdC/98qz6
    "privateKey": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCbuDukqQPrvlox\nar+0qszIEak1j141P.
  }
}
```

```
sh-5.2$
```

DID JSON IMAGE

D. IPFS Configuration

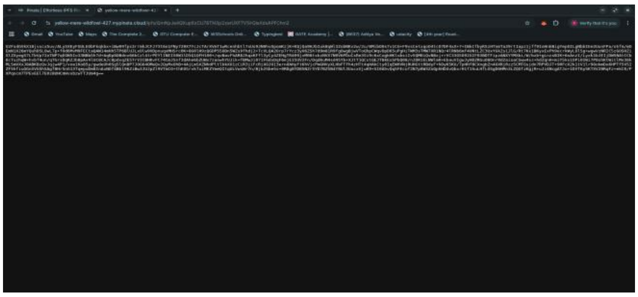
- IPFS daemon used for local development.
- Files encrypted using X25519 before uploading.
- IPFS hash (CID) returned is stored on-chain.



Public files are accessible via IPFS

NAME	CID	SIZE	CREATION DATE
encrypted_data.txt	QmNpJ....FChw2	1.51 KB	5/14/2025
encrypted_data.txt	QmVvU....X5Vb4	1.29 KB	5/14/2025
encrypted_data.txt	Qm9mx....n8Uv0	1.48 KB	5/14/2025
encrypted_data.txt	QmcPV....Zmjdj	1.29 KB	5/14/2025
encrypted_data.txt	QmSYT....J55oj	1.29 KB	5/14/2025
data.json	QmdjC....DVATr	1.23 KB	5/14/2025
data.json	Qmdkz....zoh2t	1.08 KB	5/14/2025
data.json	QmZfY....gtVbC	944 B	5/14/2025
Screenshot from 2025-03-06 23-20-16...	QmQXC....KQUnH	265.57 KB	4/1/2025
Screenshot from 2025-03-07 02-06-55...	Qmkk1....DSL6d	272.94 KB	4/1/2025

Rows per page 10 Page 1


Pinata | Effortless IPFS Fi...
yellow-mere-wildfowl-42...
yellow-mere-wildfowl-427.mypinata.cloud
Gmail
YouTube
Maps
Verify that it's you

```
[illegible]
```

IPFS IMAGE

VII. RESULTS

A. Functional Validation

Each major component was tested in isolated and integrated scenarios:

Functionality	Validation Method
DID Generation	Console + DID JSON storage
Patient Record Encryption & Share	File encrypted + IPFS CID
Doctor Diagnosis	Encrypted file written to IPFS
Claim Submission & Processing	Verified on-chain (Sepolia)
Access Control Grant/Revoke	State change in smart contract

B. Blockchain Transaction Logs

- All Ethereum transactions were successfully committed to Sepolia testnet.
- Transaction hashes can be traced via Etherscan testnet viewer.
- Gas usage remained within expected limits for all operations.

C. IPFS Access and Integrity

Files encrypted on the client using recipient DID. Uploaded to IPFS, ensuring:

- Data immutability
- Redundancy
- Integrity check via hash

D. Security Analysis

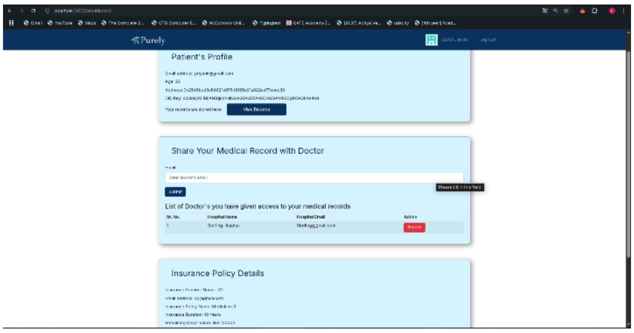
Threat Vector	Mitigation Strategy
Data Interception	End-to-end X25519 encryption
Impersonation/Replay Attacks	Signature validation via Ed25519 DID keys
Centralized Failure Point	IPFS + Ethereum = fully decentralized architecture
Unauthorized Data Access	On-chain access control + encryption

E. Performance Metrics

Metric	Value
Average DID generation time	~50ms (client-side)
Average file upload time	~2.3s (to IPFS)
Ethereum txn finality	~14–16s (Sepolia)

Metric	Value
Claim verification latency	<1s (smart contract call)

F. Screenshots and Visual Output



Patient's Profile

Email address: priyank@gmail.com

Age: 22

Address: [illegible]

DID Key: [illegible]

Your records are stored here: View Records

Share Your Medical Record with Doctor

Email

Enter doctor's email

Submit

Please fill in this field.

List of Doctor's you have given access to your medical records

Sr. No.	Hospital Name	Hospital Email	Action
1	Sterling Hospital	Sterling@gmail.com	Revoke

Insurance Policy Details

Insurance Provider Name: LIC

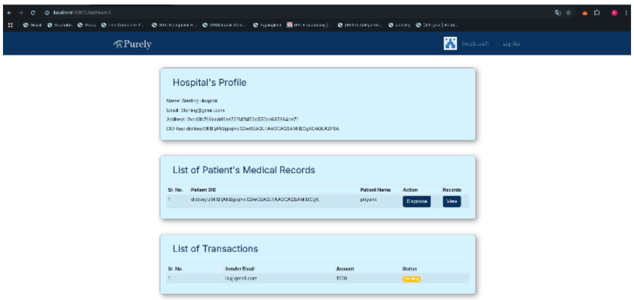
Email address: lic@gmail.com

Insurance Policy Name: Mediclaim 3

Insurance Duration: 10 Years

Remaining cover amount: INR 100000

Patient dashboard



Hospital's Profile

Name: Sterling Hospital

Email: Sterling@gmail.com

Address: [illegible]

DID Key: [illegible]

List of Patient's Medical Records

Sr. No.	Patient DID	Patient Name	Action	Records
1	[illegible]	priyank	Diagnose	View

List of Transactions

Sr. No.	Sender Email	Amount	Status
1	lic@gmail.com	1000	Pending

Doctor Dashboard

VIII. CONCLUSION

This research presents a novel decentralized architecture for managing electronic health records (EHRs) using Decentralized Identifiers (DIDs), verifiable credentials, IPFS, and blockchain smart contracts. The system, titled *Purely*, ensures that patients retain sovereignty over their medical data while enabling secure, privacy-preserving sharing with hospitals and insurance providers. By leveraging asymmetric encryption for end-to-end data protection and Ethereum smart contracts for tamper-proof logging of access and consent, the platform addresses core challenges in interoperability, consent control, and data security in existing centralized healthcare systems.

Experimental results from the Sepolia testnet validate the feasibility of the system in real-time healthcare interactions, including record sharing, diagnosis issuance, and insurance claim processing. The integration of MetaMask, IPFS, and the ethics library demonstrates the practical realization of self-sovereign identity principles. This work sets a foundational framework for future adoption of decentralized health data infrastructure and highlights the potential of DIDs and blockchain in meeting evolving regulatory and ethical standards for digital health systems.

IX. REFFERENCES

[1] A. Khurshid, R. Ayday, L. Lee, J. K. DeMatteo, and K. P. Subbian, “MediLinker: A blockchain-based decentralized health information management platform,” Blockchain in Healthcare Today, vol. 4, 2021.

[2] A. Khurshid, R. Ayday, K. Subbian, “The future of health information exchange (HIE): decentralized patient identity management,” Journal of the American Medical Informatics Association, vol. 28, no. 3, pp. 613–617, 2021.

[3] M. A. Bouras, N. Hafid, and M. Samhani, “A privacy-preserving blockchain-based system for personal data sharing,” Journal of Network and Computer Applications, vol. 178, 2021.

[4] Decentralized Identifiers (DIDs) v1.0, W3C Recommendation, Jul. 2022. [Online]. Available: https://www.w3.org/TR/did-core/

[5] C. Allen, “The Path to Self-Sovereign Identity,” [Online]. Available: https://www.coindesk.com/markets/2016/04/27/the-path-to-self-sovereign-identity/

[6] J. Bernabe, D. Hernandez-Ramos, J. L. Canovas, and A. Skarmeta, “Privacy-Preserving Solutions for Blockchain: Review and Challenges,” IEEE Access, vol. 7, pp. 164908–164940, 2019.

[7] M. Zhang, X. Zhu, H. Wang, and H. Yu, “A Blockchain-Based Access Control Framework for Secure Cloud Data Storage,” IEEE Access, vol. 8, pp. 181721–181731, 2020.

[8] Ethereum Foundation, “Ethereum Sepolia Testnet,” [Online]. Available: https://sepolia.etherscan.io/

[9] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” arXiv preprint arXiv:1407.3561, 2014.

[10] L. Zhang, Z. Yu, C. Wang, and M. M. Hassan, “Towards a Decentralized and Privacy-Preserving Healthcare Framework using Blockchain,” IEEE Access, vol. 8, pp. 183101–183112, 2020.