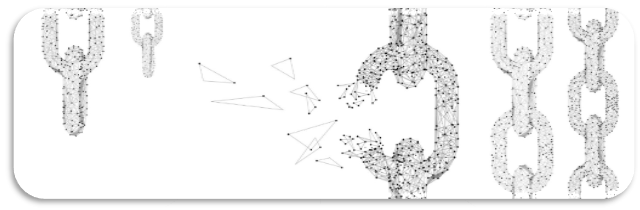
Blockchain in PHR

Blockchain is a decentralized, distributed ledger technology that records transactions in a secure and transparent manner. It was originally introduced as the underlying technology for Bitcoin, a peer-to-peer electronic cash system, but its potential applications have expanded to various industries, including healthcare. The key features of Blockchain include decentralization, immutability, security, and transparency.



Ethereum

Ethereum is one of the most widely used blockchain platforms with a robust smart contract ecosystem. You can create a custom Ethereum-based token to represent patient data or use Ethereum for identity management. However, Ethereum can have scalability and cost issues due to its Proof of Work (PoW) consensus mechanism.

Why Ethereum?

1. Smart Contracts: Ethereum is known for its robust smart contract functionality, which can be used to create self-executing agreements and automate processes in our PHR app.
2. Developer Community: Ethereum has a large and active developer community, which means we can find a wealth of resources, tools, and support when building our app.
3. Security: Ethereum has a track record of security and reliability, making it a suitable choice for handling sensitive health data.
4. Interoperability: Ethereum's widespread adoption means that our app can potentially interact with other Ethereum-based apps and services, enhancing its ecosystem.
5. Tokenization: We can create custom Ethereum-based tokens to represent health data or access rights, enabling secure and transparent transactions within our app.

Details the Blocks will possibly contain:   
In a Personal Health Record (PHR) application built on a blockchain, each block typically contains a collection of data related to healthcare records and transactions. The specific information contained in each block may vary depending on the design and requirements of PHR application, but here are some common elements we will have:

1. Patient Identifiers:

* Patient ID or a hashed representation of it.
* Patient's public key for access control and encryption.

1. Healthcare Records:

* Medical records, including diagnoses, treatments, medications, and test results.
* Timestamps for each record to track when it was added or modified.

1. Smart Contract Transactions:

* Transactions related to access control and consent management.
* Details of who has access to the patient's records and under what conditions.

1. Hashes and Pointers:

* Hashes of previous blocks (to maintain the blockchain's integrity).
* Hash of the data within the current block.

1. Metadata:

* Metadata about the healthcare records, such as the source (healthcare provider, laboratory, etc.), data format, and data provenance.

1. Digital Signatures:

* Digital signatures from healthcare providers, patients, or other authorized parties to verify the authenticity of records and transactions.

1. Access Logs:

* Logs of who accessed or modified the patient's records and when.
* Access control policies or smart contract conditions for granting access.

1. Consent Records:

* Records of patient consent for sharing data with specific individuals or organizations.

1. Encrypted Data:

* Encrypted versions of sensitive data to ensure privacy and confidentiality.
* Encryption keys or references for authorized parties to decrypt data.

1. Blockchain-Specific Information:

* Block number and timestamp.
* Transaction ID or hash.
* Reference to the previous block.

A diagram of blockchain and a blockchain diagram

Description automatically generated

*Fig. 1: Block entry in Blockchain*

A diagram of a block diagram

Description automatically generated

*Fig. 2: Representation of blockchain structure*

Database in Blockchain

Firebase

Firebase can be a valuable addition to a Personal Health Record (PHR) app, providing a range of services to enhance the functionality and user experience. Here's why we will use Firebase in our PHR app:

1. Authentication: Firebase Authentication can be used to implement secure user registration and login. Users can create accounts using email/password, social media logins, or even phone number verification. This is essential for securing access to patient health records.
2. Realtime Database: Firebase Realtime Database is a NoSQL cloud database that can store and sync data in real-time across multiple clients. It can be used to store and retrieve patient health records, appointment information, and other dynamic data. Realtime updates ensure that patients and healthcare providers always have the latest information.
3. Push Notifications: Firebase Cloud Messaging (FCM) can be used to send push notifications to users, such as appointment reminders or important updates regarding their health records.
4. Scalability and Reliability: Firebase is hosted on Google Cloud, providing scalability and reliability. It can handle sudden increases in user traffic and maintain high availability.

A diagram of a login process

Description automatically generated

SQLite

SQLite is a popular choice for implementing databases in Personal Health Record (PHR) applications on the Android platform.

Why SQLite?

1. Lightweight and Embedded: SQLite is a serverless and self-contained database engine that runs directly on the user's device. It's lightweight and doesn't require a separate database server, making it an excellent choice for mobile apps with limited system resources.
2. Default Android Database: SQLite is the default and recommended database system for Android. The Android operating system provides built-in support for SQLite, ensuring compatibility and easy integration with Android app development.
3. Transaction Support: SQLite provides transaction support, allowing developers to ensure data consistency and integrity. This is crucial for healthcare applications like PHR, where data accuracy is essential.
4. Security: While SQLite itself doesn't provide built-in encryption, it allows developers to implement encryption for sensitive data, such as healthcare records. Libraries like SQLCipher can be integrated with SQLite to add encryption capabilities.
5. Offline Access: SQLite will enable our PHR app to provide offline access to patient data. Users can view and edit their records even when they don't have an internet connection, and changes can be synchronized later.