Blockchain Computing Systems in Healthcare

By Conrad Barski, M.D.[1], Stephen Claypool, M.D.[2], and Tony Little, N.D.[3] [4]

Introduction

Academic computer science first started to notice the technology underlying Bitcoin in early 2013. Reputable researchers such as Dan Kaminsky[5] began to seriously analyze the technology underlying cryptocurrencies, principally the *blockchain* data structure. Also around this time, the first discussions appeared suggesting benefits of using blockchain technology for enterprise healthcare systems.[6] Now, several years later, the potential for these ideas is capturing the attention of the broader medical information technology (IT) community.

There are different approaches for applying blockchain technology to healthcare. One of these approaches in particular, called the *blockchain computing system* approach, can offer significant improvements over existing IT architectures in healthcare and can be reasonably implemented on top of existing technologies. Given the current excitement around blockchains, and the advantages they offer over traditional systems, it is likely that many of the concepts described in this paper will soon be leveraged in production systems, both in healthcare and other fields.

The Two Main Challenges of Modern Hospital IT

Any discussion focused on improving the performance of IT systems in medicine must begin with an evaluation of the IT infrastructure in modern hospitals. Without adequate tools in the hands of front-line medical professionals, the quality of clinical data on patients suffers and impairs other parts of our medical infrastructure, such as medical research and epidemiological forecasting, which are heavily dependent on accurate frontline data. Similarly, new treatment protocols originating from medical research, as well as the implementation of new health care policies recommended by government entities, have a much better chance of leading to improved patient outcomes if front-line providers are given the most modern and well-designed healthcare applications possible.

Based on our extensive experience with medical IT, it is our opinion there are two major technical obstacles that are significantly limiting the advancement of healthcare software systems at this time:

1. Limitations in software functionality deriving from **poor interoperability of software across vendors.**
2. Operational difficulties deriving from **poor security inherent in current systems**.

After first describing the properties of medical blockchain systems in the next few sections, we will examine these obstacles in greater depth to discover how they may be overcome through the use of the blockchain.

Two Different Designs for a Medical Blockchain System

The first step to resolving problems in medical IT with blockchain systems is to understand the different approaches that can be taken in applying this technology in healthcare. Given the fundamental requirement for data security in all healthcare systems, applicable approaches can be placed into two broad categories:

1. **Blockchain Storage Systems:** This approach, first described in the context of Bitcoin, focuses on the "data permanence" of blockchain data. A blockchain storage system would typically store "existence proofs" for patient documents in a widely shared public blockchain, by using a cryptographic hash. In this way, for healthcare, a detailed audit history could be established for a patient's records without compromising any identifiable patient health information (PHI). With this design, a separate mechanism is still required for storing the full version of the patient documents, typically using some sort of encrypted key-value datastore.

---

[1] Lead architect for blockchain technologies, [company name redacted]. 20 years of experience in Healthcare IT.
[2] Medical Director, Point of Care Advisor, Wolters Kluwer Health. 20 years of experience with Health Informatics.
[3] Senior Director for Integration Strategy, Optum360. 18 years experience with Healthcare IT.
[4] Special thanks to Jans Aasman, Ph.D. (CEO of Franz, Inc.) for his comments on drafts of this whitepaper.
[5] http://www.businessinsider.com/dan-kaminsky-highlights-flaws-bitcoin-2013-4
[6] For one example, see: https://letstalkbitcoin.com/forum/post/current-anonymous-healthare-data-on-the-blockchain

2. **Blockchain Computing Systems:** This alternate approach treats the Blockchain as a secure computation platform that can be shared among multiple vendors. This approach was pioneered by ethereum[7], the first practical platform for deploying smart contracts. Smart contracts allow for a sandboxed form of computation with strong reliability guarantees, as well as strong guarantees against denial of service attacks. In a healthcare context, a *blockchain computing system* (unlike a *blockchain storage system*) maintains all patient data directly on the blockchain, on a private network and in an encrypted format.

Even though most attention so far for medical blockchain systems has been put into approach #1, the most fruitful design for a healthcare blockchain system would be based on #2. The rest of this paper will focus on such an architecture, outlining a medical IT system designed to unlock the vast *computational* capabilities of blockchain systems.

The General Structure of a "Blockchain Computing System"

The primary goal of a *blockchain computing system* (BCS) is to allow computer code written by multiple vendors to run in a single, shared computing space, without the risk of systemic failures caused by errors in any individual unit of code. To make this possible, all necessary data needs to be stored on the blockchain[8] not merely cryptographic hashes (as is the case with *blockchain storage systems.)* This allows the majority of the code in the system to be written as smart contracts (such as the smart contract system provided by ethereum) which are an ideal platform for software when it requires strong security and reliability guarantees.

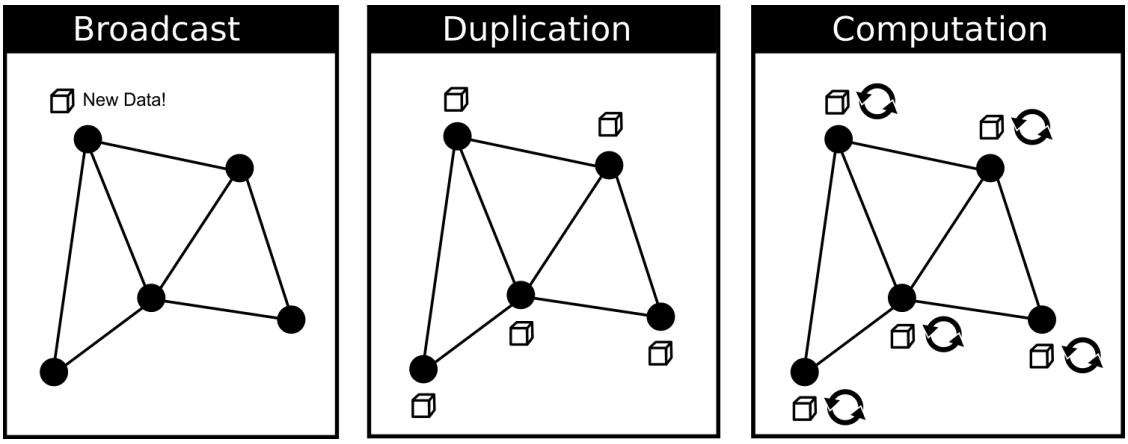

Figure 1: In a BCS, new data acquired by a node is **broadcast** to all participants on the network, in the form of a cryptographically-signed transaction. Each node then **duplicates** this data and stores it locally. Finally, identical **computations** are executed at each node in response to the new data. No central server exists with this type of architecture.

(*Note: The rest of this whitepaper will assume that the BCS in question is implemented specifically on top of the ethereum technology stack.)*

A smart contract is a small piece of computer code that is deployed into a blockchain using a standard blockchain transaction. Smart contracts can run arbitrary computation, but have severe restrictions for safety reasons:

- The only data a smart contract can access is a subset of previous transactions in the network[9] and a small number of blockchain specific data items, such as information about the current time.
- The only actions a smart contract can perform are to call functions in other smart contracts or to generate new blockchain transactions.

[7] https://ethereum.org/

[8] In a medical context this means all structured patient data would be stored in the blockchain. However, large and unstructured data types, such as scanned documents and radiological data, are stored elsewhere.

[9] In the ethereum system, historical transaction data intended for a smart contract can also be organized ahead of time by placing it into a small datastore managed by an individual smart contract.

- All computation by a contract requires the payment of transaction fees, which are paid for in a type of "private currency" provided by the maintainers of the network.[10] These fees prevent malfunctioning or malicious contracts from negatively impacting the network via denial of service.

Since a BCS requires most data to be stored directly on the blockchain, it is essential that any such system operating on sensitive data (such as in a healthcare context) properly secures the blockchain data. This means the system should maintain the following requirements:

1. No blockchain data should ever be written to permanent storage (hard drives, etc) on any computer in the system without first being encrypted.
2. The blockchain needs to be a private blockchain maintained on a private network.
3. All computers that have access to this data should be managed computers that are not able to run arbitrary applications or access public networks.

A Minimally Viable Hospital Enterprise System with Blockchain Computing Support

The architecture outlined here is designed to leverage existing software as much as possible, needing only modest additional software development to meet all requirements for usage in the healthcare field.

With this design, a single *private blockchain* is maintained for each physical healthcare facility. There would be three types of computers in the facility:

1. **Client workstations** for doctors, nurses, and other hospital staff to log into and which they can use to access patient data.
2. **Single-purpose gateway appliances** which connect the private blockchain to outside systems, such a diagnostic monitors, lab order entry systems, etc.[11]
3. **Backup clients** in the data center to provide additional redundancy.

Unlike with traditional "multi tier" hospital enterprise software, there would be no strict dependencies between these different machines; all machines, in all three categories, would be responsible for storing a full copy of the blockchain data, and no hierarchical relationship would exists between the PCs. Additionally, all PCs in the network would share similar technical specifications:

1. **They would all have a generously-sized hard drive (multiple terabytes) for storing patient data-** With this design, every machine in the hospital would contain a fully encrypted (but complete) copy of all relevant patient records.
2. **They would be fully enclosed/self contained and have restricted access to IO ports.** This is possible because the architecture of this system will not require local IT administrative staff to perform any tasks on the machine (such as software updates, etc) that require physically interacting with the computer.[12]
3. **An ethereum node would be transparently running as a service on the device.** This ethereum node would use its node discovery mechanisms to connect to other ethereum nodes on the hospital network. All medical rules in the hospital would be encoded as smart contracts in the blockchains maintained by these nodes.
4. **A single UI application runs on these machines: A stripped-down web browser designed to serve up data retrieved from the blockchain.** All patient data, application resources, and computer code is retrieved from this blockchain. This application is incapable of using the network or accessing any secondary databases or other data sources.
5. No other access is given to a user of the system to any other applications or the file system. Also, a user will have no ability to directly access the ethereum node that isn't explicitly enabled via the UI application.

---

[10] In most current blockchain systems, the maintainers of the network, which provide this currency, are called "miners". However, alternative models also exist.

[11] See HL7 standards for examples of how medical data is typically transmitted between hospital systems: http://www.hl7.org/.

[12] For exceptional circumstances, a hardware reset function with admin features would still be available, for low-level hardware and base software updates. However, activating it would begin with a full data wipe of the system.

To connect to one of these computers, an authorized person (nurse, doctor, other staff person) needs to enter two passwords into the system:

- One that allows the blockchain (and patient data therein) to be decrypted for access. This is the first line of data protection.
- Another key that is tied to the provider's identity and serves two important uses:
    a. **Authentication:** It is used by the UI front end to verify that a provider has the necessary permissions to access a patient's medical record.
    b. **Transaction Signing:** Because this is a blockchain system, all mutating operations against the patient records need to be signed. These signatures offer documentation of authorship of all new data in the system, required for trustworthy execution of medical rules in the system. Also, they allow "network fees" to be attached to all actions against the blockchain, offering protection from "Denial of Service"[13] performed against the hospital network.

(Note: For technically-minded readers, there is a more precise technical specifications of this "minimally viable BCS" available online.[14])

Because all PCs in the network share similar technical specifications, including generous storage capacity, it becomes possible for all computation in the hospital to be performed redundantly and in parallel by every computer on the network. This will significantly improve the security, reliability, and performance of the system.

And because blockchain code within the network can be written by multiple vendors and deployed on a single, shared computing space, interoperability between vendor applications can be greatly improved.

A Concrete Example of Traditional vs. Blockchain Computing in a Hospital Environment

This architecture is radically different from traditional systems and offers significant technical advantages over existing designs. To see why, let's look at a concrete example of how this system would function when handling a typical medical scenario:

> *Lisa Jones is a 29yo patient admitted to the hospital for gallbladder surgery. During her recovery, a nurse noted redness and purulent drainage at the site of her surgical wound, followed by a temperature of 101.5F and a blood pressure of 80/60. These signs may portend sepsis, a potentially life-threatening complication.*

The correct action in this scenario for a modern point-of-care diagnostic computing system in a hospital would typically be to:

1. Immediately notify key hospital staff.
2. Immediately guide staff to order relevant lab tests, including blood cultures for this patient (possibly even ordering them preemptively without human intervention, if there is an established protocol in the hospital for doing so.)
3. Guide staff to deliver expedient and appropriate treatment - for this patient, IV fluids, antibiotics and more intensive monitoring.

Now, let's look at the operational dataflow of this diagnostic software system, contrasting how it would differ if the system was built on a traditional systems architecture versus a BCS system.

STEP 1 - Acquiring New Data For our example, let's consider how the patient's blood pressure is retrieved from a vitals monitor and recorded into the patient's medical record:

Traditional System	**Blockchain Computing System**
With a traditional system, a software service provided by an	With a BCS, the data is encoded as a cryptographically

[13] https://en.wikipedia.org/wiki/Denial-of-service_attack
[14] https://goo.gl/nIAQBp

EHR system will typically connect to a bedside vitals monitor using a TCP socket connection or FTP connection to retrieve HL7 encoded data files and directly enter those values into a centralized database.	signed ethereum transaction within the vitals monitor (or via a physically connected gateway appliance for legacy monitors.) A BCS computer then passively broadcasts this transaction into the network.

The traditional approach for acquiring data from external devices, as described above, takes several unnecessary security risks that do not exist in a system based on blockchains:

1. **Traditional systems do not cryptographically sign data generated by external devices**. Instead, a separate EHR background service is set up with broad access rights to the EHR database, which is trusted to pull data from the vitals monitor or other external device on a regular basis and directly insert this data into the database. In this process, the audit trail linking a piece of data to the external device needs to be maintained manually, if it is maintained at all. Subsequently, any services that broker requests to access the patient records need to decipher the "trustworthiness" of the system making the request. Any error in "software-enabled judgement" could put the data at risk (i.e. allow an attacker submit forged data). The blockchain system, on the other hand, allows a service to be built that is entirely passive: It can retrieve data from a vitals monitor in the form of signed transactions, and record it into the patient's medical records. However, this service would have no power to alter the data, edit metadata, and would have no access at all to modify patient records. This means that medical rules running against the patient database can have a far better assurance as to the originating device that created data, in a manner that is *tamper proof*.
2. **Traditional systems do not use a "write only" protocol for interacting with external devices.** Currently, most devices like vitals monitors will have have ethernet support and use bi-directional data protocols, such as TCP or FTP. Hospital IT staff have to go to great lengths to make sure this does not pose a security risk, since this means devices have direct access to network traffic inside the hospital firewall if they are not configured properly, since ethernet, TCP, and FTP were designed as open systems. If access is not properly restricted, a vitals monitor compromised by a hacker could obtain access to information about the hospital network or protected health information (PHI). Blockchain systems, on the other hand, establish well-defined methods for creating signed transactions offline. In this way, vitals monitor software could be engineered in a way that can only write patient data to the blockchain, and even if it is compromised it would not allow a hacker to gain information about the hospital network.

In addition to improving secure transactions with devices, the BCS also enhances interoperability. In a BCS, the device vendors are responsible for blockchain code that will interact with the BCS. Other vendors do not need to make software changes to accommodate a variety of devices.. Because of this, devices can be freely exchanged on the network without disruption to other software systems.

STEP 2 - Recording the New Data in the Patient Record Now that the patient's blood pressure has been retrieved, let's look at how this fact is recorded with the other health information for a patient:

Traditional System	**Blockchain Computing System**
A trusted server, with broad access rights to the EMR database, writes the blood pressure into the appropriate tables.	In a BCS, all entities that are allowed to write to the patient records have a public key registered with a special smart contract on the block chain that represents that entity, typically called a *proxy contract* or *identity contract*. This means every vitals monitor in the hospital (or in a legacy situation, an associated gateway appliance with each vitals monitor) possesses a unique signing key.

Almost every traditional hospital application will contain an application server with services that act as a gateway between outside data sources and a central database. Because of the broad responsibilities of this server, rights assigned to its services will often encompass **full permissions to modify** the central database. This is generally unavoidable, because the server must posses a superset of all permissions required for writing all data by outside data sources into the patient record.

A blockchain system with smart contract capabilities (like ethereum) on the other hand, requires no such server: All

permissions are associated with the data sources, themselves: Every single device in the hospital can be given highly granular permissions to access the record. For example, a single vitals monitor may only have the rights to write a single data type to a single patient, all mediated by proxy contracts stored on the blockchain. At most, a blockchain system would have one or more passive servers that relay transaction from outside systems, but with **no permissions to modify** the transactions or impersonate these outside systems.

STEP 3 - Data Storage Once it has been determined that proper security steps were followed to add the blood pressure to the patient record, the new data element is written to the storage medium:

Traditional System	**Blockchain Computing System**
A centralized database server records the blood pressure value to the database tables maintained on its hard drive. Additionally, incremental backups may be performed to store the data in other independent virtual and physical locations.	As soon as the transaction that documents the new blood pressure value appears on the network, **all computers** in the system immediately store the data element to an encrypted copy of the blockchain on their local hard drive.

When it comes to data storage, a BCS differs radically from a traditional system. The workflow for data storage and retrieval are as follows:

- Traditional system: Clients send data to server → Server persists data → Clients query server for data
- BCS: Clients broadcast data on network → Clients persist data → Clients run queries locally

STEP 4 - Computation After the blood pressure has been added to the patient record, an answer to an important question needs to be computed: "Is the patient more likely to have sepsis, given the new information?" Here is how this computation is performed:

Traditional System	**Blockchain Computing System**
An application server is set up, usually with **broad rights for reading and writing to the database**. This server runs a traditional **general purpose application** written in a general purpose programming language. This application queries the database, processes medical rules, and then writes the results back to the database. A rules engine, such a drools[15], may be activated for part of this process.	A rule encoded in a smart contract is triggered on all computers in the system. This smart contract is **completely sandboxed**, with no ability to communicate with the outside world. The only data it can query is a small subset of previous transactions recorded on the blockchain. The only data it can generate as output is a new transaction that is also restricted to the blockchain.

Traditionally, when a new vendor wants to deploy a new software system at a hospital, they will first engage in months of discussions with the hospital's IT team to explain what kind of access the new system needs to other systems on the network (or even outside the network) and to verify that proper procedures are followed to safeguard all patient data. This lengthy process, driven by the need to meet HIPAA requirements, is unavoidable with traditional systems.

A BCS, on the other hand, allows new software to be deployed in a way that makes HIPAA violations completely impossible. In a blockchain environment, new software code can be deployed as smart contracts into the blockchain and can immediately run against all following medical data. These smart contracts can accomplish an impressive technical feat: They allow developers to write code that has broad access to patient data but in a way that still makes it impossible to have unintentional (or intentional) leaks of data into the outside world. *The only output of these smart contracts are additional data elements written into the blockchain record*.

BCS systems will reduce waste by limiting the expensive IT resource time traditionally required for new systems. Furthermore, they enhance interoperability by shortening deployment time of vendor systems that communicate with the BCS, enabling more rapid integration of new systems.

STEP 5 - Notifying Providers of Computation Results After the system has determined that the patient's low blood pressure may put them at risk for sepsis, it will likely need to place lab orders and notify appropriate providers to examine the patient:

[15] http://www.drools.org/

Traditional System	Blockchain Computing System
Clients continually *poll* the application server for new alerts for patients to display to providers. Or, the system relies on a push mechanism (such as Google or Amazon's push notification systems) to alert providers.	**No communication needed-** Each device already has acquired the blockchain transactions in an earlier step and determined its own computation results- Provider can be notified immediately[16].

With traditional systems, every application running in the hospital usually establishes it's own server→client communication protocol for notifying clinicians of alerts. These often involve external systems (especially when alerts need to be received by pager systems or mobile devices) and all represent vulnerability points for hackers.

With a BCS, on the other hand, the only communication that happens is *at the time the new data is first broadcast as a transaction on the blockchain network.* From that point on, a client machine has all the data and code it needs to trigger the appropriate alerts to providers- no additional communication layer needs to take place that can put patient health information at risk- this entire step does not exist with a BCS.

STEP 6 - Performing Direct Actions Based on Computation Results Aside from alerting providers, the system will also need to perform direct actions for a septic patient, such as initiating lab orders:

Traditional System	Blockchain Computing System
An ad hoc application service with access rights to both systems moves lab results from the diagnostic system to the lab order entry system. The data is usually encoded using a predetermined protocol. Messages are transmitted over the hospital network, usually in the form of HTTPS web service request calls or HL7 v2 messages over TCP/IP.	Rules encoded in smart contracts directly deploy orders as cryptographically secure transactions on the blockchain. These orders are directly visible to a lab technician, who is authorized to access the blockchain data maintained on every computer in the system.[17]

In a traditional system, the different subcomponents of the hospital infrastructure are heavily siloed; in order to get information from one silo (the point of care diagnostics system) to another silo (lab system) there are usually at least three services in play. One, a service to query, format, and export the data out of the point-of-care system. Two, another service (usually an integration engine) to translate and route data to the lab system. And three, a service associated with the lab system to receive, parse, and insert the data into the lab system's central database. Not only do all of these services open up additional opportunities for nefarious actors to access PHI but upgrades/updates to any of these layers can have the unintended consequence of breaking the communication chain. The expensive nature of siloed solutions within the traditional systems results in poor interoperability of vendor software. Updates are required to share newly available information sources, yet updates are delayed (or not deployed) because of costs.

With a BCS, multiple vendors can safely deploy software in the same computing environment, in the form of smart contracts. This means that patient data can be moved between different applications without ever leaving the single, heavily protected blockchain data structure. Additionally, atomic updates to vendor software is easier, enhancing interoperability by allowing sharing of new data sources more readily.

[16] In some circumstances, a delay of 1-3 seconds may be advised to make sure likely block transaction order is known before notifying users.

[17] In the instance where the ordering system uses a legacy design, a gateway device connected to the ordering system will relay orders using the traditional (less secure) method involving application service calls.

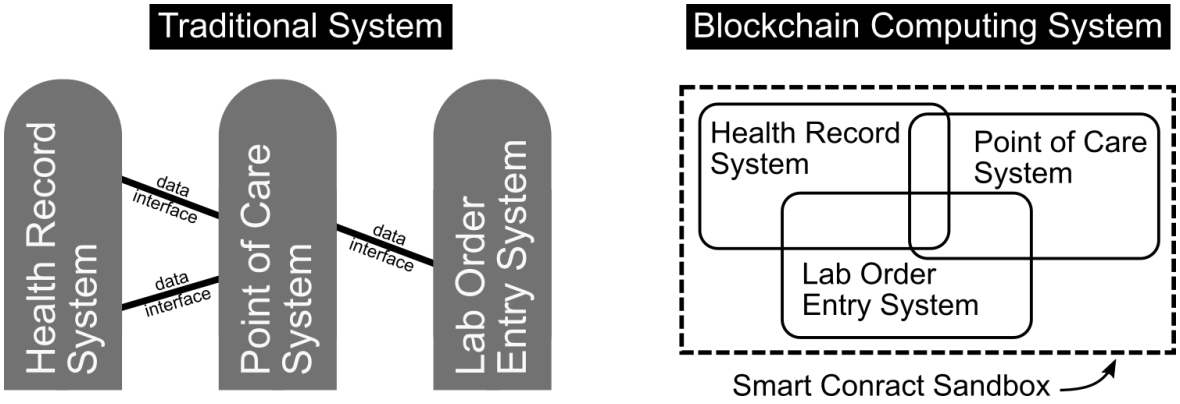

Figure 3: Traditionally, hospital software is divided into "data silos" in which each vendor's system is highly isolated with limited communication between each silo. In a blockchain computing system, on the other hand, vendor code can be much more tightly integrated, and offer a more seamless experience to clinicians. This is enabled by the persistence, denial of service protections, and PHI isolation provided by a shared smart contract environment.

With a blockchain system, diagnostic rules from one vendor can trigger orders in an medical order entry system provided by another vendor, both parties can perform arbitrary application-specific computation on HIPAA-sensitive data, in a way that is not ad hoc, is heavily sandboxed at all times, and requires no implicit trust among the individual vendors regarding proper handling of patient data.

Solving the Challenges of Medical IT with a Blockchain Computing System

The Poor Security Model of Current Computer Systems

Ever since the HIPAA Privacy and Security Rules went into effect in April 2003, hospital IT and compliance staff at hospitals have been obligated to undergo strict procedures for every new software product that is deployed at a facility. Though these procedures are extensive, there are several high level components:

- Hospital networks are placed behind a rigorous firewall to create obstacles for any attempts to compromise patient health information.
- All vendor software deployed on the network is carefully vetted to make sure it complies with HIPAA guidelines.
- All providers in the hospital are educated by hospital IT and compliance staff about appropriate HIPAA practices.

These procedures are sensible when dealing with traditional hospital systems, where every database system, application server, or computer terminal can run arbitrary software while at the same time having access to sensitive information. Unfortunately, these procedures create an enormous burden on both vendors and hospital staff; meanwhile, evidence of failures in protecting PHI in hospitals continues to increase year over year.[18]

One possible way to maintain strong HIPAA compliance while reducing the burden on vendors and hospital is to run vendor software in a secure sandboxed environment. However, this is not possible in a typical healthcare enterprise environment, where different software systems (often authored by different vendors) need to be able to communicate with each other to function. This raises an obvious question: Would it be possible for software from different vendors to operate in a shared computing environment?

Until recently, this was not possible because traditional environments are prone to catastrophic failures if a single application in a shared environment misbehaves. Essentially, the main innovation of blockchain computing systems is to provide a

[18] http://www.healthcareitnews.com/news/hipaa-data-breaches-climb-138-percent

solution to this technical obstacle, through the means of a fault tolerant consensus algorithm[19] and the use of smart contracts, a secure and heavily constrained environment for performing computation.

By using a BCS, vendors can access sensitive patient data and can directly communicate with software from other vendors, all within an environment that makes it technically impossible to violate patient privacy. Hence, these systems have the potential to improve HIPAA compliance, while simultaneously lowering the compliance burden for hospital IT staff and medical software vendors.

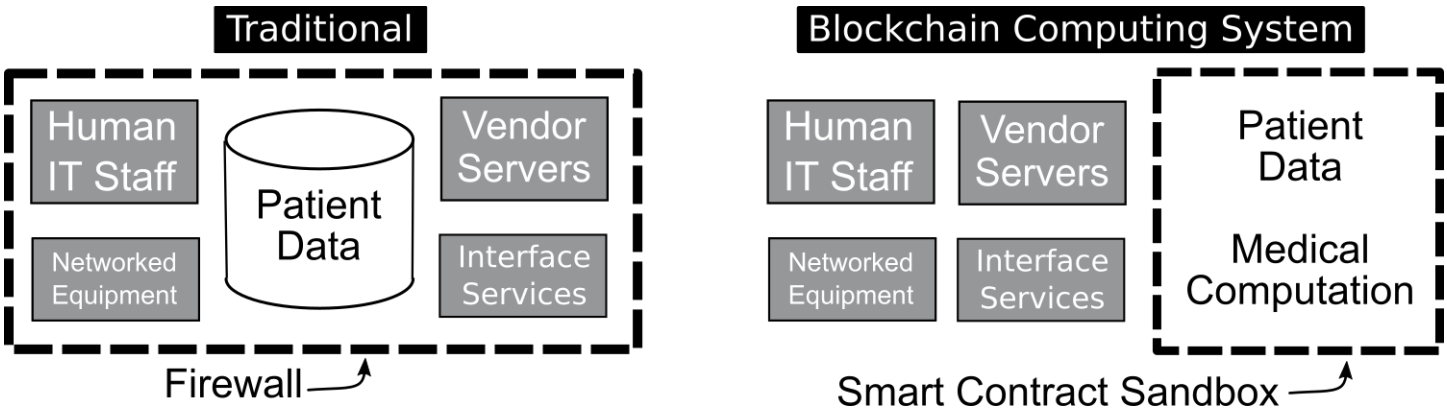

Figure 2: Traditional hospital systems protect patient data by placing a firewall around the network. Unfortunately, many entities must be given access rights inside of the firewall to allow the system to operate. A BCS, on the other hand, requires all code that needs PHI access to be in the form of heavily-constrained smart contracts. This greatly decreases the number of potential attack vectors in the system.

But a BCS can improve HIPAA compliance in other ways, as well. For instance, by providing a generic platform for securely hosting code provided by multiple vendors, the necessity for IT staff to have admin access to different hospital systems is greatly reduced, which additionally decreases the attack surface for hackers. Additionally, the ability for a blockchain system to efficiently implement single-direction information flow (as in the example of a vitals monitor that implements transaction signing) removes another potential vector for leaking sensitive patient data.

These enhancements to security come with the added boon of reducing waste; the BCS system saves much labor and time, thereby significantly reducing cost.

Poor Interoperability of Software Across Vendors

Another significant weakness of existing hospital software is that systems from different vendors usually do not offer a seamless, shared interface to hospital staff. This is true even though a large amount of the development cost in current medical applications involves developing interfaces between different systems.[20] Despite these expenses, healthcare staff are commonly frustrated by poor interoperability between applications.[21]

There are several reasons why such poor interoperability exists:

1. The need to provide frequent software updates directly conflicts with the goal of providing good interoperability, since each new software version may affect the communication protocol that another application relies on.
2. Communication protocols between different software components need to be agreed upon, often between many parties. This usually requires public committees to be established that usually take years to agree upon a new data format for communication.

[19] The byzantine fault tolerance algorithm originally described by Satoshi Nakamoto in "Bitcoin: A Peer-to-Peer Electronic Cash System" (https://bitcoin.org/bitcoin.pdf)
[20] http://www.modernhealthcare.com/article/20130727/MAGAZINE/307279974#
[21] http://www.healthcareitnews.com/news/frustrations-linger-around-electronic-health-records-and-user-centered-design

3. Competing vendors have few incentives to provide data in a manner that is useful to other vendors. Tactics such as citing intellectual property concerns and HIPAA compliance risks are too often used to thwart meaningful interoperability that could improve the patient experience.

Again, a BCS can directly address these issues:

1. Blockchain systems provide strong guarantees for data persistence, including for smart contracts. All smart contracts, once deployed on the network, are given a numerical smart contract address that is immutable. When a smart contract is updated, the old version of the smart contract will continue to be available at the existing address. This means a smart contract from one vendor that has a dependency on a smart contract from another vendor has a strong availability guarantee and can continue to use data provided by older version of other smart contract in perpetuity.
2. The persistence of smart contracts on the blockchain makes it feasible for vendors to communicate with each other through the means of *reference implementations* instead of committee-generated standards. This can often allow vendors to develop reliable communication between their systems in less time, since committee-generated standard require the coordination of many vendors to establish, whereas the development of a reference implementation can be initiated by a single vendor. Common standards still have a crucial role for larger-scale coordination between medical systems, but reference implementations can provide significant benefit for smaller-scale coordination challenges.
3. The open nature of smart contracts, and the inherent protections they provide for patient data, makes it more likely that vendors will share data between their systems.

Other Benefits over Traditional Medical IT Architectures

Beyond the benefits of blockchain computing systems for security and interoperability, there are other important benefits such systems can provide:

- **Reliability:** By their nature, blockchain systems are decentralized and lack a central point of failure. Because of this, blockchain systems are theoretically far less prone to catastrophic failures, compared to traditional systems that depend on centralized components.
- **Atomic software updates:** Medical applications implemented via smart contracts can be updated instantaneously, with an atomic transaction broadcast over the blockchain. In contrast, our experience is that typical downtime for updates/upgrades to most current-generation healthcare applications during upgrades is measured in hours.
- **Improvements in analytics and auditing capabilities:** Every modification to a blockchain system is recorded via an auditable transaction that is cryptographically signed by the originating entity. Additionally, the architecture described in this paper has all data stored on the blockchain, meaning all data in the hospital system has the same source of truth. This greatly eases analytics tasks that researchers need to run against patient records.

Conclusion

A blockchain computing system can be built with modest effort on top of current technology, primarily the ethereum technology stack. A BCS has the potential for many benefits when compared against traditional healthcare systems. Hopefully, over time, more people in the larger medical IT community will take notice of these benefits, leading to production systems built on this technology in the near future.

Even more advancements would be possible if the core technologies underlying blockchain computing systems were enhanced with features that are tailored for medical applications. These include:

- Linking blockchain encryption to electronic hardware keys that are given to providers.
- Developing better light client support[22] for ethereum for better support on mobile devices.
- Block signing algorithms that permit encrypted transactions (via an "order only" block signing methodology).
- Facilities for aggregating multiple blockchains into a single UI for providers that practice at multiple hospitals.
- Smart contracts for data aggregation could be deployed on a blockchain and combined with a batch processing API [23] to enable efficient research analytics, all while retaining strictly-enforced patient privacy guarantees.

However, further discussions of these advanced techniques are outside of the scope of this initial whitepaper.

[22] https://blog.ethereum.org/2015/01/10/light-clients-proof-stake/

[23] An API similar to that provided by Apache Spark: http://spark.apache.org/