Ampere Blockchain Model Whitepaper

Ryen Krusinga

September 21, 2023

Contents

1 Introduction

We describe a blockchain-based medical app capable of combining patient data from different institutions and formats into a coherent, decentralized whole. Our format allow for natural AI-assisted verification of data consistency and provides a secure, institution-independent way of representing personal medical information.

2 Components Overview

The app consists of several interconnected services running on cloud servers, whitelisted institutional computer networks, and whitelisted private devices. These components include:

- **Blockchain core software:** A code library that allows authorized app instances to read from and/or write to a patient's blockchain.
- **Data/AI layer software:** Code that formats patient data for use on the blockchain and checks data consistency with assistance from AI.
- **Patient database software:** Code that fingerprints patients and maps them to their corresponding blockchain.
- **FHIR/EHR bridging software:** A service that interfaces with EHR software from different institutions in order to obtain patient data and verify it.
- **Cloud service bridging software:** Software components that interface with cloud computing providers such as Microsoft Azure and Amazon Web Services. These components abstract away cloud services so that core app components can run on different providers. These cloud services include:
  - Serverless coding environments like Azure Functions and AWS Lambda
  - Scalable, replicated databases
  - API calls to AI models and other verification code.
- **Key and version management software:** A service that manages whitelisted app instances, distributes and revokes cryptographic keys, and distributes code updates to nodes running the app.
- **Frontend interface software:** A user-friendly desktop app for interacting with the blockchain service.
- **Loggin, error reporting, and analytics software:** In order to constantly improve the app, we will need to extract usage analytics, data logs, and error reports from nodes implementing our technology.

In the future, as the app grows in complexity, other services may be needed, such as a load-balancing service, a shard management service like Kubernetes, and so forth.

3 Blockchain Design

Our blockchain model consists of a header, a subheader, and a data frame. The data frame has a fractal structure, in the sense that each data frame can contain other data frames, containing other data frames, and so on, each frame at every level implementing its own hash function, linearization function, constructor, and so forth. Not much nesting is likely to be needed, but this makes the model quite extensible. All frames should begin with their type and version number, as this determines which code should be used to handle the contents.

Elements of the design shall include:

- Block headers (not included in the overall block hash) consisting of:
  - The overall block hash
  - Sufficient digital signatures of the hash from the providing node and all verifying nodes
  - Other metadata as needed
- Block subheaders (included in the overall hash) consisting of:
  - Block format version number
  - Block size
  - Timestamp the block was added
  - Block manifest: a list of data sub-blocks contained inside, including their type and size.
  - The hash of the previous block in the central chain
  - A list of hashes of other blocks and code entities in (something like) prefix notation, describing the computation (if any) that was done to generate the data in the current block (e.g., calling a specific AI model).
  - Other metadata as needed

- The data section, consisting of concatenated data blocks containing:
  - The data block type
  - The version number of the given type
  - The timestamp (or timestamps) of when the data was physically collected, or first recorded (this could be very different than the timestamp when the block was added to the chain).
  - The size of the data segment
  - The actual data

The entire blockchain can be stored in a binary file consisting of a metadata segment followed by a list of all the blocks in sequence. For any given patient, this file ought not be much longer than a few dozen kilobytes, or perhaps a megabyte or so at most, depending on the type of data inside.

Each patient, doctor, institution, and other app-related entity will have its own blockchain. Since these blockchains are privately maintained by whitelisted provider nodes, hyper-efficiency of the blockchain protocol is not paramount. This gives us room to allow the block protocol to have the complex structure (as described above) necessary to implement the desired functionality.

3.1 Block Algorithm, Data Validation Protocol, and AI Integration

The block algorithm shall be approximately as follows:

1. **Patient encounter:** A whitelisted institution sees a patient. They make an EHR entry.
2. **Consent:** Through the FHIR system or an EHR-specific extension, the blockchain software sees that a patient has a new entry. If the patient has given consent to use the new technology, then the software can proceed to add it to the blockchain.
3. **Intake:** A data entry block is created, linked to the latest block in the central chain. It stores the EHR note as-is to the blockchain, with no modification yet. It can be in any format.

4. **Further Processing:** A second, processed block is created and added to the blockchain after the unprocessed intake block. This processed block reformats the data to our own specification, possibly using the assistance of AI.

5. **Consistency checking:** The information in the processed block is checked against the history of the blockchain, possibly with the help of AI, in order to determine if there are any errors. This step is the most complex. Possibly more blocks are generated as a result, each depending on the blocks computed before.

6. **Updating the central chain:** A new central summary node is created that computes an up-to-date overall patient status based on the new information.

3.2 Philosophy Behind the Design

The flexible data formats, versioning system, and multi-block intake procedure all combine to make this blockchain maximally flexible in its ability to represent patients. It can accomodate patient information of any type in essentially any format. New formats and new data types can be added in the blockchain's future, along with better and better versions of the verification protocol. Furthermore, by storing intermediate computations as their own blocks, and by using the prefix-notation hash list to describe how each block was produced, the blockchain achieves full transparency of how it obtained, reformatted, and modified its data.

4 Security Design and Key Distribution Model

The goals of our security model include:

- HIPAA compliance: patient blockchain data should only exist in an unencrypted state on HIPAA-compliant servers.

- Whitelisting: only designated institutions should be able to read and write to the blockchain.

- Blockchain validation: new data added to the blockchain must be cryptographically signed by the institution that provided the data. Insofar

as possible, new blocks must be validated and signed off on by a selection of other trusted blockchain nodes.

- Patient control over data: there should be a secure means for patients to read the contents of their blockchain.
- Robustness to attack: it should be exceedingly difficult or impossible for attackers to make unauthorized modifications to the blockchain or to read protected patient information.
- Key revokability: there should be a decentralized protocol in place to revoke any previously whitelisted keys that may have been compromised.

To achieve these ends, several pieces of core functionality may be implemented:

- **Total blockchain encryption:** A patient's blockchain information shall be stored in a totally encrypted state using standard symmetric encryption such as AES. The encryption keys themselves shall be stored in a separate encrypted locker file (see multi-key authentication below).
- **Block validation procedures:** Before signing a new proposed block, whitelisted nodes should independently verify the integrity of the block, insofar as is possible. When blocks are partially AI-generated, this becomes tricky. New AI-handling protocols will need to be developed, such as, for example, cryptographic "receipts" from the providers of the AI model in question.
- **Multi-factor and multi-key authentication:** In addition to using standard multifactor authentication techniques, such as mobile texts or emailed codes, patient data can be further protected by "multi-key" authentication. By making the encryption keys to a locker file (see above) depend on the hash of multiple concatenated keys held by boths patients and providers, patients may use the web portals of trusted institutions to view their blockchain info. While all decryption should be done remotely on trusted, HIPAA-compliant servers, this model nevertheless allows patients, if they wish, to download their encrypted personal blockchain and then view it through their desired web portal by uploading it and entering their personal key. The advantages of this

are: (1) if the patient's personal computer and personal key are both compromised by an identity thief, the data still cannot be accessed unless the thief also has access to the patient's web portal account, including all the means of multifactor authentication necessary to login to the portal; (2) whitelisted medical institutions can still read and write to the patient's blockchain without knowing the patient's personal key via multiple locker files.

- **Everything on the blockchain:** There should be blockchains not just for patients, but also for doctors, institutions, code versions, and all other entities related to the app. The collection of whitelisted public keys should itself exist on the blockchain in a decentralized way, allowing modifications to this list to also be made in a decentralized manner. (This is similar to PGP. However, it is unclear at this time what role, if any, a centralized set of whitelisted keys belonging to our organization should play in the blockchain. Our company is the authority that allows institutions to use our blockchain app technology, after all, so perhaps instead we should retain the sole ability to add or revoke new nodes. This is a matter for further research.) There should also probably be a global blockchain for mapping patient identities to their corresponding blockchain file.

5 Data/AI Layer Design

6 Patient Database Design

7 Cloud and FHIR/EHR Bridging