Grenze International Journal of Engineering and Technology, June Issue

BHAS: A Blockchain-Based Healthcare Analytic System to Manage Privacy and Scalability Concerns

Sonali Rokade[1] Dr.Nilamadhab Mishra[2] and Naresh Kamble[3]
[1-3]School of Computer Science and Engineering, VIT Bhopal University Kothrikalan, Sehore, Madhya Pradesh India
E mail: rokade.shashikant2019@vitbhopal.ac.in, nilamadhab.mishra@vitbhopal.ac.in, naresh.kamble2019@vitbhopal.ac.in

***Abstract*— Keeping patient records with diagnostics details is the main aim of digitalized healthcare but health data in traditional storage systems will always attract a risk to data security. There is a high risk of data leak if adequate measurement is not obtained, intruders can modify or delete any of the records. Blockchain is a stable technology that will be helpful in the development of a secure yet tamper-resistant system for the management of data.**
However, setting up an all-new system based on blockchain that will replace the existing system is quite expensive and sometimes time-consuming. But to get an ultimate and temper- proof system in the proposed architecture, introduce a smart contract with a layered approached integration mechanism, named the blockchain PHR (Patient Health Recorder), Migrating the existing cloud-basis system of health records to a public yet secure blockchain infrastructure to develop a tamper-resistant managing system of the health record.

***Index Terms*— Blockchain in Healthcare, Blockchain PHR, Tamper-Proof record management, secured health record, Blockchain health Systems security.**

I. INTRODUCTION

Implementation of Blockchain-based Patient Health Recorder (PHR) Systems are known for doing an essential jobin refining healthcare intellect, quality, security, and privacy.PHR system could eventually create a robust and efficient way of arranging and recording PHR. Sharing of healthcare information will help the health of the public and disorders tomake sure of a superior standard of services to accommodate an efficient, better understanding of patterns andtrends in better recommendations for doctors, For the comfort ofargument, the use of healthcare information is done for the representation of patient data, and healthcare information systems to provide any organization that does the generation,storage and gives access to the patient's information.
The environment of Cloud computation provides a commendable opportunity to fit in Electronic Health facilities in various situations. The environment based on the cloud offersadvantages at its core features: adaptability and portability but the problems that must be managed. An Electronic HealthRecord (PHR) management system based on the cloud provides two main pros: 1) the sharing of patient records and The ability to integrate all the PHRs of a class of hospitals to make things easy for medical professionals tomanage the activities. [3]
Though PHR has a lot of benefits the PHR systems based on the cloud have security as a critical concern. PHR which is a system based on the cloud can be exposed to abuse, leakage of data, and tampering with data. [2,3] For example, There is a high risk of data leak if adequate measurement is not obtained resulting in intruders can modify or delete any of the records or specific patients' records, to give benefits to the companies that give insurance or to covermedical wrongdoings (e.g. incorrect diagnosis and late diagnosis, use of wrong treatment or overcharging

Grenze ID: 01.GIJET.9.2.152

and surgeries that weren't needed, etc.).
Medical misconducts such as misdiagnosis and wrong practice in treating patients etc are the key factors some of the covers are taken to benefit doctors or the claims of medical insurance. In some cases, the proof of medical malpractices can not be given by the patients because some problems may be a lack of medical terms and procedure information or even patients not having adequate recordsto check with. Many security solutions are put forwardto provide certainty of PHRs by the means of many cryptographic methods. Unfortunately, due to the centralization of the characterization of systems based on the cloud,there are still security threats. Figure 1, the presentation of some of the risks of security in the conventional healthcare system that are based on the cloud.
As a blockchain is decentralized in nature and uses the decentralized ledger to store data or information it has an advantage over manipulating data.[13] As these ledgers are immutable and cannot be edited once a block is created and transacted. When we use, a blockchain network tostore PHR, which is a distributed ledger that keeps and transacts the records of healthcare so that the stokeholdscan share and exchange the data. In cloud-based Electronic Health systems, the data is acquired from different places such as hospitals, independent OPD records, clinics, and pathology labs. In a decentralized blockchain-based system, the data concerning the patients in decentralized storage in the distributed ledger that the blockchain network offers. Transactions are the set of data that needs to proceed into system. Every transaction is further assessed by agents that are also called miners before it is put in storage in the distributed ledger. Blockchain networks can reject anunauthorized transaction that can try modifying or updating
of the information in the system. Due to this, only authorized system users can modify or make changes in the data hosted on the blockchain network. A trustless system used in the blockchain is the key concept that gives smart contracts and power of being controlled deployment. A computer system that holds a set of agreements and instructions for ablock is always included in a smart contract. It is bounded to all participants of the blocked network to follow instructions and functions in the smart contract. Therefore, for the storage of data in the blockchain, no trusted third party is needed.

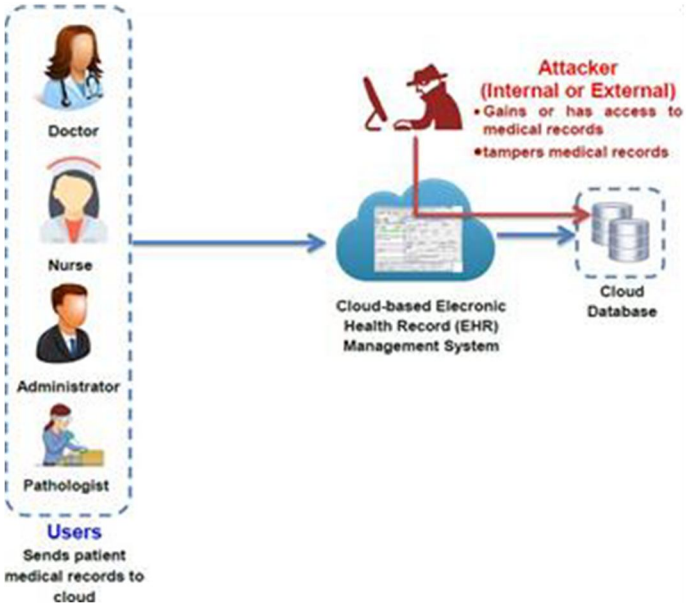

Figure 1: Security Risks posed in Cloud-based PHR

There are some challenges to obtaining blockchain in the system that is based on the cloud, the blockchain acceptance must eradicate a dominant authority's control of the data storage. In simple terms, the decentralization of data should be done as much as possible. Due to this, in the blockchain network, it becomes difficult to meddle in the data. Then, it is significantto opt for a suitable blockchain network for the PHR systemsof management. On a traditional platform installing a Blockchain-based system is quite a hard job as so many strings are in between. So, it is necessary now to develop theentire PHR system newly using Block chained based systems[3,14,15]
The task of designing and developing such a system fromthe scratch is a big task and its very time-consuming, and expensive. As a summary, in this proposed system we will answer the questions given below that related to developing blockchain-based PHR systems:
1) How to implement a blockchain-based system over anexisting system based on the cloud without affecting it?

2) How to create a decentralized blockchain applicationand choose an appropriate blockchain network?

The proposed system describes blockchain-based system architecture flow to design and development of a tamper-resistant and secure PHR system of management. In the blockchain,Three kinds of platforms exist, consortium blockchain, private blockchain, and public blockchain. Consensus is the essential service in a blocked system and a public blockchain,any participating node can join in the consensus process. As a result, the process is directly under the control of the blockchainnetwork rather than any public blockchain user – a person oran organization, keeping control decentralized. Bitcoin, Ethereum, Litecoin, etc. are some examples of the public blockchain

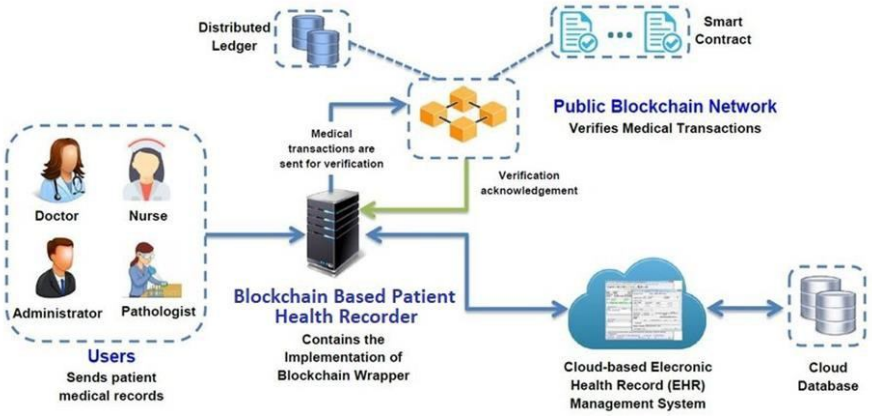

Figure 2: Proposed System Architecture

This system uses Ethereum public blockchain network tobuild a proposed PHR system. Participating in the blockchain network and the traditional PHR systems that are cloud-based is a design challenge. There is a need to design an effective methodology that is suitable to be selectedwhich will allow the blockchain to integrate the changes andmake sure that current stakeholders are also not influenced.

A bottom-up approach for the integration of blockchain to the PHR management systems that are based on the cloud is used in this proposed system. Using Ethereum blockchain uses to manage all the records, which is a public blockchain network, the following are the major contributions ofour work:

for showing the viability of the integration of blockchain into the conventional PHR systems of management that are based onthe cloud.

1) A tamper-proof and hack-proof PHR management system in blockchain technology.
2) The proposition of an innovative structuring for the integration of the established PHR management systems thatare based on the cloud to a blockchain-based system.
3) A blockchain wrapper for aiding the integration of blockchain introduces the concept of a Patient Health Record that works as
4) Discussion on the use of components in the architecture that is proposed.
5) The feasibility of the blockchain-based PHR over the existing Cloud Based system is shown in the demo application

II. BLOCKCHAIN-BASED PHR

Patient Health Record (PHR) is the key component of ourproposed structure. This constituent performs the function ofa cover part that connects user applications, a traditional PHR system that is cloud-based, and a blockchain network in this suggested architecture. PHR has three main parts, transactiontemplate manager (TTM)[3,13,12,14], transaction generator (TG), and transaction validator (TV). The internal structure of PHR is shown in Figure 3. The following is the information on each component of PHR:

- *Transaction template manager (TTM):* has a set of structured templates of transactions for the blockchain network. Using the architecture and schema of the blockedenvironment administrator creating a transaction template with all possible attributes mentioned,
- *Transaction generator (TG):* This part builds blockchaintransaction (TC) from an initial transaction (TI) using atemplate in TTM. The mapping between TI and a suitable template in TTM is done by TG.

- *Transaction validator (TV):* The important and main part of the PHR which then controls the entire interactions with blocks of PHR and handshaking user applications, cloud, and blockchain network. TV adapts an initial transaction TI from the application, forwardsit to TG, and waits to receive a blockchain transaction TC from TG. Upon receiving a TC, the TV forwards it to the blockchain network for confirmation. The blockchain network validates transaction T' If the transactional information is in the database. According touser access and request, the cloud responds with appropriate data.

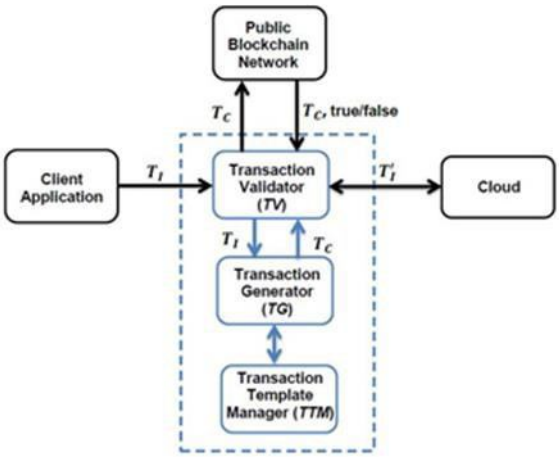

Figure 3: Internal working of System

III. SYSTEM WORKFLOW

Figure 4 shows how the components of the system interact with one another. with the system workflow of the blockchain PHR management system. Originally, the user application sends an initial transaction (TI) containing true validation, ACK is forwarded as a valid transaction tothe patient health data inserted by a user. Next, TI is sent to Blockchain Handshaker (PHR) for communicating with the public blockchain net-cloud to be stored in the storage of thecloud. Otherwise, ACK is forwarded as an invalid transaction and is kept for future auditing tasks.

A. Public Blockchain Network aggregator

Ethereum is used as a public Blockchain aggregator the blockchain network includes nodes, smart contracts, and distributed ledger. For the maintenance of the blockchain, asstated by the mechanism of consensus, Blockchain nodes, i.e.miners that are responsible based on smart contracts, receivethe transactions and validation takes place. If the validation of a transaction is successful, the data is transformed into blocks and put in the distributed ledger. An acknowledgment is sent by the public blockchain network stating whether it is true or false to the transaction validator (TV) of the Patient Health Record.

B. Cloud system

For hosting proposed system as a PHR system usescloud services integrated with Ethereum to facilitate all hosting and integration services required by the PHR management system which then receives transactions T from PHR, performs all tasks concerning it, and saves [21] work.
blockchain transaction (TC) generated by PHR using theinternal parts called the transaction generator (TG) and the transaction template manager (TTM). A significantcomponent of PHR is, the transaction validator (TV), which then forwards TC to get validation from the public blockchain network. Additionally, the validation of these transactions is done by a public blockchain networkwith the help of intelligent contracts, and then the addition of transaction data into the blockchain.
The public blockchain network sends an acknowledgment of validity to PHR at the end of the process. The transaction T that has completed validation is then sent to the cloud for further processing. Every record related to patient health is passed to Patient Health Record(PHR) for validation.The creation and publication of smart contracts have beendone on the public blockchain. Every transaction is validated randomly on the network once it's published. Thetransaction date is stored in the blockchain or represented on a distributed ledger. As the anonymity of blockchain nodes is guaranteed, likely, theycannot be compromised. Secure mining of blocks is

guaranteed by the Proof-of-Work (Pow) mechanism.Hence, the suggested system structure ensures a tamper-resistant data ledger.

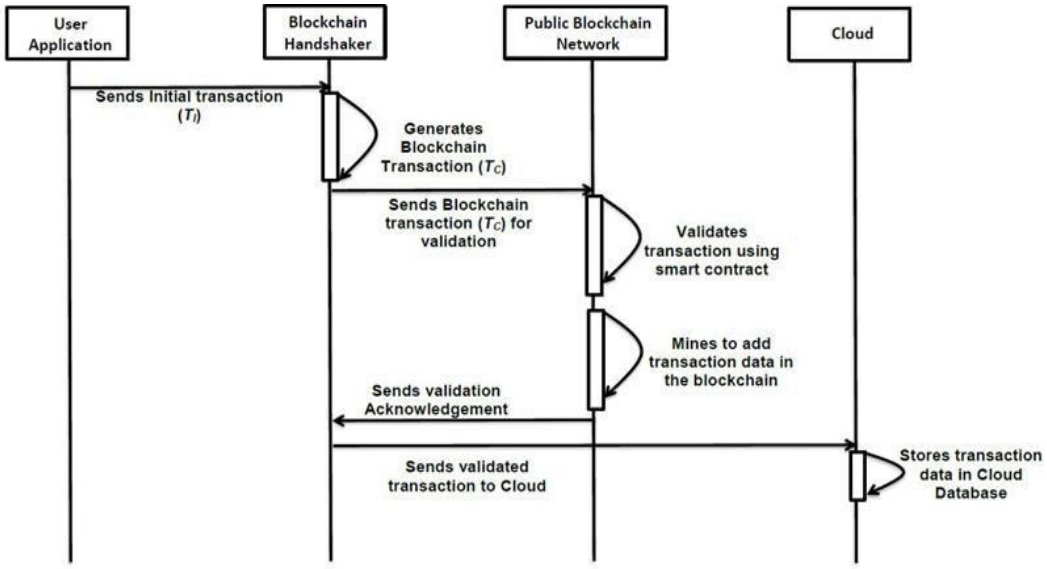

Figure 4: Communication Stack with all involvedcomponents in purposed Blockchain-based PHR.

The validation of which transaction is faulty and which is not is done with the use of ACK along with the transaction data. The tracking and tracing of data modification can be done from this. Hence, the accountability of data is assured. Since tamper-resistant data storage and responsibility of data is assured by the system, the system can be called an immutable system.

IV. IMPLEMENTATION

We consider this system to be implemented, where hospital or clinic staff store patient health information in a given framework and application in performing the patient health assessment. Upon the arrival of a new patient, their profile is created. All of the patient health conditions are entered by the assigned staff, once the patient is treated or undergoing treatment. Comments are sent to Patient Health Record (PHR).
The comments are converted by PHR into blockchain transactions (TC) and sent for validation to the public blockchain network. The comments are added as blocks in the blockchain. The comments are sent to the system that is based on the cloud after they are added to the blockchain for further processing.

A. Implementation Details

Once the system is developed and ready, it has three components: application, Patient Health Recorder, and PHR management system based on the cloud. The system uses Ethereum as the public blockchain platform. A local version is used by us. of the Ethereum blockchain, called Ganache, in place of the live blockchain. The system uses Windows 10 for the configuration of Ganache. NodeJS is used for the development of our blockchain wrapper and network node manager, i.e. Patient Health Record (PHR). To interact with PHR and Ganache, the system uses web3js and API with node IDs, is are instructions collected to interact with Ganache. The web3js uses the connections, HTTP or IPC in communicating with Ganache. Our smart contracts have been written using solidity design which is meant to support the Ethereum platform.
The writing and compilation of the smart contracts have been done using Remix and Atom react, an Interactive Development Environment (IDE) used as an editor for solidity, which helps further to write and compile codes and event scripts for smart contracts. For simulating the Ethereum network, and integrating it with the browser to give the client environment for the test cases system use Metamask which is a browser extension installed on Mozilla Firefox and Google Chrome be enabled in the client system to help communicate with the Ethereum platform.
The system proposed to develop client applications using HTML5 and Bootstrap backed with solidity. The PHR management system based on the cloud will be developed with the use of CMS and hosts on IPFS. The system will use MySQL as our cloud database. Figure 5 shows the all-around work pattern of the proposed system.

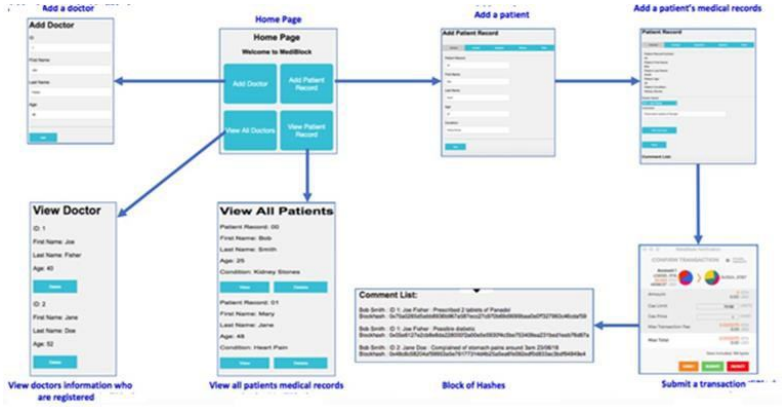

Figure 5: System Flow and modules of the prototype developed

V. Pseudocode Of The Algorithm And Framework

Table 1 illustrates the notations used in the algorithms and Algorithm represents the algorithm to create and manage PHR in this system. In the depicted framework there are a total 4 stakeholders in which D represents Doctor, N for Nurse, A for Administrator, and Ph for Pathologist.

This algorithm considers there are n participants for each stakeholder in the framework. The Hyperledger Caliper and CA provide public key cert to all participants such as Doctors, Nurses, Administrators, and pathologists. There is a key pair for each participant

- DPk and DPr as the public and private keys of Doctor Di,
- NPk and NPr as the public and private keys of Nurse Ni,
- APk and APr as the public and private keys of the Administrator Ai
- PhPk and PhPr as the public and private keys of the Pathologist Phi

TABLE I: NOTATIONS USED IN THE ALGORITHM

Notations	**Definition**
P_{Cv}	Composite data view
S_k	Session Key
N_{Pk}	Public Key for Nurse
N_{Pr}	Private Key for Nurse
A_{Pk}	Public Key for Administrator
A_{Pr}	Private Key for Administrator
P_{PHR}	Patient's Health Record
D_{Pk}	Doctors' Public Key
D_{Pr}	Doctors' Private Key
Ph_{Pk}	Pathologists' Public Key
Ph_{Pr}	Pathologists' Private Key
D_i	Doctor
N_i	Nurse
A_i	Administrator
Ph_i	Pathologist
UP_{Cv}	Updated Composite data view
UP_{PHR}	Updated Patients' Health Records

respectively where i = 1 to n. This illustration provides a detailed explanation of how the Doctor and Nurse interact for accessing health records in the blockchain-based framework.

Consider that Doctor Di grants access to medical record PPHR to Nurse Ni upon request based on access control permissions as shown in the Internal working of the System. After that, the framework then creates a composite view PCv of the patient record PPHR that can be accessible for Nurse Ni upon request consecutively for sharing the medical record of the patient.
The Composite view PCv is the only feature set of the stored PPHR that the framework generates on authenticated and verified user requests without actually sharing the whole PPHR.
Restricted access to the data is protected by the composite view of a PPHR so that only authenticated users can see and modify only selected data that the user needs and are not allowed to modify any further entries. In other words, PCv is a subset of PPHR

The Algorithm To create and manage PHR this system

Procedure System(): Create and update composite view of Medical Records

Input: A Nurse N_i with public key N_{pk} and session key S_k to access medical record P_{PHR}
Output: Creation and updation of the medical record

1: **for** each user U with access permission to P_{PHR}
2: Algorithm checks framework access permission rules to grant or deny access to the requested user.
3: **if** (permission type == "ALLOW" && role Type == 'Nurse')
4: Create composite view P_{Cv} of the medical record P_{PHR} in the framework
5: $P_{Cv} \rightarrow \int_{i=1}^{n} (DD_{Pr}(ED_{Pk}(P_{PHR})))$
6: P_{Cv} Generate a session key S_k
7: $P_i \leftarrow ED_{Pk}(S_k)$ /*Send encrypted session key to Doctor
8: $N_i \leftarrow EN_{Pk}(S_k)$ /*Send encrypted session key to Nurse
9: $N_i \leftarrow ES_k(P_{Cv})$ /*Send encrypted composite view to Nurse
10: $P_{PHR} \leftarrow (DD_{Pr}(ED_{Pr}(P_{PHR})))$ Call for Nurse record access and update
11: $UP_{Cv} \leftarrow (DS_k(ES_k(UP_{Cv})))$
12: $UP_{PHR} \leftarrow [(EP_{pk}(P_{PHR})) + (EP_{pk}(UP_{Cv}))]$ /*Framework commits the updated record
13: **return** #
14: **else**
15: access ← deny
16: **return** access

Algorithm 1 To create and manage the PHR system

The framework further generates a session key Sk which is shared between Nurse and the Doctor for a certain session. The framework then sends the encrypted session key Sk to the Doctor as EDpk(Sk) and Nurse as ENpk (Sk) by encrypting the respective public keys of the Doctor Dpk and Nurse Npk for a separate session in Algorithm 1. The Composite view PCv will also be encrypted with session key Sk as ESk(PCv) and stored in System.
In addition to security, the framework will send encrypted composite view ESk(PCv) to the Nurse. Here the Algorithm calls for nurses to update their PPHR. In this process Nurse decrypts the session key with the associated private key and decrypts the composite view with the session key as per the algorithm step 2 and step 3.
In the event there are any updates, the Nurse or Doctor updates PCv as UPCv , to determinations of the case, Further the same is encrypted with the session key and uploads UPCv to Blockchain as ESk(UPCv)
Thus the framework refers to the client-side application. The Doctor uses a passphrase to encrypt the private key Dpr and saves it on the client side. Each time for convenience, the Doctor can run this passphrase which decrypts the private key instead of sharing or uploading the private key, as well as user framework can use this private key to decrypt the PPHR.
To Update Doctors' or Nurse's record update function calls an Algorithm where the framework decrypts the encrypted record i.e. EDpk(PPHR) using Doctors' private key and decrypts the encrypted updated composite view from the Blockchain i.e. ESk(UPCv) using the session key as per steps 12 and 13. And the same procedure is repeated for the Nurse or any other participants, Finally, the Doctor can commit the updates to the existing record and encrypts the record PPHR as EDpk (PPHR) before uploading back to Blockchain.

VI. Testing And Analysis Of The System

Test of a decentralized application (also a proposed system with various self-governing parts) is a difficult job in several aspects: to introduce components and initialize them on various machines is time-consuming and attention-seeking and migration efforts; as proposed a private blockchain functionality to deploy and operate on secure infrastructure than having a traditional approach to a public blockchain, during the execution, the main communication links can return faulty or no connection or break as multiple factors are getting involved; different nodes can present failures during a sequence of operations, and after the failure of something,
it becomes important to look into the entire process history as well as the transactions with every possible event on which test sensation is applied and the interaction of all the connected components, nodes, and minor transactions. In particular, testing the performance and as well as how secure the blockchain network is poses problems as the proposed research is still discussing the representative indicators and the testing tools are still in incubation.
The proposed system for security is based on modeling, trust assumption, and assessments as the experimental validation which is not always comprehensive. On the other hand, the analysis of the performance brings up all-new challenges since it is vital to write tests that stretch to every element and reproduces a realistic scenario wherein every component is put under pressure. The following values suggest a discussion of approaches, parameters, and tools to test the performance of the proposed PHRs. Followed by the result of the test analysis is presented.

A. Hyper ledger Caliper

Hype rledger Caliper is a blockchain benchmark tool, it allows users to measure the performance of a blockchain implementation with a set of predefined use cases. Hyperledger Caliper will produce reports containing several performance indicators to serve as a reference when using the following blockchain solutions: Hyperledger Beau, Hyperledger Burrow, Ethereum, Hyperledger Fabric, FISCO BCOS, Hyperledger Iroha and Hyperledger Sawtooth.[58,24,14,5]

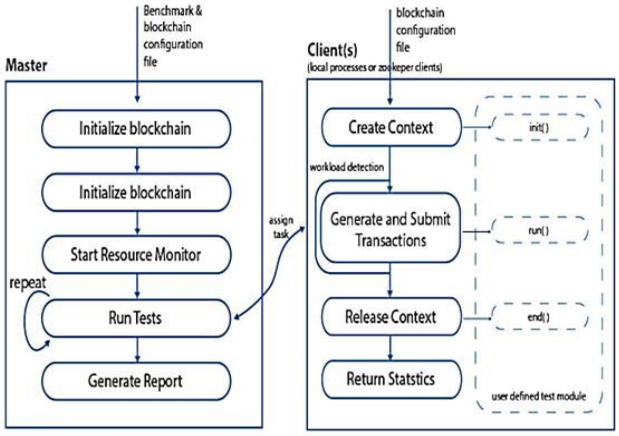

Figure 6: Flow initialization of Hyperledger caliper used in the developed system

Hyper ledger Caliper also attempts to deal with performance indicators parameters which are defined and considering these meanings have been provided by the Performance & Scalability Working Group (PSWG)
The orchestration of a pool of clients that communicate invoke and query the blockchain is implemented for the particular use case by the caliper architecture.

- *Adaptation Layer*: This contains various sets of connectors to interface with a particular use case.
- *Interface and Core layer*: The monitoring of the resources, examining the performances, invoking and fitting smart contracts on the blockchain, and generating the final report is done by this layer which is designed by various modules that allow these functions.
- *Application/Benchmark Layer*: The scripts used by the user in the testing of a specific blockchain use case. The test suites that are used to measure the performances are formed by these scripts. A standard engine that carries out the exams using a master-workers strategy as shown in the figure, the master node generates worker threads to which it assigns jobs, is in this layer. Each job is responsible for one function on the blockchain, which can be either smart contract arrangement, carrying out, or clean up.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates to and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

B. Test environment and results

The test was executed on 5 different virtual machines with configuration as follows with categorization:
Node machines Operating system: Windows Server 2019, Fedora, Ubuntu, Unix, Windows 10 Ultimate:

- CPU: Virtual 8 core 3.24 MHz each machine;
- RAM: 16GB of Virtual DDR 4 Each machine.
- Network Bandwidth: 100/1000 GBPS per NIC
- Form factor: Virtual grid i64 bit
- Blockchain: Ethereum Private Blockchain
- Frame work: Truffle Suite Ganache
- Node form factor: NodeJS 64 bit
- Integration agent: Metamask
- API/ABI support: Metalmark and Truffle
- IDE Environment: Ethereum Remix-based Solidity
- Hosting service: IPFS Network
- Compiler: Ethereum Remix 10.10 commit
- Minor: Classic minor generation algorithm randomized 1000 minor’s pool

The following is the HL Fabric network setup:

- HL Fabric version v2.2.x;
- The peer’s storage is the default MariaDB and MySQL database;
- The basis of the ordering service is Kafka, Zookeeper servers, and Kafka brokers
  The produced final report contains different performance indicators:
- Success rate;
- Transaction commit and state read Throughput (TPS);
- Transaction commit and read latency;
- Resource consumption (CPU, Memory, and network IO).

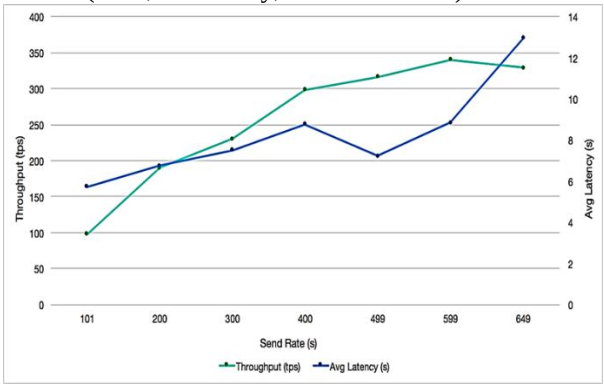

Figure 7: Aggregated HL Caliper reports

TABLE II: AGGREGATED HL CALIPER REPORTS, WRITESOPERATIONS, THE NORMAL SCENARIO

Success	Fail	SendRate (tps)	Maximum Latency (s)	Minimum Latency (s)	Average Latency (s)	75%ile Latency (s)	Throughput (tps)
4000	0	101	8,4	0,36	5,74	7,23	98
4000	0	200	8,42	1,11	6,75	7,72	190
4000	0	300	9,66	3,95	7,51	8,76	230
4000	0	400	12,1	0,92	8,77	9,41	298
4000	0	499	10,06	1,03	7,23	8,56	316
4000	0	599	21,78	0,98	8,85	12,23	340
4000	0	649	40,99	1,11	12,96	17,32	329

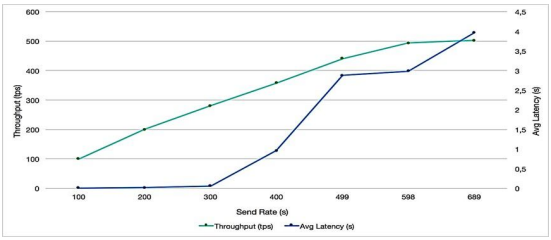

Figure 8: Throughputs and latencies

TABLE III : THROUGHPUTS AND LATENCIES, WITH A DIFFERENT TRANSACTION, SEND RATES.

Succ.	Fail	Send Rate (tps)	Maximum Latency (s)	Minimum Latency (s)	Average Latency (s)	75%ile Latency (s)	Throughput (tps)
5000	0	100	0,1	0,01	0,01	0,01	100
5000	0	200	0,16	0,01	0,02	0,06	200
5000	0	300	0,63	0,01	0,06	0,2	280
5000	0	400	3,18	0,2	0,96	1,36	358
5000	0	499	5,18	0,11	2,88	3,82	441
5000	0	598	5,87	0,13	2,99	4,03	493

VII. COMPARISON WITH THE EXISTING SYSTEM

This section will show the effectiveness of the proposed system over existing cloud-based electronic data recorder systems. These test cases are mainly around essential yet important parameters such as

- Execution time taken
- Data Transferred and records
- Security obtained

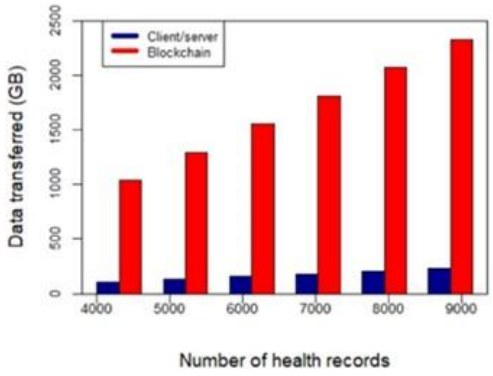

Figure 9: Data Transferred for records on client/server and blockchain with incremental numbers of records.

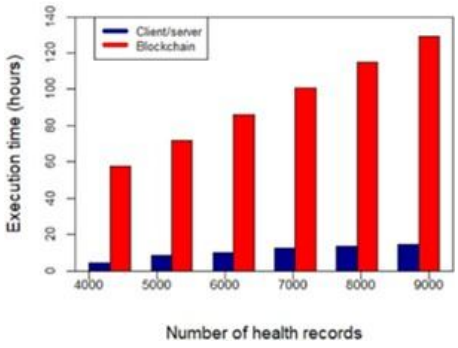

Figure 10: Execution time taken for Data Transferred for records on client/server and blockchain with incremental numbers of records.

TABLE IV. EXECUTION TIME FOR CLIENT/SERVER AND BLOCKCHAIN WITH INCREASING HEALTH RECORDS AND HOSPITALS.

		Average Increase in Execution Time (Hours)			
		Health Records Update		Health Records Query	
Increasing variable	Variable increasing factor	Client/server	Blockchain	Client/server	Blockchain
Number of health records	+1000	2.03	14.36	2.03	0.14
Number of hospitals	+10	4.42	57.44	4.42	0.58

TABLE V. AMOUNT OF DATA TRANSFERRED FOR CLIENT / SERVER AND BLOCKCHAIN WITH INCREASING HEALTH RECORDS AND HOSPITALS.

		Average Increase in Data Transfer (GB)			
		Health Records Update		Health Records Query	
Increasing variable	Variable increasing factor	Client/server	Blockchain	Client/server	Blockchain
Number of health records	+1000	25.86	258.61	25.86	28.18
Number of hospitals	+10	0	1034.98	0	10.34

IX. CONCLUSIONS AND FUTURE WORK

The suggested system is a systematic method of a secure and tamper-resistant, secure, privacy reach and immutable Patient Health Recorder (PHR) System using Ethereum as a public blockchain Network compared to the traditional systems, our approach gives a more effective instrument that uses an extraction service layer that called "Patient Health Recorder".

The current methods try to provide answers that are blockchains-based from the beginning that are feasible and inexpensive compared with procedures taken by a traditional system which quite need drastic changes. By using independent components participating in business logic and application logic this kind of ambiguity is eliminated.

REFERENCES

[1] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., Caro, A.D., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolic, M., Cocco, S., & Yellick, J. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. Proceedings of the Thirteenth EuroSys Conference.

[2] Jabarulla, M. Y., & Lee, H.-N. (2020). Blockchain-Based Distributed Patient-Centric Image Management System. Applied Sciences, 11(1), 196. doi:10.3390/app11010196

[3] Heart, T., Ben-Assuli, O., & Shabtai, I. (2017). A review of PHR, EMR and EHR integration: A more personalized healthcare and public health policy. Health policy and technology, 6, 20-25.

[4] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (updated on 22 November 2020).

[5] Benet, J. (2014). IPFS - Content Addressed, Versioned, P2P File System. ArXiv, abs/1407.3561.

[6] Roehrs, A., da Costa, C. A., da Rosa Righi, R., da Silva, V. F., Goldim, J. R., & Schmidt, D. C. (2019). Analyzing the performance of a blockchain-based personal health record implementation. Journal of biomedical informatics,92, 103140.https://doi.org/10.1016/j.jbi.2019.103140

[7] Pranjul Yadav, Michael Steinbach, Vipin Kumar, and Gyorgy Simon. (2018). Mining Electronic Health Records (EHRs): A Survey. ACM Comput. Surv.50, 6, Article 85 (January 2018), 40 pages.DOI:https://doi.org/10.1145/3127881

[8] Abdullah Al Omar, Mohammad Shahriar Rahman, Anirban Basu, and Shinsaku Kiyomoto. (2021). for healthcare data. In International conference on security, privacy, and anonymity in computation, communication, and storage. Springer, 53–543 the Year 2021

[9] Ismail, L., Materwala, H., & Zeadally, S. (2019). Lightweight Blockchain for Healthcare. IEEE Access, 7, 149935-149951.

[10] Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). MedRec: Using Blockchain for Medical Data Access and Permission Management. 2016 2nd International Conference on Open and Big Data (OBD), 25-30.

[11] Dagher, G.G., Mohler, J., Milojkovic, M., & Marella, P. (2018). Ancile: Privacy-Preserving Framework for Access Control and Interoperability of Electronic Health Records Using Blockchain Technology. Sustainable Cities and Society, 39, 283-297.

[12] Li, H., Zhu, L., Shen, M., Gao, F., Tao, X., & Liu, S. (2018). Blockchain-Based Data Preservation System for Medical Data. Journal of Medical Systems, 42, 1-13.

[13] Fan, K., Wang, S., Ren, Y., Li, H., & Yang, Y. (2018). MedBlock: Efficient and Secure Medical Data Sharing Via Blockchain. Journal of Medical Systems, 42, 1-11.
[14] Dey, T., Jaiswal, S., Sunderkrishnan, S., & Katre, N. (2017). HealthSense: A medical use case of the Internet of Things and blockchain. 2017 International Conference on Intelligent Sustainable Systems (ICISS), 486-491.
[15] Yue, X., Wang, H., Jin, D., Li, M., & Jiang, W. (2016). Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control. Journal of Medical Systems, 40, 1-8.
[16] Uddin, M., Stranieri, A., Gondal, I., & Balasubramanian, V. (2018). Continuous Patient Monitoring With a Patient-Centric Agent: A Block Architecture. IEEE Access, 6, 32700-32726.
[17] Hölbl, M., Kompara, M., Kamišalić, A., & Zlatolas, L.N. (2018). A Systematic Review of the Use of Blockchain in Healthcare. Symmetry, 10, 470.
[18] Abbas, A., & Khan, S. (2014). A Review of the State-of-the-Art Privacy-Preserving Approaches in the e-Health Clouds. IEEE Journal of Biomedical and Health Informatics, 18, 1431-1441.
[19] Omar, A., Rahman, M.S., Basu, A., & Kiyomoto, S. (2017). MediBchain: A Blockchain-Based Privacy Preserving Platform for Healthcare Data. SpaCCS Workshops.
[20] Anderson, N. R., Lee, E. S., Brockenbrough, J. S., Minie, M. E., Fuller, S., Brinkley, J., & Tarczy-Hornoch, P. (2007). Issues in biomedical research data management and analysis: needs and barriers. Journal of the American Medical Informatics Association:JAMIA, 14(4), 478–488. https://doi.org/10.1197/jamia.M2114
[21] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. 2016. Medrec: Using blockchain for medical data access and permission management. In 2016 2nd International Conference on Open and Big Data (OBD). IEEE, 25–30.
[22] Vitalik Buterin. 2015. On public and private blockchains. https://blog.ethereum. org/2015/08/07/on-public-and-private- blockchains/.
[23] Yu-Yi Chen, Jun-Chao Lu, and Jinn-Ke Jan. (2012). A secure HER Qi Xia, Emmanuel Boateng Sifah, Abla Smahi, Sandro Amofa, and Xiaosong Zhang. 2017. BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments. Information 8, 2 (2017), 44.