# Making Case for Using RAFT in Healthcare Through Hyperledger Fabric

Anastasios Alexandridis
*ECE. McGill University*
Montreal, Canada
anastasios.alexandridis@mail.mcgill.ca

Ghassan Al-Sumaidaee
*ECE. McGill University*
Montreal, Canada
ghassan.al-sumaidaee@mail.mcgill.ca

Rami Alkhudary
*LARGEPA. Université Paris II Panthéon-Assas*
Paris, France
rami.alkhudary@u-paris2.fr

Zeljko Zilic
*ECE. McGill University*
Montreal, Canada
zeljko.zilic@mcgill.ca

***Abstract*—Blockchain technology is enabled by consensus algorithms to manage the relationships among several economic or business operators without human intervention. With the help of consensus algorithms, distributed systems can reliably reach agreement even if part of the system is faulty. Blockchain yields many benefits, among others, traceability, transparency, and security. We consider using the RAFT consensus algorithm to achieve robust and scalable decentralized applications, with focus on healthcare. We propose a stylized healthcare network, enabled by RAFT and built upon Hyperledger Fabric to showcase the use of RAFT in healthcare blockchain. However, RAFT is by no means limited to healthcare record systems, and can be applied to any other record system and value chain. Our paper offers several insights to those working in value chains and information management-related fields. In addition, we end our study with some future research avenues that may inspire managers and scholars to build or refine new decentralized systems in healthcare and other related fields.**

***Index Terms*—blockchain, RAFT, consensus algorithm, Hyperledger Fabric, value chain, healthcare**

## I. Introduction

The RAFT consensus algorithm [1] produces agreement on the log of activities on the replicated state machine. While RAFT can be applied to numerous applications, we are interested in its application in blockchain. Blockchain arose as the peer-to-peer electronic cash system Bitcoin proposed by Nakamoto [2] (we still do not know the real identity). Bitcoin has solved the problem of double spending on the internet, allowing a value exchange of digital assets or financial transactions without intermediaries or trusted third parties' intervention. The underlying blockchain technology relies on three relatively old technologies: distributed ledger technologies, cryptography, and consensus algorithms.

Blockchain can be defined as a distributed ledger or decentralized database of digital records chronologically organized and immutably registered following a precise consensus mechanism [3]–[5]. Consensus is necessary in decentralized databases with no trusted third parties to ensure reliable data. Blockchain operation can be better understood by explaining the main features of the Nakamoto model (public blockchain). First, each blockchain network must have a consensus mechanism, often translated into the mining process, to verify, link, and securely register transactions or data records by cryptographic proofs. Second, in theory, the network is fully decentralized since each user can have an identical copy of the blockchain ledger, maintain its integrity, and play an essential role in the mining process. Third, transactions or digital records become traceable, transparent, and immutable when registered, although users are anonymous in the public blockchain network [3], [6].

Nakamoto's proposition was mainly limited to cryptocurrencies and distributed exchange systems. Many years later, research on blockchain in record systems such as healthcare records extended in all directions [7], particularly in the grey literature. Although the public blockchain model (Nakamoto model) was proved to add value in a few healthcare applications [8]–[10], the current mainstream is using the private blockchain model [11]–[15]. The private blockchain model is different in that some economic or business operators (I) decide to work together building a limited access (closed) network, and (II) define the mechanism of the consensus that has to be reached to validate transactions as there is no need for mining [16]–[18].

PAXOS [19] is one of the very first attempts to achieve scalable and fault-tolerant consensus between multiple distributed systems. This mechanism is based on two principles by which consensus can be achieved across distributed systems. The first principle is the existence of a node that proposes a certain value so that all other nodes in the network adopt that certain value, called the *proposer* node. The proposer should associate this value with a unique number, so that the other nodes will commit to this value if they have never seen a number higher than this number before. The second principle is that there is a type of nodes making a decision about the value sent by the proposer, a so called *acceptor*. The acceptors either confirm the value and send the approval back to the proposer, or they partially accept it by guaranteeing that none of the other values received from different proposers are accepted, or they reject the value altogether and inform the

978-1-6654-3902-2/21/$31.00 ©2021 IEEE

proposer of this rejection. However, PAXOS was not as easy to put into practice for many reasons, including the lack of tools for handling physical failures in distributed manner. It is hard, for instance, to treat a plethora of disk corruption mechanisms under data replications among multiple actors, as it result in an unknown number of *corner cases* and with uncertain correctness provability, especially when malicious actions enter the play. These difficulties in understanding and implementation have cast a shadow on applying PAXOS in the real world, so extra effort was required to achieve real consensus. For this reason, RAFT [1] was proposed as an alternative to the functions of PAXOS, but one that is much easier to implement and understand. In a nutshell, RAFT is a leader-based mechanism, which means that data can only flow from the leader node to the other server nodes in the network. Consensus in this mechanism can be achieved mainly through three processes: immediate election of a network leader in case of failure of the current leader, replication of the log entry across all nodes, and permanent safety of the committed log entry to persistent storage. More details on RAFT consensus are provided in Section II after we introduce our application domain.

Blockchain yields many benefits in healthcare and other related fields handling large amounts of data sets, among others, traceability [20]–[22], transparency [23]–[25], and security [26]–[28]. The literature explains many consensus algorithms that can be used in decentralized healthcare applications, which are the focus of this paper, for public and private blockchain systems. For example, Yazdinejad et al. [8] proposed a blockchain configuration for a decentralized hospital network enabled by Proof-of-Work. Similarly, Yang et al. [29] designed a medical data sharing system based on Proof-of-Stake. Wang et al. [30] proposed an intelligent healthcare supply chain based on Delegated-Proof-of-Stake. Zhu et al. [12] introduced a consortium blockchain system using Proof-of-Authority. Zghaibeh et al. [11] proposed a multi-layer and intelligent health management system based on Practical-Byzantine-Fault-Tolerance. To end with, Wang et al. [31], [32] proposed a blockchain-based eHealthcare system interoperating with WBANs, with a multistage approach to increase performance and throughput, that proposes a leader and follower node configuration of a Crash-Fault-Tolerance implementation that uses Apache Kafka.

In trying to apply RAFT to blockchain, the literature falls short in explaining how the RAFT consensus (equivalent to PAXOS in fault-tolerance and performance) adds value to decentralized healthcare applications. In response, we propose a stylized healthcare network enabled by RAFT using Hyperledger Fabric as a proof of concept of its applicability. Our paper is hoped to offer insights to providing value chains in information management-related fields. In addition, we end our study with some future research avenues that may inspire managers and scholars to build or refine new decentralized systems in healthcare and other related fields.

## II. RAFT Consensus

With the recent increase in data breaches, it is becoming increasingly urgent to seriously consider a protection technique that ensures the security of data and guarantees seamless access when needed. Replication of data on distributed machines has been proposed in the literature as a solution for this dilemma. However, whenever the term ”distributed systems” is mentioned, the first question that arises is how these systems achieve consensus on the data state. Even in the case that consensus is reached, the question is how can it be guaranteed that all nodes are updated synchronously. These requirements are very urgent for a fault-tolerant distributed system that can function despite the failure of some nodes. Replicated And Fault-Tolerant (RAFT) [1] is a state machine replication protocol proposed to meet these requirements and solve the problem of inconsistency of distributed databases [33].

Nodes implementing RAFT can take one of three roles [34]: Leader, Candidate, and Follower. The leader node is responsible for interacting with clients on the network. Any client’s request received by the leader is directly forwarded to the followers for confirmation. The followers are responsible for acknowledging the requests and sending them back to the leader node. When a client attempts to contact the followers directly, they forward the request to the leader node. All nodes are initiated as follower nodes in their very first inception. The network should have only one leader node at a time. If this leader node fails for any reason, it paves the way for any follower node to become a leader node. Fig. 1 shows the transitions of the node states.

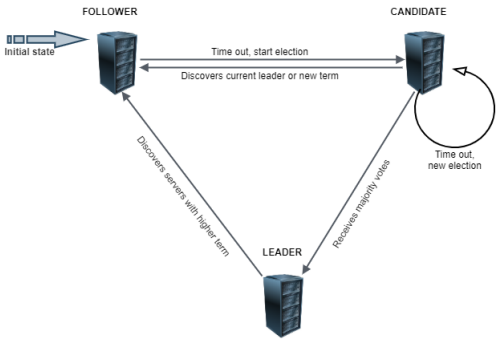


Fig. 1. Transitions of the node states as suggested in the RAFT paper.

In RAFT, time is divided into periods of arbitrary length, numbered consecutively. The election process to select a leader usually begins at the start of each period and ends the moment a winning node is elected [35]. However, the election can also be conducted if no leader node has appeared in a given period. The nodes in RAFT communicate through Remote Procedure Calls (RPCs), and consensus can be reached through two types of RPCs: RequestVote and AppendEntries. The leader node periodically sends a message to all followers to maintain its authority. There are two types of AppendEntries messages that the leader node can send to the followers; one is used to replicate the log entries when a new request is received

from the client. The other one does not contain any log entries and is only used to inform the followers about the continued validity of the leader node. However, followers have the right to change their status to candidate once a certain amount of time has passed without hearing from the leader.

Once a follower does this, it starts voting for itself and sends a RequestVote to all other nodes. Then, three scenarios can occur. The candidate node could get the most votes from the nodes and then declare itself the winner of the leader position in the network. The candidate node does not win this contest because another node has declared itself the leader. Last but not least, the time allotted for the election has elapsed without a winner. In this case, the call for the election is restarted. After the winning node is declared, it sends a message to all nodes to inform them of its authority and block the way for a new election for the rest of the term. As mentioned earlier, the new leader then begins processing the requests that come in from the clients. Whenever a new request arrives, the leader sends an AppendEntries message to the followers and waits for their confirmation that this request has been successfully replicated. If the followers do not respond appropriately, the leader sends RPC messages until the log is updated on all follower nodes.

The fork problem is prevented in the current Hyperledger Fabric enabled by RAFT since all network members are well authenticated. Thus, computationally complex consensus mechanisms (mining) are not necessary. Moreover, RAFT is efficient and implementable in private value chain networks [36], [37].

## III. Stylized healthcare network built upon Hyperledger Fabric

The example of a hospital network can be used to present our stylized healthcare configuration. A hospital or group of hospitals will typically maintain some kind of healthcare network, which facilitates communication between employees and machines, as well as a database that includes, but is not limited to, patient data such as examinations, prescriptions, or other information. It is critical that all this information exchange and storage is done in a secure and private fashion, both inside the hospitals as well as other external organizations such as laboratories, pharmacies, insurance companies, or other medical services. Other organizations are not in the scope of this work, and further work would need to examine their participation in the healthcare network.

In order to make use of blockchain, the health network can be designed as a blockchain network. As mentioned, a group of hospitals essentially has a secure communication network, and a database where data storage takes place. Different entities can interact securely with the database. Every hospital has an unspecified number of end users; employees, such as doctors, and machines, such as biomedical sensors, that can interact with each other and the database. The hospitals can also interact with each other. This is necessary in case patient data needs to be transferred to a different unit or even hospital.

Every hospital has back-end servers, which function as the blockchain database, or ledger. These servers are a core part of the blockchain network, and are hosting blockchain peer nodes who maintain the ledger, and ordering nodes. The servers can be physically separated into multiple machines, such that if there is failure in one machine, only one node will be compromised. Further redundancy can also be achieved on a hardware level; for instance the disk of a peer node which maintains a copy of the blockchain ledger can be redundantly mirrored, ensuring the peer node can continue working even if there is damage on the physical hardware. All the hospital back-end servers are connected together, and part of the same blockchain network. This can achieve data redundancy and replication, in essence producing a distributed ledger both in terms of machines, but also from a geographic point of view, and providing shared data access to all the hospitals.

The end users that interact with the network are the front-end, and can be viewed as clients of the system. Through the use of a client application external to the blockchain, the end users can interact with the blockchain and execute smart contracts on the peers. For example, if a doctor needs to insert some information about a patient in the ledger, they would use an application to run a smart contract on the peer, and provide the information as a parameter. From the doctor's point of view, the information is stored in the ledger as if it were a traditional database system, however the process is a little more nuanced than that. The client application will submit a transaction proposal to the blockchain network, which needs to be endorsed by a predefined number of peers, then ordered into a block by the ordering nodes, and then validated, before it is finalized in the ledger. During this process, the peers and ordering nodes check against policy criteria, validate that endorsements came from appropriate entities, and check the version of the ledger to ensure data integrity is not compromised, by repeatedly signing, verifying, and authenticating payloads as they pass through the network. All this process is abstracted from the end user.

Fig. 2 illustrates the process of querying or updating the ledger. When a doctor needs to retrieve or insert information in a patient health record, they can use application (A) to do so. A will connect to peer 1 (P1) and execute the appropriate section found in smart contract (S1) by submitting a transaction proposal as discussed above. It is worth mentioning that for a simple ledger query and information retrieval, P1 will endorse the transaction proposal and return the information from ledger (L1) to A in the form of a proposal response. No transaction will be committed to L1.

However, if a ledger update is needed, there are additional steps as discussed earlier. After being endorsed and a response is sent to the application, the transaction goes to orderer 1 (O1) for ordering, which will order this and other transactions from other applications into a block, that is then sent to all peers in the network in order to be added to the ledger. Fig. 3 goes into more detail about the transaction flow. Here, application A1 has received endorsement (E1) for its transaction (T1), and forwards it to O1 which orders it into a block. O1 has received more transactions, such as T2 from application A2, which in fact arrived earlier. Once a block is formed, it is sent to all

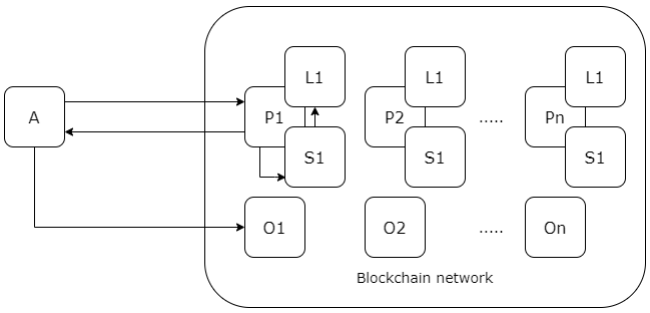


Fig. 2. Querying or updating the ledger.

peers in the network as shown in Fig. 4, which will validate it before appending it to the ledger, finalizing the process.

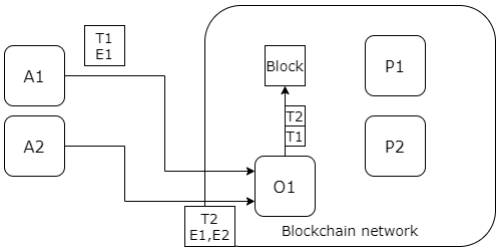


Fig. 3. Transactions are sent for ordering.

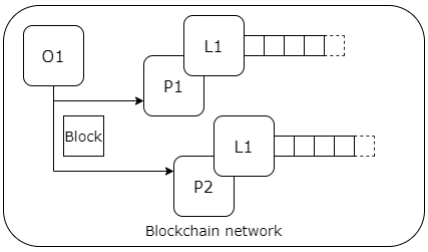


Fig. 4. A block is appended to the replicated ledger.

For all intents and purposes, necessary components of the system such as the Membership Service Provider (MSP) and Certificate Authority (CA) are abstracted here, however they are very necessary. Authentication of the different actors in the network constantly takes place, and is required to determine the exact permission each one has over different resources and information. For the purpose of this stylized healthcare network, an example of a group of five hospitals can be considered. These hospitals can be viewed as one organization, for example an organization named ”hospitals of a city”. Each hospital can have two servers, with one server being a peer node, and another server being an orderer node. All nodes are members of one channel. The peer nodes have smart contracts, also called chaincode, installed on them. As mentioned earlier, an end user with the right permissions that come from their identity can execute smart contracts. More particularly, a doctor will have permission over reading and writing data concerning their own patients, whereas a hospital supervisor may have read permission to all patient data. A system administrator will have full access. Access control through identity verification can be used to ensure that depending on the role of a system user, the appropriate permission is allowed.

In practice, the transaction flow is as follows. In the start, an end user, such as a doctor, needs to have a certificate issued by the CA in order to authenticate in the network. The doctor will use an application to initiate a database, or ledger, update, for example updating a patient’s health record with a recent test results, which will be sent to the peer node found in the hospital. It should be mentioned that the application the doctor is using could communicate with another peer located remotely, and it is not necessary to contact the local peer, or only one peer. The smart contract installed in the peer will, among others, contain a function for the doctor to submit the most recent patient test results. Additionally, an endorsement policy in the smart contract will specify how many peers are needed, in this case it can be assumed that one is sufficient.

The ledger update will initiate by the application initiating a transaction. This update will trigger the function in the smart contract that updates the patient test results. It will also contain certain parameters, for instance who the patient is and the actual test results. The certificate of the doctor is also used in this stage to sign the proposal.

The proposal will then go to the peer which will endorse it by verifying the signature, as well as that the proposal is unique and correctly formatted. This can ensure that it comes from a proper entity and not, for example, an attacker communicating with the network or a malicious application not working properly. Signature verification is done using the CA and MSP mentioned above. In short, the former ensures the certificate is valid and verifies the identity while the latter ensures that the particular identity has the right permissions, however, this is outside of the scope of this work. The endorsing peer will then execute the transaction without yet updating the ledger with the new patient test results, update the transaction with the results, sign it, and send it back to the application as a proposal response. The application will then verify the endorsing peer signature and correctness of the response, as well as compare and merge responses together in the case of multiple endorsing peers, which is not the case here. As discussed earlier, if the doctor was simply reading the ledger, the transaction would not be sent for ordering or committed to the ledger, however the patient’s recent test results need to be inserted in the ledger. For this reason, after receiving the proposal the application will ensure the endorsement policy holds true, in this case that one peer has endorsed it, and then send it to the ordering nodes.

The ordering nodes will order this and other transactions chronologically and package them into blocks, which are saved and distributed to all the peers in the network. It must be mentioned here that ordering is deterministic and not probabilistic, that is, the blocks are final and there cannot be a ledger fork. Last but not least, the peers that receive the block will need to validate all the transactions in it, to ensure the endorsements are correct and match the endorsement policy,

as well as no transaction is invalid. In the latter case, those transactions are marked as invalid and do not update the ledger. In the end, the peers will immutably append the block to the blockchain, and all the ledger updates mentioned in all the transactions inside the block will be performed on the ledger, updating its state to be the most recent one.

As discussed, attacks are eliminated by verifying the signature of the end user. Because the end user applications are external to the blockchain, anyone who can replicate one and forge a communication packet that for instance includes a transaction proposal would in theory be able to interact with the network. However this assumes that they are able to replicate the transaction proposal correctly, otherwise it would be rejected. Further, the lack of verification would prevent an attack from continuing, and it would be recorded for further investigation in the incident. Even in the case of identity, or in practice certificate, theft which is unlikely, the attacker will need to format a proposal correctly, and only have limited access to the network, as permissions are allocated only to those who need them. This shows that the system is secure, but of course not impenetrable, however by combining multi factor authentication technologies like biometrics or physical tokens, the user error can be virtually eliminated.

## IV. The role of RAFT in our proposed configuration

In this section, we make a case for using RAFT in our stylized healthcare configuration that is built upon Hyperledger Fabric discussed in the previous section.

As mentioned, an organization serves as an umbrella for five different hospitals $(H_1, H_2, H_3, H_4$ and $H_5)$. Our hospitals have 10 servers, and each hospital has a *peer* node $P$ and an *orderer* node $O$. Since only the orderer nodes are performing consensus and thus eligible to be a leader node, the organization needs at least $(O/2)+1$ as a quorum, which must endorse the request sent by the leader. More clearly, there is a need for at least 3 majority nodes and the organization can be tolerated with 2 failed nodes. Orderer nodes $O_1 \rightarrow O_5$ can be in one of three states: follower F, candidate C, and leader L. It can now be assumed that they are in their initial phase of operation, and the first cluster starts up (term 1). Now, each node among the orderer nodes $O_1 \rightarrow O_5$ will pick a different random timeout and then start counting it down. This timeout represents the interval that this node will wait until it moves from its $F$ state to $C$ state. When the time runs out for each node respectively, the following scenario will happen:

$$O_1H_1 \xrightarrow{vote\ request} O_2H_2, O_3H_3, O_4H_4, O_5H_5$$
$$O_2H_2 \xrightarrow{vote\ request} O_1H_1, O_3H_3, O_4H_4, O_5H_5$$
$$O_3H_3 \xrightarrow{vote\ request} O_1H_1, O_2H_2, O_4H_4, O_5H_5$$
$$O_4H_4 \xrightarrow{vote\ request} O_1H_1, O_2H_2, O_3H_3, O_5H_5$$
$$O_5H_5 \xrightarrow{vote\ request} O_1H_1, O_2H_2, O_3H_3, O_4H_4$$

Let us suppose that $O_5H_5$ runs out its time out first. In this case, $O_5H_5$ will be getting into $C$ state and then increasing its current term by 1. Since $O_5H_5$ is now eligible for a leadership position, the very first step in this direction is to vote for itself and then send a RequestVote to all other orderer nodes $O_1 \rightarrow O_4$. These orderer nodes must now receive the request, and if these nodes have not yet cast their vote, they reset their term to match the term of $O_5H_5$ and then cast their votes to $O_5H_5$. Has the majority responded to the vote request, $O_5H_5$ is now the leader of the entire organization. The leader $O_5H_5$ then begins to send AppendEntries messages to all orderer nodes $O_1 \rightarrow O_4$ from time to time to inform them of its continued authority as a leader, and to prevent any new election during the current term. All orderer nodes in all hospitals must then reply to the request message sent by the leader. However, the current term could be terminated the moment that $O_1 \rightarrow O_4$ nodes are no longer hearing from the $O_5H_5$ node. In this case, $O_1 \rightarrow O_4$ will assume a node termination scenario and switch to state $C$. Fig. 5 gives an overview of the proposed scenario.

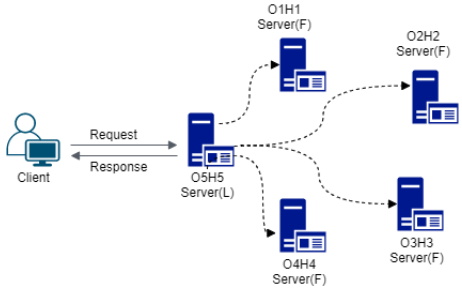


Fig. 5. The process of updating the follower logs by the leader server.

As indicated above, the organization is now led by the $O_5$ server located in hospital 5. Any ledger updates in the organization should now go through $O_5$. In the previous example, a doctor needed to update a patient's health record with recent test results. It can be assumed here that this doctor is working on $H_3$ and interacts with $P_3$, even though the latter will not always be the case. After the transaction proposal has been endorsed and a response is sent, the application will submit the transaction to $O_5$ where it is arrives among other transactions in need for ordering. However, even though these transaction have now been received, the state of $O_5$ will not be updated, given that the other orderer nodes have not yet acknowledged the submitted transactions. In order for the transactions to be packaged into a block, $O_5$ must now replicate them over $O_1, O_2, O_3$ and $O_4$. If it is assumed that $O_1, O_2, O_3$ and $O_4$ successfully added the transactions to their logs, while $O_2$ has failed to synchronize, then in this case the transactions will still be committed on $O_5$ based on the majority's confirmation. Finally, $O_5$ updates its log, creates the block and distributes it to all the peers.

It can be shown that the resulting blockchain transaction log is kept consistent due to RAFT safety guarantees and that the blockchain will be made correct and available for arbitrary sequences of transactions. Closer inspection of the leader election in RAFT shows that the algorithm maintains suitable leaders in a dynamically evolving situation.

## V. Conclusion and Future work

Our paper shows how RAFT can add value to data record systems and value chains, in particularly healthcare, using Hyperledger Fabric. This matters because the academic literature falls short in explaining how RAFT (one of the leading consensus mechanisms on Hyperledger Fabric) can play an essential role in decentralizing healthcare applications.

RAFT operates quickly and efficiently, on par with the PAXOS consensus. It can guarantee safety by using a new, two phase, mechanism when introducing new consensus nodes, thus avoiding the possibility that two majorities will be formed during configuration. It can also be seen that it is easier to understand and implement, speeding up both learning as well as development and maintenance of healthcare networks. Consequently, by accelerating blockchain development in healthcare, RAFT can provide decentralized healthcare applications that benefit from distributed ledger technologies. More precisely, powerful decentralized and append-only immutable ledgers can offer fault-tolerance in the form of redundancy, prevent cascaded failure, and offer trustworthy logging and auditing. Further, with the use of certificates and identity verification, strict access control and the property of non repudiation can be enforced. Other related fields and data record systems can also benefit from blockchain, and therefore RAFT consensus.

This paper proposes a healthcare network enabled by RAFT and built upon the Hyperledger Fabric architecture. From a technical perspective, we can confidently say that RAFT is a good candidate for healthcare and other related fields handling a large volume of data, and worth investigating further. In the future, the feasibility of implementation could be looked into further, as well as the inclusion of other organizations, such as the government, or insurance companies, for a complete system. Implementation and maintenance costs are also critical and need to be carefully examined if the system were to be built. Further improvements on the security and reliability of data are also possible. A system of this level of complexity has a multitude of attack vectors. While best practices can enhance security, it is critical to reducing user error and the impact of social engineering attacks. The use of biometrics and physical security tokens discussed above can be such an approach, eliminating the need for passwords. This can be investigated in more detail. Data reliability is also of importance, with smart contracts and endorsement policies able to offer some help. For instance, smart contracts can perform some basic error checking, and the more the system is automated and reliant on smart contracts, the more trustworthy the data can be. Finally, big data research can benefit from an append-only ledger found in a blockchain-based healthcare network. Hyperledger Fabric, in its recent versions, can take multi-thousands of transactions. However, we doubt that a decentralized healthcare system enabled by RAFT and built upon Hyperledger Fabric can handle a large quantity of data (in the case of implementing embedded medical sensors). We expect research regarding the scalability of decentralized healthcare systems that integrate big data analytics to be studied further, as blockchain and RAFT consensus are by no means limited to healthcare; there is value to be found in other related fields and distributed data record systems.

## References


[1] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, pp. 305–319, 2014.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[3] D. E. O'Leary, "Configuring blockchain architectures for transaction information in blockchain consortiums: The case of accounting and supply chain systems," *Intelligent Systems in Accounting, Finance and Management*, vol. 24, no. 4, pp. 138–147, 2017.

[4] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019.

[5] V. L. Lemieux, A. Mashatan, R. Safavi-Naini, and J. Clark, "A cross-pollination of ideas about distributed ledger technological innovation through a multidisciplinary and multisectoral lens: Insights from the blockchain technology symposium'21," *Technology Innovation Management Review*, vol. 11, no. 6, 2021.

[6] R. Alkhudary, "Blockchain technology between nakamoto and supply chain management: Insights from academia and practice," *Available at SSRN 3660342*, 2020.

[7] V. L. Lemieux, D. Hofman, H. Hamouda, D. Batista, R. Kaur, W. Pan, I. Costanzo, D. Regier, S. Pollard, D. Weymann, *et al.*, "Having our "omic" cake and eating it too?: Evaluating user response to using blockchain technology for private and secure health data management and sharing," *Frontiers in Blockchain*, vol. 3, p. 59, 2021.

[8] A. Yazdinejad, G. Srivastava, R. M. Parizi, A. Dehghantanha, K.-K. R. Choo, and M. Aledhari, "Decentralized authentication of distributed patients in hospital networks using blockchain," *IEEE journal of biomedical and health informatics*, vol. 24, no. 8, pp. 2146–2156, 2020.

[9] R. Akkaoui, X. Hei, and W. Cheng, "Edgemedichain: A hybrid edge blockchain-based framework for health data exchange," *IEEE Access*, vol. 8, pp. 113467–113486, 2020.

[10] V. Malamas, P. Kotzanikolaou, T. K. Dasaklis, and M. Burmester, "A hierarchical multi blockchain for fine grained access to medical data," *IEEE Access*, vol. 8, pp. 134393–134412, 2020.

[11] M. Zghaibeh, U. Farooq, N. U. Hasan, and I. Baig, "Shealth: a blockchain-based health system with smart contracts capabilities," *IEEE Access*, vol. 8, pp. 70030–70043, 2020.

[12] X. Zhu, J. Shi, and C. Lu, "Cloud health resource sharing based on consensus-oriented blockchain technology: Case study on a breast tumor diagnosis service," *Journal of medical Internet research*, vol. 21, no. 7, p. e13767, 2019.

[13] H.-A. Lee, H.-H. Kung, J. G. Udayasankaran, B. Kijsanayotin, A. B. Marcelo, L. R. Chao, C.-Y. Hsu, *et al.*, "An architecture and management platform for blockchain-based personal health record exchange: Development and usability study," *Journal of Medical Internet Research*, vol. 22, no. 6, p. e16748, 2020.

[14] S. M. Pournaghi, M. Bayat, and Y. Farjami, "Medsba: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–29, 2020.

[15] G. Al-Sumaidaee, R. Alkhudary, Z. Zilic, and P. Féniès, "A blueprint towards an integrated healthcare information system through blockchain technology," *HEALTHINFO 2021, The Sixth International Conference on Informatics and Assistive Technologies for Health-Care, Medical Support and Wellbeing*, 2021.

[16] R. Alkhudary, X. Brusset, and P. Fenies, "Blockchain in general management and economics: A systematic literature review," *European Business Review*, 2020.

[17] S. F. Wamba and M. M. Queiroz, "Blockchain in the operations and supply chain management: Benefits, challenges and future research opportunities," 2020.

[18] H. Treiblmaier, "Exploring the next wave of blockchain and distributed ledger technology: The overlooked potential of scenario analysis," *Future Internet*, vol. 13, no. 7, p. 183, 2021.

[19] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems*, vol. 16, no. 2, pp. 133–169, 1998.

[20] E. J. De Aguiar, B. S. Faiçal, B. Krishnamachari, and J. Ueyama, "A survey of blockchain-based strategies for healthcare," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–27, 2020.

[21] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *Journal of Network and Computer Applications*, vol. 135, pp. 62–75, 2019.

[22] T. Benil and J. Jasper, "Cloud based security on outsourcing using blockchain in e-health systems," *Computer Networks*, vol. 178, p. 107344, 2020.

[23] G. Leeming, J. Ainsworth, and D. A. Clifton, "Blockchain in health care: hype, trust, and digital health," *The Lancet*, vol. 393, no. 10190, pp. 2476–2477, 2019.

[24] F. Curbera, D. Dias, V. Simonyan, W. Yoon, and A. Casella, "Blockchain: An enabler for healthcare and life sciences transformation," *IBM Journal of Research and Development*, vol. 63, no. 2/3, pp. 8–1, 2019.

[25] A. Tandon, A. Dhir, N. Islam, and M. Mäntymäki, "Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda," *Computers in Industry*, vol. 122, p. 103290, 2020.

[26] S. Shi, D. He, L. Li, N. Kumar, M. K. Khan, and K.-K. R. Choo, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey," *Computers & Security*, p. 101966, 2020.

[27] M. Greenberger, "Block what? the unrealized potential of blockchain in healthcare," *Nursing management*, vol. 50, no. 5, pp. 9–12, 2019.

[28] R. Casado-Vara and J. Corchado, "Distributed e-health wide-world accounting ledger via blockchain," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 2381–2386, 2019.

[29] X. Yang, T. Li, X. Pei, L. Wen, and C. Wang, "Medical data sharing scheme based on attribute cryptosystem and blockchain technology," *IEEE Access*, vol. 8, pp. 45468–45476, 2020.

[30] Z. Wang, N. Luo, and P. Zhou, "Guardhealth: Blockchain empowered secure data management and graph convolutional network enabled anomaly detection in smart healthcare," *Journal of Parallel and Distributed Computing*, vol. 142, pp. 1–12, 2020.

[31] J. Wang, K. Han, A. Alexandridis, Z. Chen, Z. Zilic, Y. Pang, G. Jeon, and F. Piccialli, "A blockchain-based ehealthcare system interoperating with wbans," *Future Generation computer systems*, vol. 110, pp. 675–685, 2020.

[32] J. Wang, S. Fan, A. Alexandridis, K. Han, G. Jeon, Z. Zilic, and Y. Pang, "A multistage blockchain-based secure and trustworthy smart healthcare system using ecg characteristic," *IEEE Internet of Things Magazine*, vol. 4, no. 3, pp. 48–58, 2021.

[33] D. Woos, J. R. Wilcox, S. Anton, Z. Tatlock, M. D. Ernst, and T. Anderson, "Planning for change in a formal verification of the raft consensus protocol," in *Proceedings of the 5th ACM SIGPLAN Conference on Certified Programs and Proofs*, pp. 154–165, 2016.

[34] J. Hu and K. Liu, "Raft consensus mechanism and the applications," in *Journal of Physics: Conference Series*, vol. 1544, p. 012079, IOP Publishing, 2020.

[35] K. Nath, "Making sense of the raft distributed consensus algorithm — part 1," 2021.

[36] H. Ju, X. Zhang, H. Jia, X. Zhang, E. Zhu, K. Yan, and J. Guo, "A survey on efficient consensus mechanism for electricity information acquisition system," in *2021 9th International Conference on Smart Grid (icSmartGrid)*, pp. 124–127, IEEE, 2021.

[37] G. R. Carrara, L. M. Burle, D. S. Medeiros, C. V. N. de Albuquerque, and D. M. Mattos, "Consistency, availability, and partition tolerance in blockchain: a survey on the consensus mechanism over peer-to-peer networking," *Annals of Telecommunications*, vol. 75, no. 3, pp. 163–174, 2020.