Blockchain: An enabler for healthcare and life sciences transformation

F. Curbera
D. M. Dias
V. Simonyan
W. A. Yoon
A. Casella

Major trends in healthcare and life sciences (HCLS) include huge amounts of and longitudinal patient data, policies on a patient's rights to access and control their data, a move from fee-for-service to outcome-based contracts, and regulatory and privacy requirements. Blockchain, as a distributed transactional system of record, can provide underpinnings to enable these trends and enable transformative opportunities in HCLS by providing immutable data on a shared ledger, secure and authenticated transactions, and smart contracts that can represent rules that are executed with secure transactions. We describe HCLS use cases leveraging these facets of blockchain, including patient consent and health data exchange, outcome-based contracts, next-generation clinical trials, supply chain, and payments and claims. We then describe a blockchain-based architecture and platform for enabling these use cases. Finally, we outline a realization of this architecture in a case study and outline further research topics in this domain.

1 Introduction

The blockchain is an underlying technology of Bitcoin [1], which enables key functions for secure data access, data integrity, provenance, and transparency. These characteristics make blockchain an ideal technology to support major requirements for emerging healthcare and life sciences (HCLS) trends. A core underlying function required by many of these HCLS trends, as described further below, is the secure exchange of health data, with data integrity (i.e., guarantee that the data have not been changed or tampered with), provenance (i.e., the chain of custody in the data exchange), patient/owner consent (i.e., that the owner of the data, which may be the patient or another entity, provides explicit consent and/or contractual terms for access to the data), privacy (i.e., only authorized parties have access to the data), and audit trail (i.e., auditors have a trail of access and any changes to the data). Specific use cases, also described later, are based on data exchange between organizations participating in the healthcare value chain, e.g., between healthcare providers (e.g., physicians or hospitals) and payers (e.g., insurance companies), based on patient consent and authorization, and for specific purposes.

A variety of health data types are involved in these exchanges. While clinical data are often the focus in exchanges among healthcare providers, the faster growing data are genomics data, including those for normal cells and abnormal cells such as cancer cells; exogenous data including health and wellness data captured by mobiles; and real-time data such as ingestion of medication, blood sugar level, or other directly measurable parameters from connected devices. It is estimated [2] that clinical factors account for merely 10% of the determinants of individual health, while genomic factors and exogenous factors represent 30% and 60% of determinants of health, respectively.

In this paper, we focus on the use of blockchain for data exchange across the health ecosystem. In Section 2, we provide an overview of the blockchain conceptual architecture and the typical data flows. In Section 3, we describe use cases of blockchain for patient- and owner-mediated health data exchange, outcome-based contracts, clinical trial data exchange, exchanges for medical claims and payments, pharma supply chain, among others. Section 4 presents some technical challenges to blockchain, the detailed architecture of the blockchain-based system that supports all of the use cases described in Section 3, and how this architecture addresses these challenges. In Section 5,

Digital Object Identifier: 10.1147/JRD.2019.2913622

0018-8646/19 © 2019 IBM

we describe a case study for genomics data exchange and implementation leveraging blockchain. Finally, we summarize the paper and discuss future work and challenges.

2 Blockchain for HCLS overview

There are four primary facets of blockchain, particularly for its implementation under the Linux Foundation's Hyperledger project [3]. At the core is the shared ledger, which is a distributed transactional system of record. Transactions on the ledger are recorded in blocks, which are chained together with pointers and a hash of previous blocks of transaction; this is the reason for the name blockchain. The chaining of the blocks and hashes of the prior blocks, with replication across peers in the blockchain network, prevents updates to committed blocks, thus providing integrity of data in the blockchain. All of the nodes in a business network keep consistent copies of the blockchain records, which is done through the second facet of blockchain referred to as a consensus protocol that ensures agreement among the nodes (prior transactional models have referred to this as a commit protocol). Hyperledger allows plug-in consensus protocols, with the default being the "Practical Byzantine Fault Tolerance" (PBFT) protocol [4]. PBFT is an efficient protocol that can be used in "permissioned" networks, where all participants in the business network are identified and registered. (By contrast, Bitcoin uses the "Proof-of-Work" protocol [5], which requires "miners" in the network, which impacts scalability, but can be used in "permission-less" networks, with unidentified participants.) A third facet of blockchain is the built-in privacy and security protocols, which are described further in Section 4. The data on the blockchain are secured using public-private key protocols, and privacy is maintained by key-based access controls. Finally, Hyperledger provides "smart contracts," which are business logic invoked with transactions committed on the ledger. For example, smart contracts can be used to record the chain of custody of data access and exchange. This is similar to "stored procedures" in databases, but more general in the business logic associated with transactions. Section 4 also outlines the smart contracts for the use cases in Section 3.

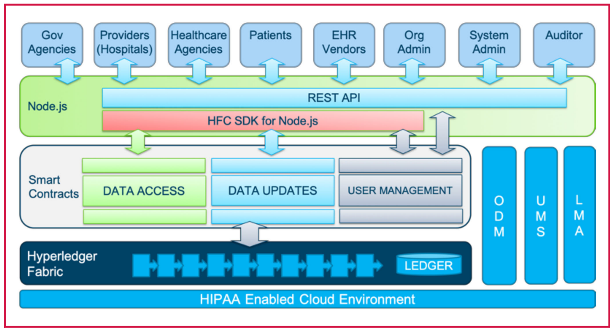


Figure 1

Blockchain software architecture for HCLS solutions.

Figure 1 illustrates the four-layer blockchain-based architecture in general, with some specifics for the use cases described in Section 3. Each of the Hyperledger nodes in the blockchain network needs to run in a Health Insurance Portability and Accountability Act (HIPAA)-enabled environment. They could also be run on a HIPAA-enabled cloud [6–8], as illustrated in Figure 1, which provides 59 controls required by HIPAA. Also shown in Figure 1 are Key Management Services (KMS), User Management Services (UMS), and Log Management Services (LMS), which are required for HIPAA enablement and provided by cloud environments such as the Watson Platform for Health [6].

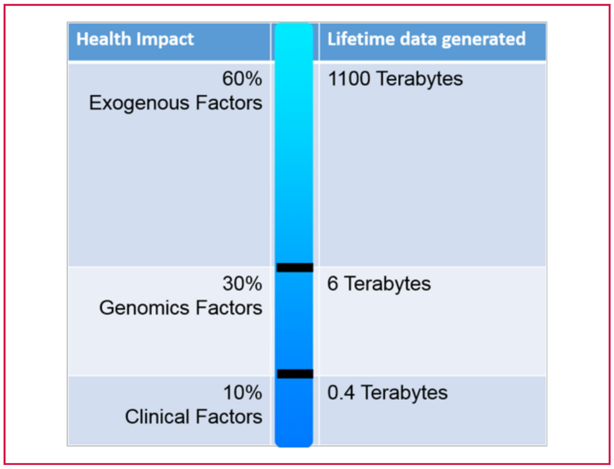
Above the core Hyperledger fabric is the Smart Contract layer, which implements core business logic associated with the use cases of Section 3 and can accommodate additional use cases. Illustrated in Figure 1, and described in detail in Section 4, are examples of reusable smart contracts, such as for patient consent management and contracts for access to data such as the case study in Section 5. The smart contracts are available as application programming interfaces (APIs) to the Application layer above this. To simplify integration with the front-end systems at the top of Figure 1, services are exposed as RESTful APIs (implemented in Node.js). These three layers can be made available as cloud services, with the front ends running in user environments, as shown in Figure 1.

3 Blockchain for HCLS use cases

In this section, we present several use cases across the HCLS industries, in which blockchain technology has the potential to disrupt and transform current practices. As outlined in Section 1, blockchain can provide high-value solutions when introduced to augment or re-engineer processes that are interorganizational by nature and benefit from additional trust between participants. Among those, we focus here on those related to healthcare data exchange, payments, clinical trials, and the pharmaceutical supply chain.

3.1 Blockchain for patient-mediated data exchange

The amount of healthcare data available in electronic format has dramatically increased in recent years and is expected to continue growing at a high pace. Nearly all electronic health records (EHRs) are universally available, while access to personal genomics data is more and more common due to the lower cost of genome sequencing and the recent focus on precision medicine treatments. In addition, data generated outside the healthcare system, sometimes referred to as "exogenous" healthcare data, are being collected in increasingly large amounts, including



Health Impact	Lifetime data generated
60% Exogenous Factors	1100 Terabytes
30% Genomics Factors	6 Terabytes
10% Clinical Factors	0.4 Terabytes

Figure 2

Health data generated in a person's lifetime.

data generated by consumer-grade wearable devices and data related to social determinants of health. The amounts and explanatory power of these different data sources were estimated in [2] and are summarized in **Figure 2**.

Figure 2 shows the amount of health data generated in a person's lifetime. The availability of data is driving the creation of new treatments and solutions, targeted at better managing the health of individuals and populations, fueled by the application of statistical data analysis, predictive methods, and risk analysis techniques. However, access to these data is seriously hampered by several factors, including regulatory requirements (HIPAA and HITECH in particular), interoperability and IT barriers, and lack of business incentives for data sharing. Regulations demand, on the other hand, that data be made available to patients on request and to third parties on availability of patient consent. Patient consent can thus provide the key to increased fluidity of healthcare data across the healthcare system and deliver data where it can be most useful.

Leveraging patient consent to make patient data available through the healthcare system requires a secure, globally accessible system of record where patient access authorization documents can be maintained, and where data holders and data requestors can verify patient consent for data exchange transactions. It should provide mechanisms for data use requestors to obtain patient consent to access their information, in very specific and granular terms, by specifying what types of records are being requested, from what date periods, and the purpose and scope of the data use, including the date when the authorization will expire. Patients should be able to approve and revoke consent at any time, and have a full record of which data were accessed at each time, by whom, and with what purpose. To this end, all data access and exchange activity should be recorded together by the organizations involved.

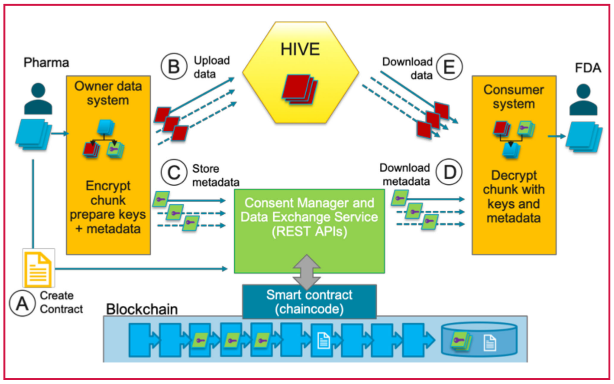

Pharma
B Upload data
HIVE
Download data E
FDA
Owner data system
Consumer system
Encrypt chunk prepare keys + metadata
C Store metadata
Download metadata D
Decrypt chunk with keys and metadata
Consent Manager and Data Exchange Service (REST APIs)
A Create Contract
Blockchain
Smart contract (chaincode)

Figure 3

Blockchain for owner-mediated data exchange.

Blockchain provides a way to capture and store documents and metadata that can be trusted by all participants in the data exchange process and, more importantly, by the patient whose data is being exchanged. The shared immutable ledger can keep a full record of consent requests, authorization, and data exchange. Providing transparency that supports patient visibility and control over the use of her data assures all participants of the integrity of the transaction.

3.2 Blockchain for owner-mediated health data exchange

A generalization of the patient-mediated health data exchange described above is for exchange of datasets, which may be owned by a third party, e.g., a sponsor of a study or clinical trial. This may, for example, include de-identified data from many patients.

Figure 3 illustrates this use case, which is described in more detail in Section 5. The user consent from the previous use case is generalized to a contract for access to the data. Blockchain is used to store the contract template (step A in the figure); the data are uploaded to the cloud data store (high-performance integrated virtual environment—HIVE, in step B), and metadata including pointers and hash of the data is stored in the blockchain (step C). Once the contract is negotiated between the data owner and data requester, the metadata of the dataset is released to the data requester (step D), who is now able to download and decrypt the data (step E). Blockchain is also used to track the chain of custody and access to the data. This method can be used to exchange big data, with the data split into chunks and distributed across repositories, and the metadata for the chunks stored in blockchain. This can enable exchange of genomics data, imaging data, and other emerging big data, across partners in the HCLS ecosystem.

3.3 Blockchain for value-based care models

While increased data sharing across the healthcare system can foster development and adoption of innovative care programs and increased efficiency in drug research, a major societal concern with today's healthcare system is the sustained increase in cost with no correlated improvement in patient outcomes [9]. For more than a decade, the U.S. Department of Health and Human Services (HHS) has been promoting new reimbursement models that focus on the quality of the care provided and moving away from fee-for-service payments. HHS targets are to link 90% of healthcare payments to value-based contracting by 2018. There are different modalities of value-based reimbursement programs, but they all aim to shift part of the risk associated with patient outcomes from payers to healthcare providers. Value-based contracts include, among others: "pay-for-performance" contracts, where adjustments to provider payment are made based on reported quality metrics; "bundled payments," where a fixed price is set for all the collection or services provided as part of a defined episode of care, such as joint replacements, and the providers involved share the resulting risk or rewards; and "outcome-based contracts," in which new treatments are reimbursed on condition of demonstrated improved patient outcomes.

These reimbursement models share the common characteristic of using data reported from different sources to make a determination of payment on which all parties need to agree. Third-party processors are often hired to establish the settlement of payments or to provide auditing services, but the work is largely a manual process of data collection and reconciliation that only allows settlement at fixed intervals. A platform for automating and streamlining of value-based contracts would reduce the cost of administering those contracts and allow "quasi-real-time" settlement (as opposed to quarterly or yearly settlement), thus freeing up valuable resources to participants and allowing more effective management of cash positions and risk. Such a platform would need to collect data from multiple sources and execute contract settlement logic against those data, in a way that results can be automatically trusted by all participants. Transparency and visibility into the data and settlement process are a critical requirement in order to maintain trust and to allow risk management in a transparent way; settlement logic can be executed in blockchain smart contracts, while ensuring that all the parties trust the distributed data on which it is based.

3.4 Blockchain for clinical trials

Clinical trials have multiple stages, from definition and approval of the protocol, execution and data collection, analysis, and auditing. Steps include creation of the study, obtaining patient consent, data gathering during treatment with Electronic Case Report Forms, adverse event tracking, monitoring, and review. Each of these steps needs data recording, exchange, and governance, for which blockchain is ideal, particularly since several different organizations are involved including pharma companies, patients, healthcare providers including investigators, and regulators. Blockchain can provide the underlying technology base, and the owner-mediated health data exchange technology can be used as the underlying mechanism for exchange data capture, tracking, and exchange.

3.5 Other use cases

The use cases described in this section are only a sample of HCLS use cases for blockchain. Other use cases include:

1) Accumulators for health-care deductibles: Often there are multiple payers providing coverage, such as health insurance companies and pharmacy plan providers, while a patient's deductible includes payments to multiple physicians, labs, hospitals, and other health service providers. Blockchain can be used across such organizations to coordinate deductibles paid by patients.
2) Preauthorization: These are primarily done by multiple Electronic Data Exchange (EDI) messages across providers, payers, and third parties. Blockchain-based preauthorization can provide transparency to all parties involved and streamline the process.
3) Claims: Again, this is currently done using age-old EDI transactions, which can be modernized with blockchain.
4) Payments: Given the complex payment contracts, particularly for value-based care, as outlined earlier, blockchain can provide the trust, transparency, tracking, and integrity, particularly for complex payment contracts.
5) Supply chain: Pharma supply chain provenance, traceability, and visibility can be transformed by blockchain.

4 Blockchain for HCLS architecture and design

In this section, we will discuss each layer of our reference architecture in depth, and how this design addresses some common technical challenges to Blockchain.

The HCLS blockchain use cases listed in Section 3 all conform to the conceptual architecture outlined by Figure 1 in Section 2. **Figure 4** illustrates a more detailed design of each layer as well as the functional component blocks within them.

At the bottom of Figure 4 is the Cloud Infrastructure layer. It provides a cloud environment compliant with the HIPAA such as the Watson Platform for Health [6].

Above the Cloud Infrastructure layer is the Hyperledger Fabric layer. This layer contains the Hyperledger Fabric framework and services that run on the HIPAA-enabled cloud environment. Services such as UMS, KMS, and LMS are integrated with Hyperledger Fabric for HIPAA

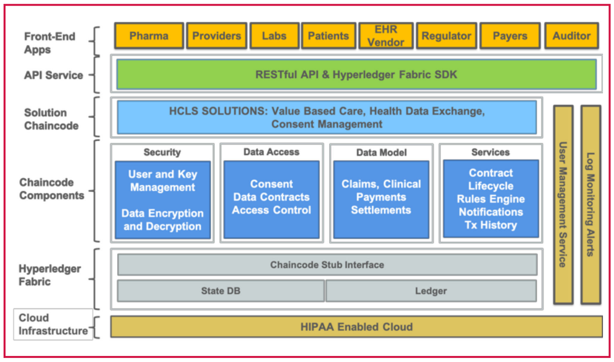


Figure 4

Blockchain for HCLS design and smart contract building blocks.

compliance. KMS is used in addition to Hyperledger's native key management system to generate Rivest–Shamir–Adleman (RSA) private and public key pairs (with the RSA 2048 algorithm) and AES 256-byte symmetric keys. UMS is used for managing user access control alongside Hyperledger's native user membership service. More details regarding integrated user access control can be found later in this section. LMS is used to comply with HIPAA's logging and monitoring requirements.

High-level component blocks of the Hyperledger Fabric layer are as follow. A Certificate Authority (CA) provides membership and certificate management. An Orderer or Ordering-service-node runs the communication services that implement a delivery guarantee, such as atomic or total order broadcast [10]. A Peer Node on the blockchain network commits transactions and maintains the state database (DB) and a copy of the ledger. A State DB is the current state of the blockchain network. It is usually implemented as a key/value store where keys are identifiers of assets in the blockchain and values are arbitrary blobs of data. A Ledger records the history of all transactions in the form of encrypted hash-chain blocks.

Access to state DB and ledger from the chaincode is done via the chaincode stub interface. Besides PUT and GET functions for writing and reading from the blockchain, the chaincode stub interface also provides several other interface functions for communicating with Hyperledger Fabric. However, the chaincode stub interface does not give full access to all of Fabric's internal functions. Only certain functions are exposed to the chaincode in order to maintain security and privacy control within the blockchain. In any case, chaincode can only access the Fabric through the chaincode stub interface functions.

Above the Hyperledger Fabric Layer is the Smart Contracts layer. This layer contains core business logic associated with the use cases described in Section 3. This layer consists of two smaller chaincode layers. The bottom layer represents the chaincode core components, and the top layer represents the solution chaincode. The chaincode core components are the building blocks of business logic. These core components are reusable modularized lower-level functions; each component refers to a core foundational functionality such as data access management, user access management, patient consent management, and error handling. On top of these core components, solution-level business logic and high-level chaincode functions can be implemented for all solutions described in Section 3. For example, the patient consent and health data exchange solution requires a set of high-level chaincode functions, namely *addConsent, validateConsents, uploadUserData, downloadUserData*, and *consentChangeHistory,* which can all be implemented on the solution chaincode layer using component blocks. Note that some of the core components are in fact wrappers for external solutions. For example, the core component *Rule Engine* is not an actual implementation of a rule engine, but instead it is a wrapper for a third-party rule engine like the IBM Operational Decision Manager [11].

Above the Smart Contract Layer is the API Service layer. The API Service layer provides RESTful APIs for front-end clients to access the business logic implemented in the Smart Contract layer. Although it is technically possible to expose APIs to lower-level chaincode core components, we discourage this approach and encourage exposing high-level solution chaincode functions as RESTful APIs since all lower-level operations are handled in the Chaincode layer. In this architecture, the API Service layer is implemented with the Node.js framework, and Hyperledger Fabric SDK (HFS) is used for communication with the chaincode. These communications are done via the gRPC protocol; HFC provides JavaScript wrapper functions for wrapping these lower-level communications [12]. Hyperledger also provides the HFS for Java [13]. These RESTful APIs make up the complete set of business logic and admin functions needed to implement solutions in the Smart Contract layer.

Above the API Service layer is the Front-End Applications layer, which consumes RESTful APIs provided by the API Service layer. Front-end applications can be written solely based on RESTful APIs. Although the HCLS blockchain platform provides a basic front-end application for a system administrator, all front-end applications are use-case-specific and can be implemented using a platform of their choosing.

In addition to architecture design, user access control is an important aspect of blockchain for HCLS. **Figures 5** and **6** illustrate how user management with UMS and KMS integration is implemented. As shown in Figure 5, the first step of user registration is creating RSA key pairs and AES symmetric keys in KMS. In step 2, the blockchain user registers with the CA of the blockchain network, and user

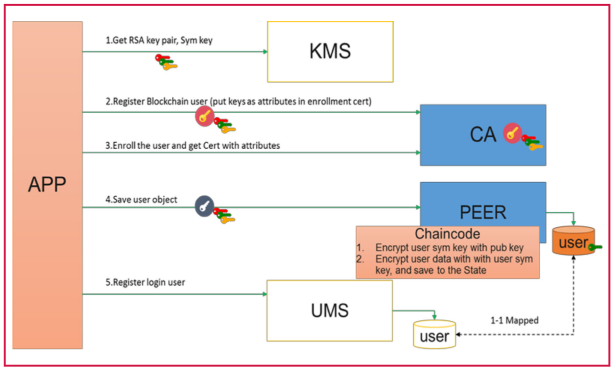

APP
1.Get RSA key pair, Sym key
KMS
2.Register Blockchain user (put keys as attributes in enrollment cert)
CA
3.Enroll the user and get Cert with attributes
4.Save user object
PEER
Chaincode
1. Encrypt user sym key with pub key
2. Encrypt user data with with user sym key, and save to the State
user
5.Register login user
UMS
user
1-1 Mapped

Figure 5

User registration process.

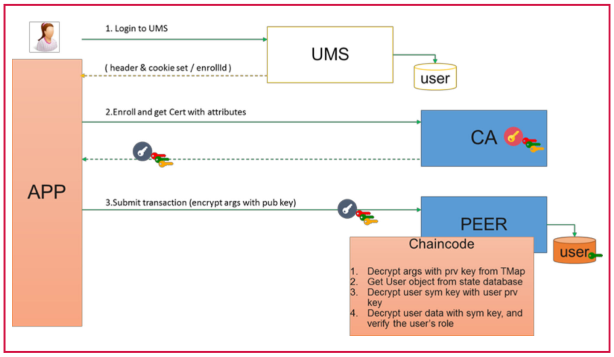

1. Login to UMS
UMS
user
(header & cookie set / enrollId)
2.Enroll and get Cert with attributes
CA
APP
3.Submit transaction (encrypt args with pub key)
PEER
Chaincode
1. Decrypt args with prv key from TMap
2. Get User object from state database
3. Decrypt user sym key with user prv key
4. Decrypt user data with sym key, and verify the user's role
user

Figure 6

User login and submitting transaction.

keys are stored as CA certificate attributes. Next, the user object is stored on the ledger. After this step is complete, user data are encrypted with the user's symmetric key which is in turn encrypted with the user's public key. As a result, stored data in state DB contain encrypted data and encrypted symmetric key of the user. The last step of user registration is to create a user in UMS. In UMS, users have a 1-to-1 mapping with the blockchain users created in step 2, and the mapping information is stored in UMS along with user ID, user password, user role, and any other user information.

Figure 6 illustrates the process of user login and submitting a transaction. In step 1, a user logs in to UMS using user ID and user password, and retrieves the mapping information pointing to the blockchain user. In step 2, using this mapping information, the user logs in to the CA of the network and retrieves his private key. In step 3, the user submits a transaction to the blockchain, and the keys are securely passed along with the transaction. The arguments of the transaction are encrypted with the user's public key. Note that the user's public key is available to all users.

In the chaincode, this process is validated by performing the following steps. First, decrypt the arguments of the transaction with the private key of the user. This will only succeed if the user's private key matches the user's public key. Second, obtain the encrypted user object and encrypted user symmetric key from the ledger. If successful, decrypt the user object with the symmetric key. Once the user object has been decrypted, the chaincode can validate user role type defined in the user object as well as access rights with the smart contract logic. By performing these steps, user identity is validated and user permissions can be properly granted according to user role type.

This section defined the shared architectural design and layers of an HCLS blockchain system and the flow of user management using UMS and KMS. Following this architecture and design, all use cases described in Section 3 and any other blockchain solutions for healthcare can be implemented.

4.1 Technical challenges

Privacy and security of data: Applying blockchain to solutions in the HCLS domain requires addressing data privacy concerns, as required by HIPAA regulations in the United States and similar ones in other countries. Hyperledger's native data encryption scheme is not sufficient since it does not allow defining fine-grain visibility rules per data element and per user. The architecture we describe here adds an additional layer of data encryption and restricts data access to authorized users only through the KMS, as described above. Each piece of data stored on the ledger is encrypted with a data key, and only authorized users are given access to the decryption key. When access is revoked, the decryption key will no longer be available to the user. Furthermore, Hyperledger Fabric supports the use of private channels where data can only be shared within members of the channel. Using this design, we ensure HIPAA compliance and the privacy and security of patient data.

Scalability: A fundamental design of blockchain is storing the same copy of data across the blockchain network to ledgers hosted on peer nodes. For n number of peer nodes in a network, the ledger is replicated n number of times. Due to this design, it can be expensive to store large amounts of data. However, our architecture solves this problem by giving the option to store only metadata acting as pointers to the actual data, which does not have to be stored on the ledger itself. Data integrity is ensured by storing a hash of the data within the ledger. As shown in Figure 3, actual data can be split into chunks and distributed across repositories, and only the metadata for each chunk needs to be stored in blockchain.

Speed: Transactions involving large amounts of data can impact the speed of the blockchain network. In addition, the distributed nature of the ledger and the consensus protocol add latency to every ledger transaction. Our latency tests have confirmed that most of the time each transaction takes is the communication overhead within Hyperledger Fabric itself; at this stage, speed is mostly limited by the performance of Fabric. However, through Hyperledger Fabric, our architecture supports custom configuration of endorsement policies so that a blockchain network administrator can choose how many endorsing peers must execute the transaction before a proposal response is sent to the Ordering Service. This allows for more flexibility and improves speed. It needs to be noted that the extra layer of encryption included in our architecture adds to the overall latency of chaincode calls, but it can be effectively managed through the use of hardware-assisted encryption.

4.2 Comparison with other blockchain for healthcare projects

The MedRec prototype is a blockchain record management system for EHR [14]. It incentivizes researchers and public health authorities to participate in the network as miners by giving them access to data as rewards. It is built on Ethereum and uses Proof of Work protocol. Identity is managed using public key cryptography and maps user identity to a person's Ethereum address. Data are stored on off-chain databases, and data access is governed by permissions stored on the blockchain. Compared to our architecture design, MedRec lacks the same level of data privacy; since Ethereum is a public blockchain, anyone can join the network and query the blockchain, which allows for attacks such as using frequency analysis to infer user identity. In addition, it does not encrypt data stored on off-chain databases, so anyone who gains access is exposed to sensitive EHR directly.

Medicalchain [15] is built on two different blockchains: Hyperledger for data access management, and Ethereum to facilitate the exchange of data using ERC20 tokens. Identity is managed using Civic, an Ethereum-based authentication system. Medicalchain allows users to give conditional access to medical personnel. Data are stored in a data store and encrypted using a combination of symmetric and public key cryptography. When a user's access to data is removed, the data are re-encrypted with a new symmetric key, which is then encrypted with all authorized users' public keys. If there are many authorized users, this could be a very expensive operation and lead to scalability issues. In our architecture, when access to removed, we simply remove the authorization of that user to access the data and update the key graph accordingly.

Embleema [16] is built on Ethereum and uses proof of authority protocol. It stores de-identified health records both on and off the blockchain; text-based protected health information (PHI) collected during encounters are stored on-chain; images, scans, and genomics data are stored off-chain, with hashes and pointers to data stored on-chain. It facilitates data transfer through ERC20 tokens, and data owners must pay miners to store data. In our architecture, we support storing data entirely off-chain for improved scalability with store only hash and pointers on-chain.

5 Case study: Health Data Exchange Framework (HDEF)

One of the most critical stages of healthcare research and development is the analysis of biomedical and health research data by regulatory organizations such as the Food and Drug Administration (FDA). With scientific validity, treatment efficacy, and patient safety in mind, operations frequently involve a large amount of sponsor-submitted pre-market, clinical, and post-market data. The current state of the business is somewhat rudimentary and involves paper printouts, PDF documents, mailed hard drives, and exception-based *ad hoc* solutions. The immense variety of the different types, Peta-scale size, and complex structured and unstructured nature of the data, combined with privacy and security needs of proprietary information, demands a specific solution that is the subject of this section.

To address this need, we implemented a distributed technology platform HDEF using blockchain technology as a backbone for e-contract transactions, and HIVE technology [10] for big-data transactions in a regulatory compliant environment. A few key functions of this architecture that consists of a delocalized set of HDEF nodes include:

1) data upload by owner;
2) e-contract negotiations;
3) data download by requestor.

5.1 Data upload by owner

After registration and signing into the secure HIVE portal (supported by FDA), the owner of the data can initiate an upload process of virtually unlimited amounts of data through a head-node and initiate the following flow of actions:

1) The data are striped on the fly into smaller pieces by the HDEF engine running on a head-node. While striping, the datatype-aware engine ensures no single patient data or no single genomic sequence of a single patient ends up in a single stripe.
2) Each slice is encrypted using a randomly chosen encryption algorithm into an unidentifiable blob of data.
3) The encrypted blob is sent into a randomly chosen HDEF node that is unaware of its nature and character; a unique identification number is sent back to the transmitting/striping node.
4) The blob is deleted from the head-node before the next stripe can arrive from the owner's computer.
5) All stripes are processed the same way through steps 1–4, and locations of destination blobs, unique

identifiers of blobs, encryption algorithms, and decryption keys are accumulated into Stripe Engineering Metadata (SEM).

6) SEM is encrypted additionally and transmitted to blockchain using the above-mentioned API.

It is important to note that actual health data does not end up on blockchain; only the striping information containing back pointers and decryption keys are transmitted. Given the size of the genomic and other biomedical research data—transferring the entire content would be taxing on the blockchain network and renders the entire system dysfunctional. Importantly, at no point during the upload process is an entire dataset available on a single data node in the system, even the head-node.

5.2 E-contract negotiations

The owner or a requestor of data can start forming an electronic document of e-contract using HIVE record engine that supports hierarchical object-oriented data typing and instantiation. The default contract structure includes a set of fields allowing for granular Data Use Agreement (DUA). The base contract includes permission controls for health data fields, clinical domain elements, patient subsets, genomic profile filters, and other relevant information. Permissions can be time-limited or indefinite, allowing or denying a particular use pattern. There are also controls in the e-contract permitting data brokerage, derived data production, and data analysis using predefined types of vetted analysis [11]. The hierarchical nature of HIVE data-typing engine also allows expansion and design of new contract types by inheriting from basic forms of contracts already provided in the system.

Once the contract is formed, it can be submitted to the other party using provided API and appropriately built visual web interfaces. The parties can modify the values of the contract, add new ones, and negotiate terms until an agreement is reached. At this point, the data requestor is given an opportunity to sign the contract electronically, and by doing that submits a legal formal request to obtain the data for preagreed conditions. The requestor can then specify the number of uses under the signed DUA contract.

Only after the e-contract is signed and downloads are permitted, the legally binding contract that contains the SEM information becomes visible to the data requestor. An HIVE head-node (to which the requestor is connected during the contract negotiations) can use the SEM information to proceed to the download stage.

5.3 Data download by requestor

The download process is symmetric to the upload process in reverse chronological order. SEM information is decrypted by an HIVE head-node, and one by one the stripe pieces are received from delocalized remote nodes as blobs using the unique identifiers in SEM. The head-node receives a single slice at a time, decrypts it, recomposes the data, and sends it to the requestor. After the download of a chunk has completed, and while the connection to the user's computer is still "alive," the head-node deletes the blob from its internal cache, and only then proceeds to receive the next slice for further processing. It is important to notice that at no point in time the entire content is on a single data-node or even a head-node. The only location where the content is available entirely is the requestor's computer.

5.4 Pilot

A pilot study was conducted using HDEF engine to test the feasibility and validate the robustness of the designed processes. A genomic file of 200 GB size that is equivalent to a 200-million-page document containing Next Generation Sequences information has been uploaded by the data owner into HDEF. The data were delocalized into 20 × 10-GB stripe blobs in between eight distributed remote computer nodes across the United States. The contract was composed by an FDA scientist and submitted to the data owner; negotiations were "played out" using a complex pattern of data access, and DUA was signed by both parties. The final data were downloaded at the FDA, verified, and validated by checksum and Linux diff utility. The entire process took about 10 minutes. Compare this to the similar operation using conventional protocols: It would take ~2 months of paper, PDF documents, hard drive mailing, error-prone copy/paste transactions, and lengthy human DUA contract negotiations through in-person meetings and remote conferences.

6 Conclusion

In this paper, we presented blockchain as a differentiating technology to enable transformative use cases in HCLS. These use cases leverage blockchain for secure data exchange, with integrity, provenance, patient or owner consent, privacy, audit trails, and transparency. In addition, blockchain smart contracts can be used to process the data exchanged to enable functions such as outcome-based contracts and bundled payments. We also presented a shared architecture pattern that supports these use cases, and a system design and prototype that realize this architecture. We then presented a case study for an HDEF and an implementation of this case study.

Some of the ongoing work and challenges include: multiblockchain environments, which can be for scalability, performance, and isolation; interoperability across multiple underlying blockchain infrastructure; standardization of key blockchain services for HCLS; and sharing of the blockchain platform with other industry solutions, particularly with shared building blocks across multiple industries.

Acknowledgment

We would like to thank many of our colleagues in IBM Watson Health and the FDA who have collaborated with us on this work. S. Ebadollahi at IBM and S. Khozin initiated the collaboration across IBM and the FDA, which is a key aspect of this work. Other key collaborators who shared their technical expertise include M. Faradzheva, A. Golikov, and A. Myers at the FDA, and J. McAfoose, D. Sadler, and A. Chow at IBM.

References

1. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2018. [Online]. Available: http://bitcoin.org/bitcoin.pdf.
2. J. M. McGinnis, P. Williams-Russo, and J. R. Knickman, “The case for more active policy attention to health promotion,” *Health Affairs*, vol. 21, no. 2, pp. 78–93, 2002.
3. Hyperledger, Architecture Volume 1. 2017. [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2017/08/HyperLedger_Arch_WG_Paper_1_Consensus.pdf.
4. M. Castro and B. Liskov, “Practical byzantine fault tolerance,” in *Proc. 3rd Symp. Operating Syst. Des. Implementation*, Feb. 1999, pp. 173–186.
5. M. Jakobsson and A. Juels, “Proofs of work and bread pudding protocols,” in *Proc. IFIP TC6/TC11 Joint Working Conf. Secure Inf. Netw., Commun. Mulitmedia Security*, 1999, pp. 258–272.
6. A. Mohindra, D. Dias, and H. Lei, “Health cloud: An enabler for healthcare transformation,” in *Proc. IEEE Int. Conf. Serv. Comput.,* 2016, pp. 451–458.
7. A. Mohindra and D. Dias, “Industry cloud: A driver for enterprise transformation,” in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jul. 2015, pp. 1125–1130.
8. P. F. Edemekong and M. J. Haydel, “The Health Insurance Portability and Accountability Act (HIPAA),” 1996. [Online]. Available: http://www.hhs.gov/ocr/privacy.
9. M. B. Rothberg, J. Cohen, P. Lindenauer, et al., “Little evidence of correlation between growth in health care spending NND reduced mortality,” *Health Affairs*, vol. 29, no. 8, pp. 1523–1531, 2010.
10. V. Simonyan, K. Chumakov, H. Dingerdissen, et al., “High-performance integrated virtual environment (HIVE): A robust infrastructure for next-generation sequence data analysis,” Database (Oxford), Mar. 17, 2016.
11. V. Simonyan, J. Goecks, and R. Mazumder, “Biocompute objects-a step towards evaluation and validation of biomedical scientific computations,” *PDA J. Pharm. Sci. Technol.*, vol. 71, pp.136–146, Mar./Apr. 2017.
12. Hyperledger Fabric Official Documentation, “Securing communication with transport layer security,” 2018. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.3/enable_tls.html.
13. Hyperledger Fabric Official Documentation, “What’s new in v1.3,” 2018. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.3/whatsnew.html.
14. A. Ekblaw, A. Azaria, J. D. Halamka, et al., “A case study for blockchain in healthcare: ‘MedRec’ prototype for electronic health records and medical research data,” MIT Media Lab., Beth Israel Deaconess Med. Center, Boston, MA, USA, Aug. 2016. [Online]. Available: https://www.healthit.gov/sites/default/files/5-56-onc_blockchainchallenge_mitwhitepaper.pdf.
15. Medicalchain Whitepaper 2.1, 2018. [Online]. Available: https://medicalchain.com/Medicalchain-Whitepaper-EN.pdf.
16. Embleema Blockchain Network V.2, Sep. 2018. [Online]. Available: http://whitepaper.embleema.com.

Received June 29, 2018; accepted for publication January 25, 2019

Francisco Curbera *IBM Corporation, Yorktown Heights, NY 10598 USA (curbera@us.ibm.com).* Dr. Curbera received a B.S. degree in physics from Universidad Complutense in Madrid, Madrid, Spain, and a Ph.D. degree in computer science from Columbia University, New York, NY, USA, in 2001. Between 1996 and 2015, he worked as a Research Staff Member and a Senior Research Manager with IBM Research. He later served as the Director of Technology Innovation with IBM Watson Health, and is currently the Director of Innovation and Technology Incubation with IBM, where he leads the development of blockchain-based solutions in the healthcare and life sciences industries. He has authored more than 30 academic publications in computer science.

Daniel M. Dias *IBM Corporation, Yorktown Heights, NY 10598 USA (Daniel.Dias@ibm.com).* Dr. Dias received both an M.S. degree and a Ph.D. degree in electrical engineering both from Rice University, Houston, TX, USA. For more than three decades, he has led pioneering research in scalable commercial computing systems, and he has been the Director of research labs and teams on three continents. He has served in leadership roles with IBM Research, including the Director of cloud innovation technologies with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, a founding Director of IBM Research–Brazil, and the Director of the IBM India Research Lab, among others. In 2015, he joined IBM Watson Health as a VP of Development, where he was responsible for the design and development of the Watson Health Cloud and insight solutions on this platform. He currently leads special projects in IBM Watson Health Innovations. He is a co-inventor on more than 50 issued U.S. patents, and has coauthored more than 90 refereed publications. He is an emeritus IBM Distinguished Engineer and a Fellow of the IEEE.

Vahan Simonyan *George Washington University, Washington, DC 20052 USA (vahansim@gmail.com).* Dr. Simonyan received a Ph.D. degree in physics and mathematics from Moscow State University, Moscow, Russia, in 1993. He was a Postdoctoral Fellow of nanotechnology with the University of Pittsburgh and went on to serve at the Food and Drug Administration as a Lead Scientist of HIVE and an R&D Director of Bioinformatics. He is an Adjunct Professor with the George Washington University, where he teaches biomedical big data informatics and biostatistics. He has authored more than 60 publications in scientific journals and conferences proceedings in several disciplines including genetics, bioinformatics, physics, and chemistry, among others.

Woong A. Yoon *IBM Corporation, Cambridge, MA 02142 USA (wayoon@us.ibm.com).* Mr. Yoon received both a B.S. degree and an M.S. degree in computer science from Cornell University, Ithaca, NY, USA, in 1993 and 2000, respectively. He served as the Directore de Informatica with Samsung Electronics Portugal, and was a Founder and CTO of Amos524, where he led the development of copyright protection algorithms. He is currently a Chief Architect of Blockchain Solutions for Healthcare and Life Sciences with IBM, where he leads design and development of various healthcare solutions on Hyperledger blockchain network. He is a Committee Member of the Association of Science & Technology Information, South Korea, and a Member of the Korea Institute of Science and Technology Information.

Alex Casella *IBM Corporation, Cambridge, MA 02142 USA (Alex.Casella@ibm.com).* Mr. Casella received a B.S. degree in computer science from Boston University, Boston, MA, USA, in 2016. He previously worked with the Cyber Defenders program with Lawrence Livermore National Laboratory and currently works as a Staff Software Engineer with IBM Watson. He was a recipient of the IBM Eminence and Excellence Award in 2017.