MedRec: Using Blockchain for Medical Data Access and Permission Management

Asaph Azaria, Ariel Ekblaw, Thiago Vieira and Andrew Lippman
Media Lab
Massachusetts Institute of Technology
Cambridge, MA, 02139, USA
Email: {azaria, aekblaw, tvieira, lip}@mit.edu

***Abstract*—Years of heavy regulation and bureaucratic inefficiency have slowed innovation for electronic medical records (EMRs). We now face a critical need for such innovation, as personalization and data science prompt patients to engage in the details of their healthcare and restore agency over their medical data. In this paper, we propose MedRec: a novel, decentralized record management system to handle EMRs, using blockchain technology. Our system gives patients a comprehensive, immutable log and easy access to their medical information across providers and treatment sites. Leveraging unique blockchain properties, MedRec manages authentication, confidentiality, accountability and data sharing– crucial considerations when handling sensitive information. A modular design integrates with providers' existing, local data storage solutions, facilitating interoperability and making our system convenient and adaptable. We incentivize medical stakeholders (researchers, public health authorities, etc.) to participate in the network as blockchain "miners". This provides them with access to aggregate, anonymized data as mining rewards, in return for sustaining and securing the network via Proof of Work. MedRec thus enables the emergence of data economics, supplying big data to empower researchers while engaging patients and providers in the choice to release metadata. The purpose of this short paper is to expose, prior to field tests, a working prototype through which we analyze and discuss our approach.**

***Keywords*-electronic medical records; cryptographic protocols; access control; distributed information systems**

I. Introduction

Medical records crave innovation. Patients leave data scattered across various jurisdictions as life events take them away from one provider's data silo and into another. In doing so they lose easy access to past data, as the provider, not the patient, generally retains primary stewardship (either through explicit legal means in over 21 states, or through default arrangements in the process of providing care) [1]. Through the HIPAA Privacy Rule, providers can take up to 60 days to respond (not necessarily to comply) to a request for updating or removing a record that was erroneously added [2]. Beyond the time delay, record maintenance can prove quite challenging to initiate as patients are rarely encouraged and seldom enabled to review their full record [1], [2]. Patients thus interact with records in a broken manner that reflects the nature of how these records are managed.

Interoperability challenges between different provider and hospital systems pose additional barriers to effective data sharing. This lack of coordinated data management and exchange means health records are fragmented, rather than cohesive [3]. Patients and providers may face significant hurdles in initiating data retrieval and sharing due to economic incentives that encourage "health information blocking". A recent ONC report details several examples on this topic, namely health IT developers interfering with the flow of data by charging exorbitant prices for data exchange interfaces [4].

When designing new systems to overcome these barriers, we must prioritize patient agency. Patients benefit from a holistic, transparent picture of their medical history [3]. This proves crucial in establishing trust and continued participation in the medical system, as patients that doubt the confidentiality of their records may abstain from full, honest disclosures or even avoid treatment. In the age of online banking and social media, patients are increasingly willing, able and desirous of managing their data on the web and on the go [3]. However, proposed systems must also recognize that not all provider records can or should be made available to patients (i.e. provider psychotherapy notes, or physician intellectual property), and should remain flexible regarding such record-onboarding exceptions [5], [6].

Medical records also prove critical for research. The ONC's report emphasizes that biomedical and public health researchers "require the ability to analyze information from many sources in order to identify public health risks, develop new treatments and cures, and enable precision medicine" [4]. Though some data trickles through to researchers from clinical studies, surveys and teaching hospitals, we note a growing interest among patients, care providers and regulatory bodies to responsibly share more data, and thus enable better care for others [7], [4].

In this work, we explore a blockchain structure applied to EMRs. We build on this distributed ledger protocol originally associated with Bitcoin [8]. The blockchain uses public key cryptography to create an append-only, immutable, timestamped chain of content. Copies of the blockchain are distributed on each participating "node" in the network. The Proof of Work algorithm used to secure the content from tampering depends on a "trustless" model, where individual nodes must compete to solve computationally-intensive "puzzles" (hashing exercises) before the next block

978-1-5090-4054-4/16 $31.00 © 2016 IEEE
DOI 10.1109/OBD.2016.11

of content can be appended to the chain. These worker nodes are known as "miners", and the work required of miners to append blocks ensures that it is difficult to rewrite history on the blockchain.

Our MedRec blockchain implementation addresses the four major issues highlighted above: fragmented, slow access to medical data; system interoperability; patient agency; improved data quality and quantity for medical research. We assemble references to disparate medical data and encode these onto a blockchain ledger. We organize these references to explicitly create an accessible bread crumb trail for medical history. Our system supplements these pointers with on-chain permissioning and data integrity logic, empowering individuals with record authenticity, auditability and data sharing. We build robust, modular APIs to integrate with existing provider databases for interoperability. A novel data-mining scheme is proposed to sustain the MedRec network and bring open, big data to medical researchers. We present MedRec not as the panacea for medical record management, but as a foray into this space to demonstrate innovative EMR solutions with blockchain technology.

II. Prior Art

Recent work by Zyskind et al. has demonstrated the use of blockchain protocols for permission management. They implement a trusted blind escrow service, storing encrypted data while logging pointers on the blockchain [9]. Kish proposed the blockchain for hypothetical key management in a medical context [7]. We build on these ideas and develop original work in distributed record retrieval, smart contract permissioning schemes, data sharing, and the economics of information supply and demand via blockchain mining.

We know of two efforts nominally involved in medical records on the blockchain, notably Factom [10] and MedVault [11]. Neither have yet to publish specific methods or a summary of technical work. To the best of our knowledge, we are the first to introduce a fully functional prototype, applying blockchain technology to medical records.

III. System Implementation

A. Overview

For MedRec, the block content represents data ownership and viewership permissions shared by members of a private, peer-to-peer network. Blockchain technology supports the use of "smart contracts", which allow us to automate and track certain state transitions (such as a change in viewership rights, or the birth of a new record in the system). Via smart contracts on an Ethereum blockchain [12], we log patient-provider relationships that associate a medical record with viewing permissions and data retrieval instructions (essentially data pointers) for execution on external databases. We include on the blockchain a cryptographic hash of the record to ensure against tampering, thus guaranteeing data integrity. Providers can add a new record associated with a particular patient, and patients can authorize sharing of records between providers. In both cases, the party receiving new information receives an automated notification and can verify the proposed record before accepting or rejecting the data. This keeps participants informed and engaged in the evolution of their records.

MedRec prioritizes usability by also offering a designated contract which aggregates references to all of a user's patient-provider relationships, thus providing a single point of reference to check for any updates to medical history. We handle identity confirmation via public key cryptography and employ a DNS-like implementation that maps an already existing and widely accepted form of ID (e.g. name, or social security number) to the person's Ethereum address. A syncing algorithm handles data exchange "off-chain" between a patient database and a provider database, after referencing the blockchain to confirm permissions via our database authentication server.

In the following sections we present the design principles of our distributed system and its implementation.

B. Blockchain

Originally designed for keeping a financial ledger, the blockchain paradigm can be extended to provide a generalized framework for implementing decentralized compute resources [12]. Each compute resource can be thought of as a singleton state-machine that can transition between states via cryptographically-secured transactions. When generating a new state-machine, the nodes encode logic which defines valid state transitions and upload it onto the blockchain. From there on, the blocks journal a series of valid transactions that, when incrementally executed with the state from the previous block, morph the state-machine into its current state. The Proof of Work consensus algorithm and its underlying peer-to-peer protocol secure the state-machines' state and transitioning logic from tampering, and also share this information with all nodes participating in the system. Nodes can therefore query the state-machines at any time and obtain a result which is accepted by the entire network with high certainty.

This transaction-based state-machine generalization of the blockchain is informally referred to as smart contracts. Ethereum is the first to attempt a full implementation of this idea. It builds into the blockchain a Turing-complete instruction set to allow smart-contract programming and a storage capability to accommodate on-chain state. We regard the flexibility of its programming language as an important property in the context of EMR management. This property can enable advanced functionality (multi-party arbitration, bidding, reputation, etc.) to be coded into our proposed system, adapting to comply with differences in regulation and changes in stakeholders needs.

We utilize Ethereum's smart contracts to create intelligent representations of existing medical records that are stored

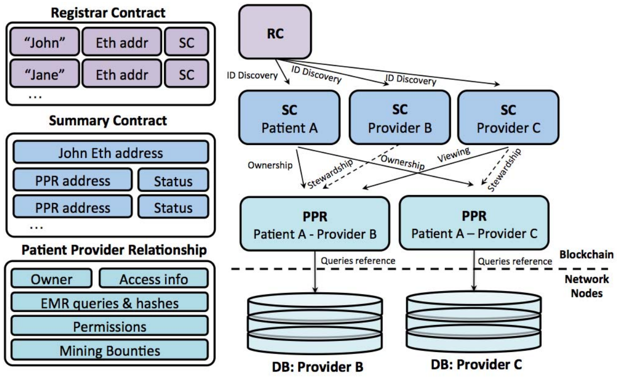

Figure 1. MedRec smart contracts on the blockchain and data references

within individual nodes on the network. We construct the contracts to contain metadata about the record ownership, permissions and data integrity. The blockchain transactions in our system carry cryptographically signed instructions to manage these properties. The contract's state-transition functions carry out policies, enforcing data alternation only by legitimate transactions. Such policies can be designed to implement any set of rules which govern a particular medical record, as long as it can be represented computationally. For example, a policy may enforce that separate transactions representing consent are sent from both patients and care providers, before granting viewing permissions to a third party.

To navigate the potentially large amount of record representations, our system structures them on the blockchain by implementing three types of contracts. Figure 1 illustrates the contract structures and relationships.

1) Registrar Contract (RC): This global contract maps participant identification strings to their Ethereum address identity (equivalent to a public key). We intentionally use strings rather than the cryptographic public key identities directly, allowing the use of already existing form of ID. Policies coded into the contract can regulate registering new identities or changing the mapping of existing ones. Identity registration can thus be restricted only to certified institutions. The RC also maps identity strings to an address on the blockchain, where a special contract described below, called the Summary Contract, can be found.

2) Patient-Provider Relationship Contract (PPR): A Patient-Provider Relationship Contract is issued between two nodes in the system when one node stores and manages medical records for the other. While we use the case of care provider and patient, this notion extends to any pairwise data stewardship interaction. The PPR defines an assortment of data pointers and associated access permissions that identify the records held by the care provider. Each pointer consists of a query string that, when executed on the provider's database, returns a subset of patient data. The query string is affixed with the hash of this data subset, to guarantee that data have not been altered at the source. Additional information indicates where the provider's database can be accessed in the network, i.e. hostname and port in a standard network topology. The data queries and their associated information are crafted by the care provider and modified when new records are added. To enable patients to share records with others, a dictionary implementation (hash table) maps viewers' addresses to a list of additional query strings. Each string can specify a portion of the patient's data to which the third party viewer is allowed access.

Our prototype demonstrates this design with SQL data queries. In a simple case, the provider references the patient's data with a simple SELECT query conditioned on the patient's address. For patients, we designed a simple tool which allows them to check off fields they wish to share through our graphical interface. Under the hood, our system formulates the appropriate SQL queries and uploads them to the PPR on the blockchain.

Note that by using generic strings our design can robustly interface with any string queried database implementation. Hence, it can conveniently integrate with existing provider data storage infrastructure. At the same time, patients are enabled with fine-grained access control of their medical records, selecting essentially any portion of it they wish to share.

3) Summary Contract (SC): This contract functions as a bread crumb trail for participants in the system to locate their medical record history. It holds a list of references to Patient-Provider Relationship contracts (PPRs), representing all the participant's previous and current engagements with other nodes in the system. Patients, for instance, would have their SC populated with references to all care providers they have been engaged with. Providers, on the other hand, are likely to have references to patients they serve and third-parties with whom their patients have authorized data sharing.

The SC persists in the distributed network, adding crucial backup and restore functionality. Patients can leave and rejoin the system multiple times, for arbitrary periods, and always regain access to their history by downloading the latest blockchain from the network. As long as there are nodes participating in the network, the blockchain log is maintained.

The SC also implements functionality to enable user notifications. Each relationship stores a status variable. This indicates whether the relationship is newly established, awaiting pending updates and has or has not acknowledged patient approval. Providers in our system set the relationship status in their patients' SC whenever they update records or as part of creating a new relationship. Accordingly, the patients can poll their SC and be notified whenever a new relationship is suggested or an update is available. Patients can accept, reject or delete relationships, deciding which

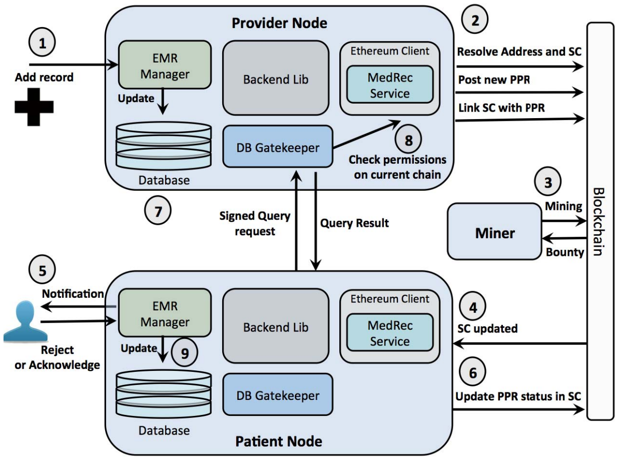

Figure 2. System orchestration: provider adds a record for new patient

records in their history they acknowledge.

Our prototype ensures that accepting or rejecting relationships is done only by the patients. To avoid notification spamming from malicious participants, only providers can update the status variable. These administration principles can be extended, adding additional verifications to confirm proper actor behavior.

C. System Node

We design the components of our system nodes to integrate with existing EMR infrastructure. We assume that many nodes, and in particular care providers, already trustfully manage databases with patient data stored on servers with network connectivity. Our design introduces four software components: Backend Library, Ethereum Client, Database Gatekeeper and EMR Manager. These can be executed on servers, combining to create a coherent, distributed system. We provide a prototype implementation of these components that integrates with a SQLite database and is managed through our web user interface. Notably, any provider backend and user interface implementations can participate in the system by employing the modular interoperability protocol as defined through our blockchain contracts.

Patient nodes in our system contain the same basic components as providers. An implementation of these can be executed on a local PC or even a mobile phone. Their local database can be one of many lightweight database implementations. The databases can function merely as cache storage of the patient's medical data. Missing data can be retrieved from the network at any time by following the node's Summary Contract.

1) Backend Library: We construct multiple utilities, bundled in a backend library, to facilitate the system's operation. Our library abstracts the communications with the blockchain and exports a function-call API. Record management applications and their user interfaces can thus avoid the hurdles of working directly with the blockchain. One such hurdle is verifying that each sent transaction is accepted with high confidence by the network. Our library automatically handles the uncertainty of when transactions are mined and deals with cases when they are discarded. The backend library interacts with an Ethereum client to exercise the low-level formatting and parsing of the Ethereum protocol.

Steps 1 and 2 in Figure 2 illustrate our backend implementation of a scenario where a provider adds a record for a new patient. Using the Registrar Contract on the blockchain, the patient's identifying information is first resolved to their matching Ethereum address and the corresponding Summary Contract is located. Next, the provider uploads a new PPR to the blockchain, indicating their stewardship of the data owned by the patient's Ethereum address. The provider node then crafts a query to reference this data and updates the PPR accordingly. Finally, the node sends a transaction which links the new PPR to the patient's Summary Contract, allowing the patient node to later locate it on the blockchain.

2) Ethereum Client: This component implements the full functionality required to join and participate in the Ethereum blockchain network. This handles a broad set of tasks, such as connecting to the peer-to-peer network, encoding and sending transactions and keeping a verified local copy of the blockchain. For our prototype implementation we use PyEthereum and the PyEthApp client.

We modify the client to be aware of our mapping of identity and addresses. We then implement a service to locate the node's Summary Contract (SC), via Registrar Contract address lookup. This service runs continuously within the client to monitor real-time changes to the SC. In the event of an update, the service signals the EMR Manager to issue a user notification and, if necessary, sync the local database.

Steps 4 to 6 in Figure 2 continue the use case described above from the patient node perspective. The patient's modified Ethereum client continuously monitors her SC. Once a new block is mined with the newly linked PPR, the client issues a signal which results in a user notification. The user can then acknowledge or decline her communication with the provider, updating the Summary Contract accordingly. If the communication is accepted, our prototype implementation automatically issues a query request to obtain the new medical data. It uses the information in the new PPR to locate the provider on the network and connect to its Database Gatekeeper server.

3) Database Gatekeeper: The Database Gatekeeper implements an off-chain, access interface to the node's local database, governed by permissions stored on the blockchain. The Gatekeeper runs a server listening to query requests from clients on the network. A request contains a query string, as well as a reference to the blockchain PPR that warrants permissions to run it. The request is cryptographically signed by the issuer, allowing the gatekeeper to confirm identities. Once the issuer's signature is certified, the gatekeeper checks the blockchain contracts to verify if the address issuing the request is allowed access to the query. If the address checks out, it runs the query on the node's local database and returns the result over to the client.

Steps 7 to 9 in Figure 2 illustrate how a patient retrieves personal data from the provider node. Note that our components similarly support third-parties retrieving patient-shared data: the patient selects data to share and updates the corresponding PPR with the third-party address and query string. If necessary, the patient's node can resolve the third party address using the Registrar Contract on the blockchain. Then, the patient node links their existing PPR with the care provider to the third-party's Summary Contract. The third party is automatically notified of new permissions, and can follow the link to discover all information needed for retrieval. The provider's Database Gatekeeper will permit access to such a request, corroborating that it was issued by the patient on the PPR they share.

4) EMR Manager: We tie together all the software components previously mentioned with our EMR management and user interface application. The application renders data from local SQLite databases for viewing, and presents the users with update notifications, and data sharing and retrieval options. Our user interface prioritizes intuitive, crisp, and informative design, as recommended by the Department of Veteran Affairs' Blue Button competition [13]. Our application is conveniently accessed through a web-based interface, built on a python micro-framework called Flask. Through this choice, we are especially cognizant of compatibility for Mobile devices, as modern users expect easy access and high-quality experiences while on-the-go.

D. Mining

We incentivize "miners" to participate in the network and contribute their computational resources to achieve a trustworthy, gradual advancement of the chain. We propose two incentivizing models, taking into account the diverse interests that stakeholders share in a healthcare ecosystem.

The first is based on Ethereum's inherent incentivizing model. In this model, transactions require Ether, a network currency unit, to be processed by the network. Ether can be earned by mining, awarding an acceptable amount of it to a node that solves the computational puzzle. Care providers are thus incentivized to participate in mining in order to fund the continuation of their activities (posting and updating PPRs, accepting viewing permissions, et cetera). Likewise, when patients wish to share their information, they will be required to spend Ether or have the destination party fund them. Seeding patients with Ether or having them pay for it can be determined by health care regulation.

A second incentivizing model brings medical researchers and health care authorities to mine in the network. In return the network beneficiaries, i.e. providers and patients, release access to aggregate, anonymized medical data as mining rewards. We explore this idea in our prototype by implementing a special function in the PPR. It requires care providers to attach a bounty query to any transaction they send updating the PPR. For example, this bounty query can be formulated to return the average iron levels in blood tests done by the provider, across all patients, in the previous week. When the block containing the transaction is mined, the function automatically appends the block's miner as the owner of the bounty. The miner can then collect it by simply issuing a request for this bounty to the provider's Database gatekeeper. Because it is signed by the provider as part of the transaction, the bounty query is safe from malicious alterations.

IV. Discussion

MedRec gives patients a log of their medical history, which is not only comprehensive, but also accessible and credible. This restores patient agency, as participants are now more fully informed of their medical history and any modifications to it. Through permission management on the blockchain, we enable patient-initiated data exchange between medical jurisdictions. To respect the need for confidentiality at a granular scale [3], MedRec allows for specific authorizations. Different metadata fields within a *single* record can be shared separately and may include further restrictions such as an expiration date for viewership rights (enabled via the smart contract provisions). The blockchain ledger keeps an auditable history of medical interactions for patients, providers and regulators.

By integrating with providers' existing data storage infrastructure, we facilitate continued use of their existing systems. We believe this will ease adoption and aid compliance with HIPAA regulations. Building on the principle of interoperability, we have designed the system with flexibility to support open standards for health data exchange– be that FHIR and other flavors of HL7 [14], or combination proposals like the Continuity of Care Document [15]. In addition, MedRec is source agnostic, i.e. able to receive data from any number of endpoints (physician offices, hospital servers, patient home computers, et cetera).

Our blockchain implementation gives us several key properties of decentralization. MedRec enjoys a strong failover model, relying on the many participating entities in the system to avoid a single point of failure. Medical records are stored locally in separate provider and patient databases; copies of authorization data are stored on each node in the network. Furthermore, because the medical data stays distributed, our system does not create a new, central target for content attack.

Notably, MedRec does not claim to address the security of individual databases– this must still be managed properly by the local system admin. Nor does MedRec attempt to solve the Digital Rights Management problem. Our system assumes provider nodes that are bound by external regulation governing data copying in the medical use case, i.e. HIPAA.

Use of blockchain technology introduces several limitations. The pseudonymous property of transactions currently allows for data forensics, or inferring patterns of treatment from frequency analysis. Though a person's name and PII may remain private, one could infer that some individual has repeatedly interacted with a certain provider. Improving obfuscation while preserving auditability on the blockchain is an ongoing area of exploration. Though we have not explicitly added contract encryption in the initial prototype, our system could be easily modified to do so. In addition, blockchain implementations still grapple with how best to scale the technology for high transaction volume. This may affect our system, determining the natural size of each MedRec community (i.e. whether a single implementation of MedRec could support a regional consortium of providers, or hundreds of providers across a larger scale).

Even well distributed log data must still be sustained, however, and we believe our proposal of data as a mining incentive answers a pressing need in the medical research community while supplying effective Proof of Work. Our platform enables the emergence of data economics, matching demand and supply between data producers and consumers. Researchers can influence the bounties that providers propose by selecting which transactions to mine and validate. Providers are then incentivized to match what researchers are willing to accept, within the boundaries of proper privacy preservation. Research miners can now access a regular source of anonymized, large-scale medical data. This opens the opportunity to observe wide-reaching patterns in medical treatment, while still preserving the privacy of individuals and lowering the overhead associated with traditional research trials. While outside the scope of the initial prototype (but unarguably crucial for future development), a rigorous k-anonymity analysis of privacy-preserving query construction is needed.

V. Conclusion and Future Work

Leveraging blockchain technology, MedRec has shown how principles of decentralization might be applied to large-scale data management in an EMR system. We demonstrate an innovative approach for handling medical records, providing auditability, interoperability and accessibility via a comprehensive log. Designed for record flexibility and granularity, MedRec enables patient data sharing and incentives for medical researchers to sustain the system. We look forward to formalizing an onboarding procedure for medical research "miners", and exploring mining data economics. In the near future, we intend to carry out user studies to assess the feasibility of the system and to gauge patient and provider interest. This may include partnering with local healthcare bodies, and simulating aspects of system efficiency in the wild. We remain committed to the principles of open source software and intend to make our framework available as a platform for further development.

Acknowledgment

We thank the MIT Digital Currency Initiative and the MIT Media Lab Consortium for their support.

References

[1] Health Information and the Law, "Who owns medial records: 50 state comparison," 2015. [Online]. Available: \url{http://www.healthinfolaw.org}

[2] U.S. Department of HSS, "Hipaa administrative simplification: Regulation text," 2006.

[3] K. D. Mandl *et al.*, "Public standards and patients' control: how to keep electronic medical records accessible but private," *BMJ*, vol. 322, no. 7281, pp. 283–287, 2001.

[4] The Office of the Nat. Coordinator for Health Information Technology, "Report on health information blocking," U.S. Department of HHS, Tech. Rep., 2015.

[5] U.S. Department of HHS, "Individuals' right under hipaa to access their health information," 2015. [Online]. Available: \url{http://www.hhs.gov}

[6] M. McGinnis *et al.*, *Clinical Data as the Basic Staple of Health Learning: Creating and Protecting a Public Good.* National Academies Press, 2010.

[7] L. J. Kish and E. J. Topol, "Unpatients– why patients should own their medical data," *Nature biotechnology*, vol. 33, no. 9, pp. 921–924, 2015.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *White Paper*, 2008.

[9] G. Zyskind *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *Security and Privacy Workshops (SPW), 2015 IEEE.* IEEE, 2015, pp. 180–184.

[10] Factom, "Healthnautica + factom announce partnership," 2015. [Online]. Available: \url{http://blog.factom.org}

[11] CoinDesk, "Medical records project wins top prize at blockchain hackathon," 2015. [Online]. Available: \url{http://www.coindesk.com}

[12] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, 2014.

[13] U.S. Department of VA, "The patient record," 2013. [Online]. Available: \url{http://healthdesignchallenge.com}

[14] HL7 International, "Fhir overview," 2015. [Online]. Available: \url{https://www.hl7.org}

[15] Corepoint Health, "The continuity of care document: Changing the landscape of healthcare information exchange," *White Paper*, 2009.