Comparison of Smart Contract Blockchains for Healthcare Applications

Hongru Yu[1,*], Haiyang Sun, BE[2,*], Danyi Wu[1,*], and Tsung-Ting Kuo, PhD[1]

[1] University of California San Diego, La Jolla, CA, USA;
[2] Syracuse University, Syracuse, NY, USA;
[*] Contributed Equally

Abstract

Blockchain and smart contracts (i.e., computer code that can be run on blockchain) are increasingly popular for healthcare applications. However, only very few implementations exist because of the complexity of the technologies. Although there are tutorials and reviews to introduce blockchain and smart contracts, a pragmatic comparison of such platforms is needed. In this study, we addressed practical considerations while building a healthcare blockchain and smart contract system, by (1) comparing technical features of platforms, (2) selecting three platforms, (3) constructing blockchain networks, (4) testing the blockchains, and (5) summarizing the experience and time used for implementation by students. We evaluated Ethereum, Hyperledger Fabric, and MultiChain, and confirmed that the selection of a proper platform depends on the requirements of the application. The findings of our study can accelerate the process and reduce the risk of adopting blockchain technology in biomedical and healthcare domain.

Keywords: Data Sharing, Interoperability and Health Information Exchange, Privacy and Security

Learning Objective: Understand smart contract blockchain platforms and their comparison for healthcare.

Introduction

Since Bitcoin,[1, 2] blockchain technology has evolved and is now considered as a new form of a distributed ledger, since arbitrary data can be stored in the metadata of the transactions.[3] With this capability, many blockchain-based healthcare applications are emerging rapidly. Example applications include unalterable patient records,[4] improved care data sharing/analysis without ceding control,[5-7] and enhanced robustness for counterfeit drug prevention/detection systems in pharmaceutical supply chains.[8]

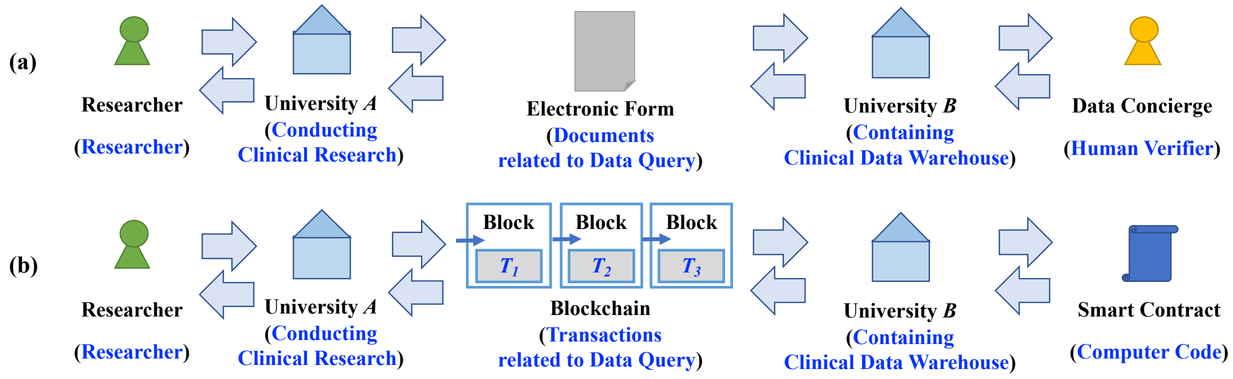


Figure 1. Example of smart contract for data queries such as “how many patients have hemoglobin test results for postmenopausal females”. **(a)** Traditional process of a researcher requesting a data query within a clinical data research network (i.e., from University *A* to University *B*), including the use of an electronic form as well as the need for a data concierge person to manage and respond to the queries. **(b)** Blockchain-based smart contract approach to manage/record the data queries automatically without human. Note that the choice of the query depends on the Institutional Review Board (IRB) approval as well as the Data Usage Agreement (DUA) signed between universities.

Among the new developments of the blockchain technology, smart contract is one of the most important advancements.[9-19] Smart contracts refer to computer programs/code devised to manage smart properties, which are the digital properties recorded by blockchain.[11] That is, a smart contract is a piece of computer code that can be run within a blockchain transaction. For example, a smart contract has been proposed to automate the workflow of medical licensure across multiple states.[20, 21] It removes the requirement for human confirmation of precursors and results, and instead relies on machines to automatically execute the logic stated in the program to accept or reject a request. Smart contracts are also transparent and immutable. An example is shown in Figure 1, demonstrating the potential to manage the data queries requested by an institution to another within a clinical data research network. Therefore, we consider smart contract as a very useful and desired feature for healthcare applications.

Many researchers have proposed to adopt blockchain and smart contracts in healthcare, genomic, or biomedical applications,[4-6, 8, 22-35] but only very few have actually implemented these technologies due to their complexity.[3] Therefore, understanding practical considerations while exploring smart contract technology is imperative, and can facilitate the adoption of the new technology for a wild spectrum of healthcare applications. Although there are broad tutorials and reviews about blockchain and smart contracts,[36-39] a pragmatic comparison and evaluation of the smart contract platforms is needed.

Table 1. Technical features to compare smart contract blockchain platforms.

Technical Feature	**Description**
Type of Smart Contract	The type of smart contract supported in the platform, can be either first generation (i.e., supports scripting functions without full programming capacity such as looping) or second generation (i.e., supports any algorithm).
Programming Language	The language for smart contract (e.g., Solidity[40-42] for Ethereum[11, 43]).
Network Permission	The supported permission type of blockchain network, can be permissionless (i.e., any machine can join the network freely, as for Bitcoin Blockchain), permissioned (i.e., only an authenticated machine can join the network), or both.
Open Source License	The license to distribute derivative products (e.g., MIT[44] or GPL[45])
Applications	Proposed applications in healthcare.

Methods

In this study, we addressed the practical considerations while building a healthcare blockchain and smart contract system. Specifically, we compared the technical features of the smart contract blockchain platforms, selected platforms for our study, constructed a blockchain network for each of the platform, tested the blockchains, and summarized our experience and time used for the implementation by students. First, we selected technical features of the smart contract blockchain platforms based on our recent review/report,[3, 38] as shown in Table 1. Based on the technical features described in Table 1, we identified three platforms to be compared in this study, out of ten popular platforms:[3, 38]

- Ethereum[11, 43] is one of the most well-known decentralized smart contract blockchain platforms.[11, 17, 43, 46] Ethereum is also a cryptocurrency itself, and has a large market cap.[47] Microsoft adopted Ethereum as the core of its Blockchain-as-a-Service (BaaS) on the Azure cloud computing environment,[48] however Ethereum can be run in any cloud computing environment. Ethereum has been used in healthcare applications and, most importantly, supports second-generation smart contracts.[3] Ethereum can work with both external owned accounts and internal contract accounts.[49] It also uses the concept of 'gas' to regulate computationally intensive smart contract code.[46] Ethereum has a build-in currency (Ether), and Wei is the smallest fraction of Ether.[50] The main reasons to select Ethereum were the capability to be configured as both a permission-less and a permissioned blockchain network, as well as the community-based development of the platform.

- Hyperledger Fabric[51-53] is an enterprise-grade cross-industry platform supported by IBM.[53, 54] Hyperledger, a project under the Linux Foundation,[51, 52] is a platform with various "dialects", and we selected Hyperledger Fabric because it is one of the popular smart contract platform that requires no specific hardware to be executed.[3, 38] Hyperledger also has a Composer component that supports a purpose-built modeling language for defining a business network, multiple ways of integrating blockchains with other services, and a platform for developing, testing and deploying custom business logic.[55] The Hyperledger Composer, along with its Playground feature, enables developers to use pre-existing sample network frameworks to plan their network construction through a user-friendly, easily accessible web interface. There are also Software Development Kits (SDKs) available on Hyperledger Fabric's website for potential C and Java developers. The main considerations to choose Hyperledger Fabric were the capability to be configured as a permissioned blockchain network and the access control feature with high granularity.

- MultiChain[56, 57] is a platform that currently only supports first-generation smart contracts as well as limited programs, or "smart filters".[58] It also does not yet support complex syntaxes based on Turing-complete programming languages that can execute any algorithm. Currently, a MultiChain smart filter is restricted to programming instructions that do not allow for operations such as loops and recursion. However, MultiChain provides many built-in functions to support basic building blocks for applications,[58] and can be used directly for operations like sending or receiving transactions. MultiChain also supports data streams (i.e., key-value pairs) based on transaction metadata.[59] Many MultiChain APIs have been developed for mainstream programming languages such as Java and JavaScript, enabling developers to run complex programs on MultiChain. The main reasons to include MultiChain were the simple setup process and the features inherited from Bitcoin Blockchain.[1, 2]

Our comparative analysis was conducted by three undergraduate/graduate students with no prior experience in blockchain technologies to construct, test, and summarize the experiences related to the three selected platforms in parallel. While only one student developed code for each one of the platforms, they compared notes and discussed solutions as a group, with facilitation from a faculty member (T-TK). First, we constructed three selected smart contract blockchain networks. We used six virtual machines (VMs), each has 2-Core CPU, 8GB RAMs and 100GB storage, with 64-bit Ubuntu 14.04 operating system, on the Google Cloud Platform.[60] We then tested the blockchain platforms by submitting transactions or deploying/executing smart contracts on them. Finally, we summarized our implementation experiences for these blockchain platforms, with an estimation of the required time for setup and learning time for an undergraduate or a graduate student.

Results

We compared the platforms by extracting the technical features,[3, 38] and the results are shown in Table 2. Our experience and setup/learning time of constructing three blockchain platforms are summarized below.

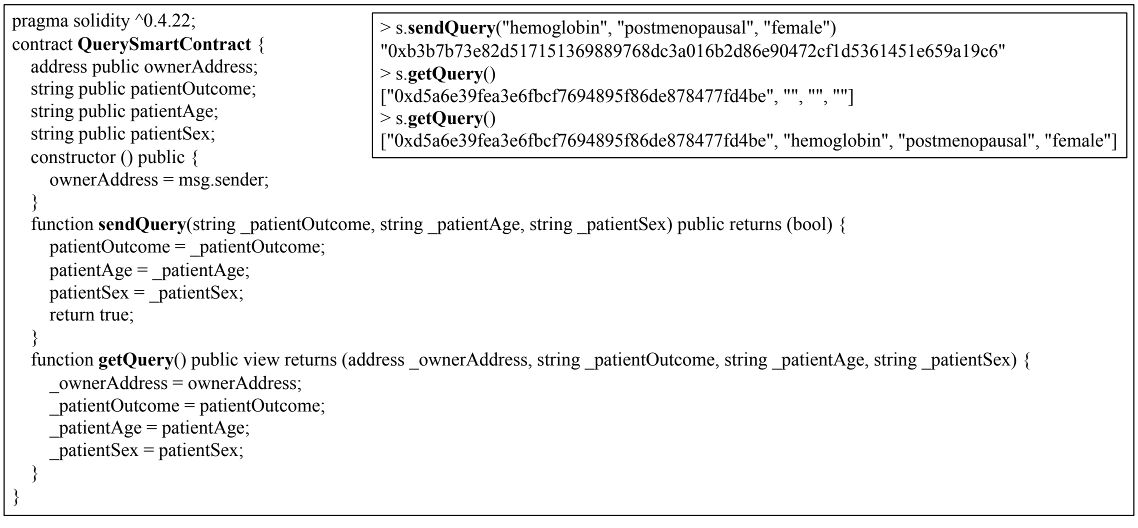
- *Ethereum*. We constructed a blockchain network using Go-Ethereum.[61] Then, we sent transactions with specified data and checked the transaction input for both servers were the same. Finally, we wrote, compiled, deployed and executed smart contracts using Solidity[40, 41] on our blockchain network. An example smart contract pseudo code written in Solidity and executed on Ethereum[11, 43] is shown in Figure 2, which corresponds to the use case described in Figure 1. The whole process to build the blockchain network took about 30 minutes, while the learning time for an undergraduate student (DW) was about 7 days.

- *Hyperledger Fabric*. We built a blockchain network using the Hyperledger Fabric images on Docker.[62] Then, we send transactions to verify the connectivity of the network, and implemented/executed smart contracts using Chaincode in the Go language. An example pseudo code, corresponding to Figure 1 (b), is shown in Figure 3. The whole process to build the blockchain network took about 45 minutes, and the overall learning period for an undergraduate student (HY) was about 14 days.

- *MultiChain*. We setup a blockchain network using MultiChain.[57] Then, we sent transactions between nodes to ensure the validity of the network. We also tried to create data streams to disseminate key-value paired data. Finally, we investigated smart filters, and an example pseudo code is illustrated in Figure 4. The whole process to build the blockchain network took about 30 minutes, while the overall learning period for a graduate student (HS) was about 2 days.

Table 2. Comparison of the technical features of the platforms and their applications.[3, 38]

Platform	Ethereum	Hyperledger Fabric	MultiChain
Type of Smart Contract	Second Generation	Second Generation	First Generation
Programming Language	Solidity, Serpent, and LLL[63]	Go and Node.js for Chaincode[52]	JavaScript for Smart Filter[64]
Network Permission	Permissioned/Permissionless	Permissioned	Permissioned
Open Source License	Go-Ethereum: LGPL v3.0,[61] CPP-Ethereum: GPL v3.0,[65] Py-Ethereum: MIT License,[66] EthereumJ: GPL v3.0,[67] and Parity: GPL v3.0 [68]	Apache License v2.0[69]	GPL v3.0[70]
Applications	MedRec,[31, 32] Patientory,[71, 72] Nebula Genomics,[73, 74] clinical data sharing,[75] and patient monitoring.[76]	Clinical data sharing,[77] Institutional Review Boards (IRB) regulation enforcement,[78] mobile healthcare,[79] and medical data storage/access.[80]	Privacy-preserving predictive modeling[5, 7] and its application in anesthesia.[6]



```
> s.sendQuery("hemoglobin", "postmenopausal", "female")
"0xb3b7b73e82d51715136989768dc3a016b2d86e90472cf1d5361451e659a19c6"
> s.getQuery()
["0xd5a6e39fea3e6fbcf7694895f86de878477fd4be", "", "", ""]
> s.getQuery()
["0xd5a6e39fea3e6fbcf7694895f86de878477fd4be", "hemoglobin", "postmenopausal", "female"]
```

```
pragma solidity ^0.4.22;
contract QuerySmartContract {
  address public ownerAddress;
  string public patientOutcome;
  string public patientAge;
  string public patientSex;
  constructor () public {
    ownerAddress = msg.sender;
  }
  function sendQuery(string _patientOutcome, string _patientAge, string _patientSex) public returns (bool) {
    patientOutcome = _patientOutcome;
    patientAge = _patientAge;
    patientSex = _patientSex;
    return true;
  }
  function getQuery() public view returns (address _ownerAddress, string _patientOutcome, string _patientAge, string _patientSex) {
    _ownerAddress = ownerAddress;
    _patientOutcome = patientOutcome;
    _patientAge = patientAge;
    _patientSex = patientSex;
  }
}
```

Figure 2. Example Ethereum Solidity pseudo code involved in a simple contract between two institutions (as described in Figure 1 (b)). University *B* deployed this *QuerySmartContract*, with two functions: *sendQuery* and *getQuery*, to the blockchain network. University *A* executed this smart contract and queried the data for "how many patients have hemoglobin test results for postmenopausal females" (the upper-right box). Then, University *A* retrieved the query from the blockchain, and can see the query criteria as well as the address of the contract owner (which is University *B*). Note that it takes some time for the smart contract to be enclosed in a verified transaction of a block, therefore the first query only returns empty values for the query. Also, this smart contract, as well as any function calls that change the state of the variables in the smart contract, will be logged in the immutable blockchain, which enables future attribution, auditing, dispute and resolution for both (and even a third-party) institutions.

```
func (t *SimpleChaincode) sendQuery(stub
shim.ChaincodeStubInterface) pb.Response {
    fmt.Println("sendQuery")

    _, args := stub.GetFunctionAndParameters()

    var O, A, S string
    var patientOutcome, patientAge, patientSex string

    O = args[0]
    patientOutcome = args[1]
    A= args[2]
    patientAge = args[3]
    S = args[4]
    patientSex = args[5]

    fmt.Printf("%s, %s, %s\n", patientOutcome,
patientAge, patientSex)

    stub.PutState(O, []byte(patientOutcome))
    stub.PutState(A, []byte(patientAge))
    stub.PutState(S, []byte(patientSex))

    return shim.Success(nil)
}
```

```
> sendQuery ("patientOutcome",
"hemoglobin", "patientAge",
"postmenopausal", "patientSex", "female")

> getQuery()
hemoglobin, postmenopausal, female
```

```
func (t *SimpleChaincode) getQuery( )
pb.Response {
    var O, A, S string
    var err error

    O = "patientOutcome"
    A = "patientAge"
    S = "patientSex"

    outcome, err := stub.GetState(O)
    age, err := stub.GetState(A)
    sex, err := stub.GetState(S)

    fmt.Printf(%s, %s, %s\n", patientOutcome,
patientAge, patientSex)

    return shim.Success(nil)
}
```

Figure 3. Example Hyperledger Fabric pseudo code for the example shown in Figure 1 (b), and the execution results are shown in the upper-right box. Similar to the code shown in Figure 2, University *B* deployed this contract, and University *A* send the query (by calling *sendQuery* and *getQuery* functions).

```
University A:

$ multichain-cli query-chain
Interactive mode

> create stream query false
{"method": "create", "params": ["stream", "query", false], "id": "1","chain_name":"query-chain"}
c84a8aac79287a064c26a04ad971eafb35a03ee01e3d033216bfd8f6b1965b96

> publish query patientOutcome 68656d6f676c6f62696e
{"method": "publish", "params": ["query", "patientOutcome", "68656d6f676c6f62696e"], "id": "2","chain_name":"chain1"}
f22c6c4b38e0932218629380f5559afedaec8dfb99569f4eef3d9acd889b9777
```

```
University B:

$ multichain-cli query-chain
Interactive mode

> subscribe query
{"method": "subscribe", "params": ["query"],"id": "3" , "chain_name":" query-chain"}
2a7c93f9f2ecab342ca8c3279349ee39843a9a92f23e874b2a47298394728900

> liststreamkeyitems query patientOutcome
{"method": "liststreamkeyitems", "params": ["query", "patientOutcome"], "id": "4", "chain_name":"chain1"}
[data: " 68656d6f676c6f62696e"]
```

Figure 4. Example MultiChain code between two universities. University *A* creates a stream *query* in the blockchain, and publishes a key (*patientOutcome* only, for simplicity purpose) with a value (*hemoglobin*, which is represented in hexadecimal number "*68656d6f676c6f62696e*") in the stream. Then, University *B* subscribes the stream to inspect the value of the *patientOutcome*.

Comparison and Discussion

Based on our implementations, MultiChain is the easiest blockchain platform in terms of setup. The setup of Ethereum as a permissioned network took longer than MultiChain. Hyperledger Fabric contains more layers in its network to increases manageability and security, which can result in long setup time compared to other platforms. The installation of the software prerequisites for MultiChain and Ethereum are also easier comparing to Hyperledger Fabric. On the other hand, Ethereum and Hyperledger provide the full-functional smart contract capability, while the support of smart contract for MultiChain is limited. The smart contracts are highly readable and easily programmable.

Considering the biomedical/healthcare applications, the salient characteristic of the three platforms are as follows. Ethereum is supported and maintained by a large community of developers around the world. This open property makes it a good choice while considering the long-term sustainability of the platform that the applications rely on. Hyperledger Fabric's well-designed, multi-layered access control framework along with Hyperledger Fabric's own certificates leads to high versatility, security, and manageability of the blockchain platform. MultiChain is designed to be a permissioned blockchain that is very simple yet powerful to use and inherits proven features of the famous Bitcoin Blockchain by forking from it.

Conclusion

To support the development of healthcare applications based on blockchain and smart contracts, we conducted a pragmatic comparison and evaluation of smart contract blockchain platforms. The main limitation of this study is that we only compared three platforms, so the scope may be restricted. Based on our results, an informatics researcher, IT expert, or technical leader in healthcare or other institution can assess the various practical aspects, such as setup/learning time and unique technical features, of the platforms. The selection of a proper platform depends on requirements of the application (e.g., maintenance for Ethereum, fine-grained access control for Hyperledger Fabric, and rapid-development for MultiChain). Such an assessment can accelerate the process and reduce the risks in adopting blockchain, an immutable, distributed, and automated technology, for biomedical and healthcare applications. We plan to identify more blockchain requirements for healthcare applications, conduct more structured and large-scale comparisons to include additional blockchain platforms (such as Corda[81-83]) that may fulfill the requirements, and analyze the operation/performance aspects of the platforms in our future work.

Acknowledgements

The authors would like to thank Lucila Ohno-Machado, MD, PhD, and Jejo David Koola, MD, for very helpful discussions and support. Research reported in this publication was supported by the National Human Genome Research Institute (NHGRI), the National Library of Medicine (NLM), and the Office of the Director (OD), National Institutes of Health (NIH). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH. All authors were partly funded by NIH, including T-TK (OT3OD025462, K99/R00HG009680), HS (OT3OD025462, K99/R00HG009680), HY (K99/R00HG009680), and DW (T15LM011271). This research benefited from the use of credits from the National Institutes of Health (NIH) Cloud Credits Model Pilot, a component of the NIH Big Data to Knowledge (BD2K) program. T-TK is also funded by the UCSD Academic Senate Research Grant RG084150 and Travel Award TG087935.

References

1. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Available from: https://bitcoin.org/bitcoin.pdf. 2008 (Accessed July 3, 2019)

2. TheBitcoinProject. Bitcoin. Available from: https://bitcoin.org/en/. (Accessed July 5, 2019)

3. TeamO2 DCPPC. Towards a Sustainable Commons: The Role of Blockchain Technology. Available from: https://public.nihdatacommons.us/Blockchain/. 2018 (Accessed July 3, 2019)

4. Yue X, Wang H, Jin D, Li M, Jiang W. Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control. Journal of medical systems. 2016;40(10):218.

5. Kuo T-T, Hsu C-N, Ohno-Machado L. ModelChain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.

6. Kuo T-T, Gabriel RA, Ohno-Machado L. Fair compute loads enabled by blockchain: sharing models by alternating client and server roles. Journal of the American Medical Informatics Association. 2019;26(5):392-403. Epub March 20, 2019. doi: 10.1093/jamia/ocy180. PubMed PMID: 30892656.

7. Kuo T-T, Gabriel RA, Ohno-Machado L. EXpectation Propagation LOgistic REgRession on Permissioned BlockCHAIN (ExplorerChain): Decentralized Privacy-Preserving Online Healthcare/Genomics Predictive Model Learning. Zenodo. Available from: https://doi.org/10.5281/zenodo.1492820. 2018 (Accessed July 3, 2019)

8. Attili S, Ladwa SK, Sharma U, Trenkle AF. Blockchain: The Chain of Trust and its Potential to Transform Healthcare – Our Point of View. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.

9. Rosenfeld M. Overview of colored coins. Available from: https://bitcoil.co.il/BitcoinX.pdf. 2012 (Accessed July 3, 2019)

10. Back A, Corallo M, Dashjr L, Friedenbach M, Maxwell G, Miller A, Poelstra A, Timón J, Wuille P. Enabling blockchain innovations with pegged sidechains. Available from: https://blockstream.com/sidechains.pdf. 2014 (Accessed July 3, 2019)

11. Buterin V. A next-generation smart contract and decentralized application platform. Available from: https://github.com/ethereum/wiki/wiki/White-Paper. 2014 (Accessed July 3, 2019)

12. Omohundro S. Cryptocurrencies, smart contracts, and artificial intelligence. AI matters. 2014;1(2):19-21.

13. Swan M. Blockchain: Blueprint for a new economy. Sebastopol, CA, Unitited States: O'Reilly Media, Inc.; 2015.

14. Swan M. Blockchain thinking: The brain as a dac (decentralized autonomous organization). Texas Bitcoin Conference; Austin, Texas, United States: Texas Bitcoin Association; 2015. p. 27-9.

15. Swan M. Blockchain Temporality: Smart Contract Time Specifiability with Blocktime. International Symposium on Rules and Rule Markup Languages for the Semantic Web; Stony Brook, New York, United States: Springer International Publishing; 2016. p. 184-96.

16. Szabo N. The idea of smart contracts. Available from: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html. 1997 (Accessed July 3, 2019)

17. Wood G. Ethereum: A secure decentralised generalised transaction ledger. Available from: https://gavwood.com/paper.pdf. 2014 (Accessed July 3, 2019)

18. Watanabe H, Fujimura S, Nakadaira A, Miyazaki Y, Akutsu A, Kishigami J. Blockchain contract: Securing a blockchain applied to smart contracts. 2016 IEEE International Conference on Consumer Electronics (ICCE); Las Vegas, Nevada, United States: IEEE; 2016. p. 467-8.

19. Kosba A, Miller A, Shi E, Wen Z, Papamanthou C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. 2016 IEEE Symposium on Security and Privacy (SP); San Jose, California, United States: IEEE; 2016. p. 839-58.

20. Mackey TK, Kuo T-T, Gummadi B, Clauson KA, Church G, Grishin D, Obbad K, Barkovich R, Palombini M. 'Fit-for-purpose?' – challenges and opportunities for applications of blockchain technology in the future of healthcare. BMC Medicine. 2019;17(1):68. doi: 10.1186/s12916-019-1296-7. PubMed PMID: 30914045.

21. Weiss D. The State of Illinois Moves Provider Credentialing to the Blockchain. Available from: https://distributed.com/news/state-illinois-moves-provider-credentialing-blockchain. (Accessed July 3, 2019)

22. Kuo T-T, Kim H-E, Ohno-Machado L. Blockchain distributed ledger technologies for biomedical and health care applications. Journal of the American Medical Informatics Association. 2017;24(6):1211-20. Epub September 8, 2017. PubMed PMID: 29016974; PMCID: PMC6080687.

23. Ivan D. Moving Toward a Blockchain-based Method for the Secure Storage of Patient Records. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.
24. Culver K. Blockchain Technologies: A whitepaper discussing how the claims process can be improved. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.
25. Goldwater J. The Use of a Blockchain to Foster the Development of Patient-Reported Outcome Measures. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.
26. Healthbank.coop. HealthBank. Available from: https://www.healthbank.coop. (Accessed July 5, 2019)
27. Mettler M. Blockchain technology in healthcare: The revolution starts here. 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom); Munich, Germany: IEEE; 2016. p. 1-3.
28. Topol EJ. Money back guarantees for non-reproducible results? BMJ. 2016;353:i2770.
29. Xia Q, Sifah EB, Asamoah KO, Gao J, Du X, Guizani M. MeDShare: Trust-less medical data sharing among cloud service providers via blockchain. IEEE Access. 2017;5:14757-67.
30. Vian K, Voto A, Haynes-Sanstead K. A Blockchain Profile for Medicaid Applicants and Recipients. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.
31. Ekblaw A, Azaria A, Halamka JD, Lippman A. A Case Study for Blockchain in Healthcare:"MedRec" prototype for electronic health records and medical research data. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.
32. Azaria A, Ekblaw A, Vieira T, Lippman A. MedRec: Using Blockchain for Medical Data Access and Permission Management. International Conference on Open and Big Data (OBD); Vienna, Austria: IEEE; 2016. p. 25-30.
33. Taylor P. Applying blockchain technology to medicine traceability. Available from: https://www.securingindustry.com/pharmaceuticals/applying-blockchain-technology-to-medicine-traceability/s40/a2766/#.WFnPQ7GZNzg. 2016 (Accessed July 5, 2019)
34. Linn LA, Koo MB. Blockchain For Health Data and Its Potential Use in Health IT and Health Care Related Research. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.
35. C B, Kalis B, Mitchell E, Pupo E, Truscott A. Blockchain: Securing a New Health Interoperability Experience. ONC/NIST Use of Blockchain for Healthcare and Research Workshop; Gaithersburg, Maryland, United States 2016.
36. Risius M, Spohrer K. A Blockchain Research Framework. Business & Information Systems Engineering. 2017:1-25.
37. Yli-Huumo J, Ko D, Choi S, Park S, Smolander K. Where Is Current Research on Blockchain Technology?—A Systematic Review. PloS one. 2016;11(10):e0163477.
38. Kuo T-T, Zavaleta Rojas H, Ohno-Machado L. Comparison of blockchain platforms: a systematic review and healthcare examples. Journal of the American Medical Informatics Association. 2019;26(5):462-78. Epub March 25, 2019. doi: 10.1093/jamia/ocy185. PubMed PMID: 30907419.
39. Key Metrics of Blockchain Platforms.
40. Solidity. Available from: https://solidity.readthedocs.io/en/develop/. (Accessed July 5, 2019)
41. The Solidity Contract-Oriented Programming Language. Available from: https://github.com/ethereum/solidity. (Accessed July 3, 2019)
42. Macdonald M, Liu-Thorrold L, Julien R. The Blockchain: A Comparison of Platforms and Their Uses Beyond Bitcoin. Available from: https://www.researchgate.net/publication/313249614_The_Blockchain_A_Comparison_of_Platforms_and_Their_Uses_Beyond_Bitcoin. 2017 (Accessed July 5, 2019)
43. TheEthereumFoundation. The Ethereum Project. . Available from: https://www.ethereum.org. (Accessed July 3, 2019)
44. The MIT License. Available from: https://opensource.org/licenses/MIT. (Accessed July 5, 2019)
45. GNU General Public License. Available from: https://www.gnu.org/licenses/gpl-3.0.en.html. (Accessed July 5, 2019)
46. Delmolino K, Arnett M, Kosba A, Miller A, Shi E, editors. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. International Conference on Financial Cryptography and Data Security; 2016: Springer.
47. CoinMarketCap. Crypto-Currency Market Capitalizations. Available from: https://coinmarketcap.com. (Accessed July 5, 2019)

48. Castillo Md. Microsoft Doubles Down on Ethereum With New Blockchain Product. Available from: http://www.coindesk.com/microsoft-launching-new-ethereum-blockchain-product/. (Accessed July 5, 2019)
49. TheEthereumCommunity. Account Types, Gas, and Transactions. Available from: http://ethdocs.org/en/latest/contracts-and-transactions/account-types-gas-and-transactions.html?highlight=gas. (Accessed July 5, 2019)
50. Ether. Available from: http://ethdocs.org/en/latest/ether.html. (Accessed July 5, 2019)
51. TheLinuxFoundation. Hyperledger Architecture, Volume I: Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus. Available from: https://www.hyperledger.org/wp-content/uploads/2017/08/HyperLedger_Arch_WG_Paper_1_Consensus.pdf. (Accessed July 5, 2019)
52. TheLinuxFoundation. Hyperledger Architecture, Volume II: Smart Contracts. Available from: https://www.hyperledger.org/wp-content/uploads/2018/04/Hyperledger_Arch_WG_Paper_2_SmartContracts.pdf. (Accessed July 5, 2019)
53. TheLinuxFoundation. Hyperledger. Available from: https://www.hyperledger.org/. (Accessed July 5, 2019)
54. Smith B, Christidis K. IBM Blockchain: An Enterprise Deployment of a Distributed Consensus-based Transaction Log.
55. Zupan N, Zhang K, Jacobsen H-A, editors. Hyperpubsub: a decentralized, permissioned, publish/subscribe service using blockchains. Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos; 2017: ACM.
56. Greenspan G. MultiChain Private Blockchain - White Paper. Available from: http://www.multichain.com/download/MultiChain-White-Paper.pdf. 2015 (Accessed July 5, 2019)
57. CoinSciencesLtd. MultiChain open platform for blockchain applications. Available from: http://www.multichain.com. (Accessed July 5, 2019)
58. Greenspan G. Smart contract showdown: Hyperledger Fabric vs MultiChain vs Ethereum vs Corda. Available from: https://www.multichain.com/blog/2018/12/smart-contract-showdown/. 2018 (Accessed July 5, 2019)
59. MultiChain Data Stream for shared immutable key–value and time series databases. Available from: https://www.multichain.com/developers/data-streams/. (Accessed July 5, 2019)
60. Google. Google Cloud Platform. . Available from: https://cloud.google.com. (Accessed July 5, 2019)
61. Official Go implementation of the Ethereum protocol Available from: https://github.com/ethereum/go-ethereum. (Accessed July 5, 2019)
62. Merkel D. Docker: lightweight linux containers for consistent development and deployment. Linux Journal. 2014;2014(239):2.
63. Ethereum Programming Languages. Available from: https://github.com/ethereum/wiki/wiki/Programming-languages-intro. (Accessed July 5, 2019)
64. CoinSciencesLtd. Working with Smart Filters. Available from: https://www.multichain.com/developers/smart-filters/. (Accessed July 5, 2019)
65. Ethereum C++ client. Available from: https://github.com/ethereum/cpp-ethereum. (Accessed July 5, 2019)
66. Next generation cryptocurrency network. Available from: https://github.com/ethereum/pyethereum. (Accessed July 5, 2019)
67. Java implementation of the Ethereum yellowpaper. Available from: https://github.com/ethereum/ethereumj. (Accessed July 5, 2019)
68. Fast, light, robust Ethereum implementation. Available from: https://github.com/paritytech/parity. (Accessed July 5, 2019)
69. Hyperledger Project; 2017. Available from: https://github.com/hyperledger. (Accessed July 5, 2019)
70. Source code for multichaind, multichain-cli and multichain-uti. Available from: https://github.com/MultiChain/multichain. (Accessed July 5, 2019)
71. McFarlane C, Beer M, Brown J, Prendergast N. Patientory: A Healthcare Peer-to-Peer EMR Storage Network v1. 2017.
72. Patientory. Available from: https://patientory.com/technology/. (Accessed July 3, 2019)
73. Genomics N. Nebula Genomics Website. Available from: https://www.nebula.org/. (Accessed July 3, 2019)
74. Genomics N. Nebula Genomics Whitepaper. Available from: https://nebulas.io/docs/NebulasTechnicalWhitepaper.pdf. (Accessed July 3, 2019)
75. Zhang P, White J, Schmidt DC, Lenz G, Rosenbloom ST. Fhirchain: applying blockchain to securely and scalably share clinical data. Computational and structural biotechnology journal. 2018;16:267-78.

76. Griggs KN, Ossipova O, Kohlios CP, Baccarini AN, Howson EA, Hayajneh T. Healthcare Blockchain System Using Smart Contracts for Secure Automated Remote Patient Monitoring. Journal of medical systems. 2018;42(7):130.

77. Dubovitskaya A, Xu Z, Ryu S, Schumacher M, Wang F, editors. Secure and trustable electronic medical records sharing using blockchain. AMIA Annual Symposium Proceedings; 2017: American Medical Informatics Association.

78. Choudhury O, Sarker H, Rudolph N, Foreman M, Fay N, Dhuliawala M, Sylla I, Fairoza N, Das AK. Enforcing Human Subject Regulations using Blockchain and Smart Contracts. Blockchain in Healthcare Today. 2018.

79. Liang X, Zhao J, Shetty S, Liu J, Li D. Integrating blockchain for data sharing and collaboration in mobile healthcare applications. Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on: IEEE; 2017. p. 1-5.

80. Kovach A, Ronai G. MyMEDIS: a new medical data storage and access system2018.

81. Corda: An Open Source Blockchain Platform for Businesses. Available from: https://www.corda.net. (Accessed July 3, 2019)

82. Brown RG. The Corda Platform: An Introduction. Retrieved. 2018;27:2018.

83. Brown RG, Carlyle J, Grigg I, Hearn M. Corda: an introduction. R3 CEV, August. 2016.