mit media lab · viral communications · Robert Wood Johnson Foundation

MedRec

1. Overview

What is Medrec?

Electronic Health Records (EHRs) were never designed to manage the complexities of multi-institutional, lifetime medical records. As patients move between providers, their data becomes scattered across different organizations, losing easy access to past records. As providers — not patients — are the primary stewards of EHRs, patients face significant hurdles in viewing their reports, correcting erroneous data, and distributing the information. The situation is much like consumer finance, where an individual may have several bank accounts, credit cards, loans, and assets but no unified way to access and control them. In the case of finance, however, there is an infrastructure in place that greases the wheels: currency. With medical information we are still in the age of barter.

MedRec is the combination of a social need with a technological enabler: a system that prioritizes patient agency, giving a transparent and accessible view of medical history. To continue the banking analogy, financial systems may contain multiple different depositories of currency, perhaps one for each provider network. The problem is that health records are not fungible; each is an individual's unique imprint. There is no like-for-like trade possible as we can do with money. Whereas competition and multiplicity often results in lower consumer costs, here it risks a mass of incompatible or inaccessible barriers to interchange and control. We propose an alternative: a distributed access and validation system using the blockchain to replace centralized intermediaries.

What are the technologies we use?

Developed as a means of recording financial transactions for digital currencies, the blockchain may be extended more generally as a platform for distributed computation. the blockchain uses public key cryptography to create a tamper-proof chain of content, supported not by a central server but a decentralised network of participating 'nodes'. Conventionally, each individual node works to solve a series of hashing exercises, which contribute to the formation of the chain — a process known as ‘mining’.   However, there are variations that are more efficient, as will be described later.

The first implementation of Medrec was developed using the Ethereum blockchain. Ethereum uses a system called ‘smart contracts’ to execute scripts on the blockchain, extending bitcoin to a turing-complete language. This system of smart contracts significantly broadens the possibilities for decentralised computation using the blockchain, and it is this system which formed the backbone for the development of Medrec.

Does MedRec store patient data?

Smart contracts act as an intelligent representation that links patients and providers to the addresses of existing medical records. Medrec does not ‘store’ the record directly; rather encodes metadata that allows records to be accessed securely by patients, unifying access to data across disparate providers. The metadata contains information about ownership, permission and the integrity of the data being requested. The full details of the smart contract structure and operation can be found in the technical documentation.

Is MedRec economically viable?

The first implementation of Medrec used the value of data to incentivise ‘mining’ of the Medrec blockchain, which is required for the system to scale and the chain to advance. The mining process — a computationally intensive hashing exercise — would be performed by medical researchers, who in turn would gain access to aggregate and anonymised data that might be used to further medical research.

In the current design, a proposal is to encode this exchange as a ‘bounty query’: any transaction that updates a patient provider record attaches a particular query (e.g. a request to return the average iron levels in blood tests done by the provider, across all patients in the previous week). When the block is mined, the block’s miner is appended as the author of nether query, allowing them to collect the data as part go the transaction. This enabling of population-level insights into healthcare outcomes, thus providing a positive benefit to medical research.

This proposal, however, raises concerns with data security and patient privacy. An alternate is that the providers themselves maintain the blockchain. The argument for this is that a closed group can form a consensus in an efficient manner. The overhead for doing so is minimal, and this approach allows the system to grow incrementally rather than require a priori agreement among providers and miners.

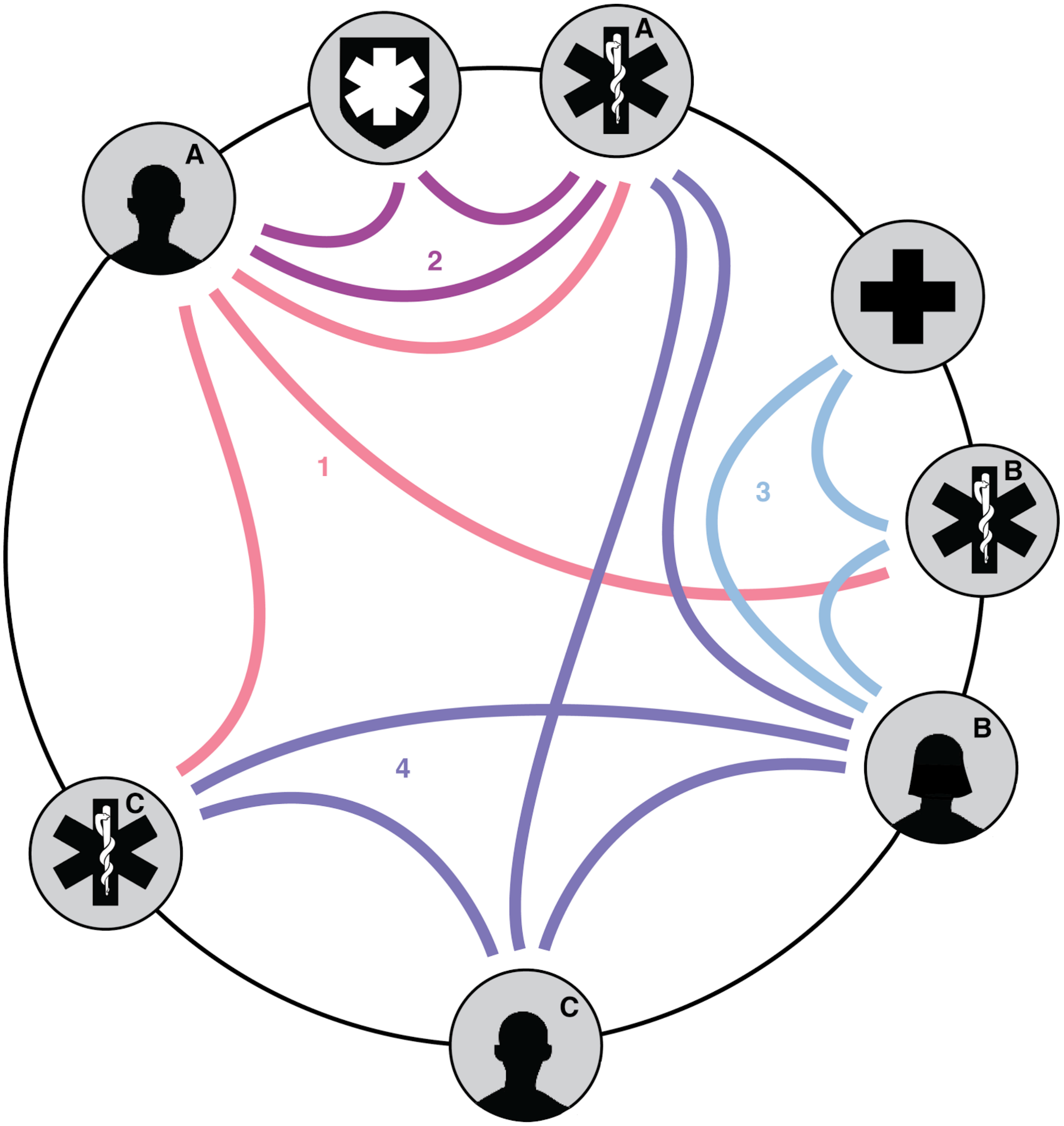
Do we welcome external developers?

Currently, the group are working on bridging from a previous iteration of the Ethereum blockchain to a more up-to-date chain, considering revisions to the architecture as we do so. The overarching aim of

Medrec — to provide secure, transparent and scalable access to medical records — remains the same, and we will open the source code once we have developed a working prototype of Medrec 2.0. More information about the implementation of this version can be found in the technical documentation.

Who supports this project?

Funding from this project comes from the Robert W Johnson foundation and the MIT Media Lab.



2. Technical Documentation

This section of the site lists the proposed architecture of MedRec 2.0 in detail, along with a background that covers the original implementation, and an overview of technology underpinning the decisions we've made. This is intended to be a working document, and will grow as the new codebase develops. MedRec 2.0 is currently under development, but we hope to have an open-source version up and running very soon (to be notified of this, sign up here).

MedRec 1.0

MedRec's first implementation, designed by Ariel Ekblaw and Asaph Azaria is detailed in the white paper "A Case Study for Blockchain in Healthcare". This was the implementation piloted in August 2016 At the Beth Israel Deaconess Medical Centre. This implementation forms an architectural basis for MedRec 2.0, but with significant changes. In particular, MedRec 2.0 is developed using Go-ethereum (Geth) and Solidity — as opposed to the Pyethereum and Serpent libraries in which the original was developed and changes are made to the amount of information stored on the blockchain, with a view to improve both the scaling and privacy properties of the transaction.

MedRec 1.0 was not built on the live Ethereum network, instead creating a small- scale private blockchain with extensive, specific APIs. In order to maintain the flexibility of this platform, modifications (such as wait time before accepting new blocks and the 'difficulty' of the Proof-of-Work consensus mechanism) were made to a number of otherwise rigid blockchain parameters. Since the original chain was forked from Ethereum, the official client has changed to a point where the MedRec app could not be returned to the public blockchain (as was originally intended), and the code has become obsolete. A discussion of public, private and consortium models for blockchain management is explored later on.

The Ethereum Blockchain

MedRec is built on the Ethereum blockchain, released by founder by Vitalik Buterin in 2015. Though not the first to propose them, Ethereum is associated with the use of 'smart contracts' — scripts that enable the exchange of value without deference to a middleman -- which allow more complex transactions to be executed on the blockchain. This significantly broadens the scope of the Bitcoin blockchain from a purely transactional set of operations to a Turing-complete language that supports diverse applications. Ethereum consists of a system of nodes (personal computers, clusters, virtual machines) in a decentralised network which.. Smart contracts are not a replacement for contracts in the traditional sense, rather they are agreements to execute some action or code given a set of conditions. In the case of MedRec, these contracts can be used to encode a set of pointers to the locations of medical data.

Smart Contract Structure

Medrec does not store EHRs directly on the Ethereum blockchain, instead using a relational set of smart contracts to encode pointers, which may be used to locate and authenticate to record locations. MedRec defines three main kinds of smart contract, within which there is some variation between contracts belonging to patients, providers and other forms of user.

Registrar Contract

The registrar contract maps participant IDs (patients, providers, insurers) to their Ethereum address identity (equivalent to a public key). The regulation of new identities can be encoded into the contract, ensuring that only certified institutions may add new information onto the blockchain. In turn, new information concerning a patient (e.g., a new relationship) is only added with the approval of that patient. Each identity string is located at an address on the blockchain, where it is referenced by a Summary Contract. This raises some questions about the suitability of a global system of IDs, which is addressed later on in a discussion of privacy.

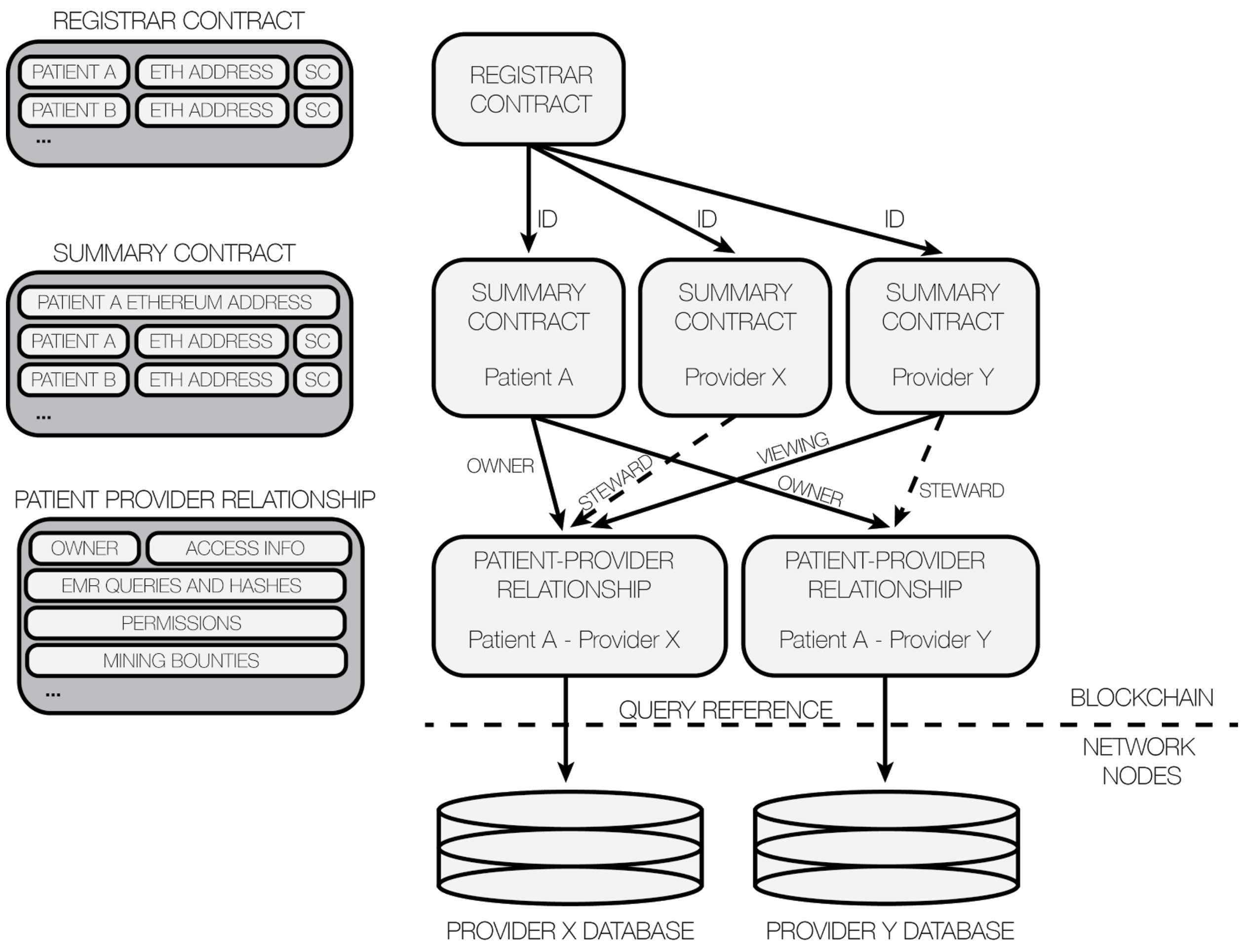
Patient-Provider Relationship Contract

The patient provider relationship contract links two nodes in the system, where one node stores and manages medical records for the other. This relationship could exist between a particular care provider and patient, but extends to cover any pairwise data stewardship interaction.

Summary Contract

The summary contract serves as a trail of bread-crumbs, where each participant in the system can locate a summary of their relationships with each other participant. The summary contract encodes a list of references to Patient-Provider Relationship contracts, giving both current and previous engagements with other nodes on the system. Each relationship also stores a ‘status’ variable, indicating when the relationship was established, and whether it has been approved by the patient. The acceptation, rejection or deletion of relationships is controlled by the patient, giving full control over which records in their history they wish to acknowledge. This feature of MedRec is key to satisfying its usability criterion: pointer to fragmented records are drawn together in a single, dedicated location.

A diagram showing the possible relationships between different patient and provider contracts is shown below. Note that the ‘status’ variable of a particular contract can hold different values depending on the permissions it allows (e.g., the status variable for the Patient A -> Provider X PPR encodes permissions for Provider Y to view, but the Patient A -> Provider Y PPR does not encode those same permissions for Provider X). Note also that the contracts are only used to provide pointers: the database queries that return the records themselves are handled off- chain.



Data Handling

When a patient requests access to a particular medical record, it sends a request to the provider's Database Gatekeeper, part of the off-chain infrastructure of MedRec. The Database Gatekeeper implements an access interface to the patient node's local database, governed by permissions stored on the blockchain. The Gatekeeper runs a server listening to query requests, which are cryptographically signed by the issuer, from clients on the network. The cryptographic signature allows the gatekeeper to confirm identities, and then check the blockchain contracts to verify if the address issuing the request is allowed access to the query. If the address is valid, it runs the query on the node's local database and returns the result over to the client.

Here, we assume that many ‘provider’ nodes, especially those associated with care providers, already store data on networked servers with a high degree of security. MedRec also defines a modular interoperability protocol, which can interface with any provider backend and UI implementation. Patient nodes also contain a local database, though more lightweight, which functions as a cache storage of the patient’s data. The patient node implementation is a ‘light’ node, which could be executed on a PC or mobile phone.

Public and Private Blockchains

MedRec does not seek to build directly on the public Ethereum blockchain, instead constructing a private peer-to-peer network of trusted, verified nodes. It is this private client that defines a Global Registrar (a contract mapping all public identities to Ethereum addresses), ensuring that only registered healthcare providers are permitted to append blocks to the MedRec blockchain. Though this use of a centralised arbiter of a ‘provider whitelist’ to determine the validity of each potential sealer contradicts the decentralised spirit of blockchain technology, we believe that this level of centralised trust in healthcare providers is not unreasonable, as patients already place a high degree of trust in their healthcare providers.

If a private blockchain is used, we may also use a different consensus mechanism to the main Ethereum blockchain. Here we only allow verified sealers to vote on appended blocks, rather than the Proof-of-Work algorithm, which requires the computation of hashes to contribute to the maintenance of a distributed ledger. Mining mechanisms to sustain the distributed ledger would still be required: this could be a requirement of provider nodes, who could use spare computing power (this would be relatively less intensive than proof-of-work). Another suggestion, made by MedRec 1.0 would be to incentivise mining from providers and academic institutions, who would in turn receive anonymised, aggregate medical data that could be used in data analysis.

Privacy

Blockchain technologies introduce a number of privacy limitations, some of which are mitigated by the use of a permissioned read access structure and private blockchain. A key issue is that, even without the direct disclosure of a patient name, inference about who a particular patient is could be drawn from metadata of one ethereum address with multiple others. Even with a private blockchain, we must consider mining nodes, who process this sensitive metadata. One solution is to require all miners to be fully-permissioned, and require medical researchers running mining nodes only to do so on secured systems. One future goal of the project is to perform a full k-anonymity analysis of the query structure to ensure that privacy is preserved. In addition, MedRec proposes the addition of encryption in the off-blockchain synchronisation steps, safeguarding against accidental or malicious content access.

A proposed solution to the problem of privacy is to use a system of ‘delegated contracts’, where each provider creates a separate Ethereum identity for each new Patient Provider Relationship. This means that, instead of a single provider address, from which relationships with particular patients could easily be derived, the provider’s identity is distributed across the network. In order for the relationship to be created securely, however, the provider could not append the fresh block containing this new address itself (as it would be simple to trace back each of these delegate addresses to the original). Thus, on creating a new delegate account, the provider performs an off-blockchain transaction with another verified provider at random, giving them the details of the new delegate account, which they may append as a verified account to the blockchain.

Security

The decentralised nature of blockchain-based systems gives MedRec the advantage of robustness both in the maintenance of authorisation data (which is maintained in every node in the network), and in the storage of records themselves (which are persisted in both patient and relevant provider nodes). As many entities participate in the system, consensus mechanisms also mean that single points of failure are avoided. As the medical data and the global authorisation log are distributed, there is no central target for attack or failure, and the network is robust to tampering (as an altered node would be in conflict with the other, unaltered nodes and would not pass a consensus).

MedRec does not seek to address the security concerns at the level of provider databases (which must be properly managed by the provider’s IT admin), and neither is end-point security addressed (a compromised patient computer could potentially allow data to be stolen). MedRec also assumes that nodes are bound by external regulation governing data copying, and does not seek to address this in its own right.

MedRec 1.0 never underwent a full security review (although measures were taken to avoid obvious attacks such as SQL injection), and in implementing MedRec 2.0 we will need to address the potential for vulnerability in the smart contracts, particularly in the wake of the DAO hack.

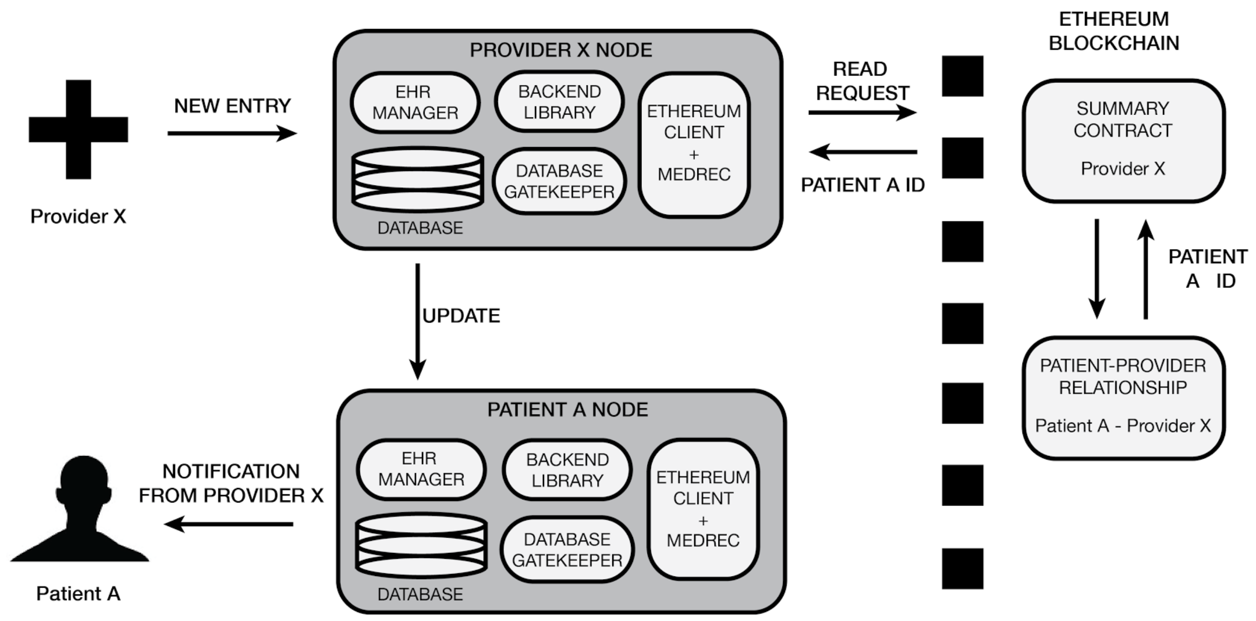
Scalability

Scalability is an ongoing concern in the Ethereum community and one that has yet to be fully resolved. One of the key issues is that any event stored at any time on the blockchain will appear in every subsequent block. Though this is also a property of the Bitcoin blockchain, as Ethereum provides the potential for both data storage and more complex operations, the repercussions of this growth are more of an issue.

One advantage of a private blockchain is that the app only needs to track back to the genesis of the private chain, not Ethereum as a whole. However, MedRec must still address its own scaling. A key architectural modification between MedRec 1.0 and 2.0 is to bypass the blockchain for patient notifications. This prevents an ‘event’ needing to be recorded on the blockchain every time a medical record undergoes a change of state (which could happen up to several times a day, per PPR, for a patient under constant medical attention), and instead restricts blockchain storage to the creation and modification of identities and relationships, rather than the metadata surrounding them.

This change also means that the pseudonymity issue associated with communication between patients and providers is somewhat mitigated. Looking at the blockchain, an external observer might view a relationship between patient A and provider X, but would not be able to determine the frequency by which the 2 communicated, as that interaction.

The alternative approach (changed between MedRec 1.0 and 2.0) is detailed in the diagram below:


Provider X
NEW ENTRY
PROVIDER X NODE
EHR MANAGER
BACKEND LIBRARY
ETHEREUM CLIENT + MEDREC
DATABASE GATEKEEPER
DATABASE
READ REQUEST
PATIENT A ID
ETHEREUM BLOCKCHAIN
SUMMARY CONTRACT
Provider X
PATIENT A ID
PATIENT-PROVIDER RELATIONSHIP
Patient A - Provider X
UPDATE
PATIENT A NODE
EHR MANAGER
BACKEND LIBRARY
ETHEREUM CLIENT + MEDREC
DATABASE GATEKEEPER
DATABASE
NOTIFICATION FROM PROVIDER X
Patient A