Lekana - Blockchain Based Archive Storage for Large-scale Cloud Systems

Eranga Bandara[1], Xueping Liang[2], Sachin Shetty[3], Wee Keong Ng[4], Peter Foytik[3], Nalin Ranasinghe[5], Kasun De Zoysa[5], Bård Langöy[1], and David Larsson[1]

[1] Pagero AB, Gothenburg Sweden
{eranga.herath, bard.langoy, david.larsson}@pagero.com
[2] Virginia State University, Virginia USA
{xliang}@vsu.edu
[3] Old Dominion University, Virginia USA
{sshetty, pfoytik}@odu.edu
[4] School of Computer Science and Engineering
Nanyang Technological University, Singapore
{awkng}@ntu.edu.sg
[5] University of Colombo School of Computing, Sri Lanka
{dnr, kasun}@ucsc.cmb.ac.lk

Abstract. Blockchain is a form of a distributed storage system that stores a chronological sequence of transactions in a tamper-evident manner. Due to the decentralized trust ecosystem in blockchain, various industries have adopted blockchain to build their applications. This paper presents a novel approach to building a blockchain-based document archive storage platform, "Lekana". The Lekana platform can be adopted by service providers that require frequent document operations, such as "Pageroonline", a cloud-based e-invoicing provider in Europe. With Lekana we introduce a novel approach to store an immutable hash chain of archived data which are owned by the customers in the blockchain. The proposed Lekana platform is built on top of Mystiko which is a highly scalable blockchain storage platform targeted for big data. We have integrated real-time data analytics and machine learning techniques into the Lekana platform by using the Mystiko-Ml machine learning service on Mystiko blockchain. By integrating the Lekana platform with blockchain technology, we have addressed major issues in most cloud-based, centralized storage platforms(e.g. lack of data privacy, lack of data immutability, lack of traceability and lack of data provenance). As a case study in the paper, we present how "Pageroonline" cloud-based e-invoicing provider stores their archived document data and archived document hash chain in the blockchain-based Lekana platform.

1 Introduction

1.1 Cloud storage

There are various types of cloud-based storage platforms available, such as Amazon S3 [1], Google cloud platform [2] and Azure cloud archive [3]. Most of these storage services are governed by a central authority. Centrally controlled services tend to lack privacy, traceability, immutability, or data provenance features. Due to these reasons, data fraud and attacks are easier to accomplish and can happen more

frequently. Unauthorized third parties such as hackers or employees of the cloud service company may access the data and alter them. Once data fraud happens, it's hard to trace and identify the attackers. To address these issues on centralized cloud storage systems, system designers have integrated decentralized blockchain platform due to its decentralized and immutable ecosystem.

1.2 Blockchain

Blockchain provides a tamper-evident, shared digital ledger that records data in a public or private peer-to-peer network in the form of a distributed peer-to-peer storage. Each node in the blockchain has the same order of data, which is immutable. Since blockchain is a distributed storage, it needs a consensus algorithm to order and maintain consistency of data among the nodes. Currently, there are various blockchain platforms in the market. Bitcoin [4], Ethereum [5], Bigchaindb [6], Hyperledger [7], Mystiko [8] are some examples. Some of these blockchains are mostly used for electronic currencies such as Bitcoin. Ethereum and Hyperledger blockchains go beyond crypto-currency to support different types of asset storage models that relate to other forms of business or e-commerce activities. Mytiko, Bigchaindb blockchains are targeted for big data applications.

Novel blockchain platforms introduce a programming interface referred to as "smart contracts" to interact with the blockchain ledger. The smart contracts interpose additional software layers between the clients and the blockchain storage. Client requests are directed to scripts called smart contracts that perform the logic needed to provide a complex service such as managing state, enforcing governance, or checking credentials. With smart contracts, developers do not need to execute queries to save or retrieve data from blockchain storage. Instead, smart contracts provide a programming interface to interact with the underlying blockchain storage models. The existing blockchain systems come with different smart contract platforms. For example, Ethereum has the Solidity platform [9], and Hyperledger Fabric has the Chaincode platform [7]. Kadena has Pact [10], and RChain has Rholang [11]. Mystiko comes with a concurrency enabled Aplos smart contract platform [8, 12], which is adopted in this paper.

1.3 Lekana

This paper introduces a novel approach to build the blockchain-based document archive platform, Lekana, with the ability to address the previously mentioned issues on cloud storage systems and support data provenance in the cloud. As a case study, this platform is built on Pageroonline which is a cloud-based e-invoicing platform [13]. Pageroonilne is one of the largest e-invoicing providers in the world. It accumulates over one millions electronic documents per day. Pageroonline keeps documents for at most two years. After two years, the document data is archived.

Lekana platform is built to store Pagerooinline's archive documents information and their payloads(pdf, xml, image byte streams). All two-year-old documents, and their payloads are saved in the Lekana archive storage and its integrity is protected in the off-chain storage of the blockchain.

The Lekana platform is built on top of Mystiko blockchain, which is highly scalable in terms of storage capability. Mytiko blockchain maintains a hash chain of the archived documents. Actual archived document information and payloads(pdf, xml, image byte streams) are stored in an Apache Cassandra [14] off-chain storage platform. When the document is archived, the payloads are saved in off-chain storage, and the hash chain will be updated on Mystiko providing an immutable record ensuring the integrity of the payload. Archive document hash chain creation, retrieval and validation functionalities are implemented with Aplos smart contracts on Mystiko blockchain. Mytiko blockchain comes with the Mystiko-Ml machine learning service, which is integrated with Apache Spark [15]. Lekana platform is integrated with the Mystiko-Ml service to do real-time data analytics, machine learning, and visualization with the data on the blockchain.

In this paper, a performance evaluation of the underlying blockchain storage in Lekana platform is presented. The evaluation shows the scalability and transaction throughput features in Lekana platform when using different blockchain systems. Following are the main contributions from Lekana.

1. Blockchain based document archive storage platform, with the ability to address the common issues in cloud based data storage(lack of data privacy, lack of data immutability, lack of traceability, lack of data provenance).
2. Blockchain has been used to store an immutable hash chain of archiving documents which are owned by the customers.
3. To address the privacy concerns in the blockchain, off-chain storage has been integrated to the blockchain.
4. As a case study, discussion about integrating Lekana platform to build production grade archive storage service in Pageroonline [13], one of the largest cloud-based e-invoicing platform in the world.
5. Introduced a mechanism to build machine learning models(e.g isolation forest [16] unsupervised machine learning algorithm based anomaly detection model) with the data on blockchain and off-chain storage in the Lekana platform.

1.4 Outline

The paper is organized as follows. Section II introduces the architecture of Lekana platform. Section III illustrates the Lekana platform implementation and functionality. Section IV presents the performance evaluation of the Lekana platform. Section V introduces the related work. Section VI is the conclusion and some future work of the Lekana platform.

2 Lekana Architecture

2.1 Overview

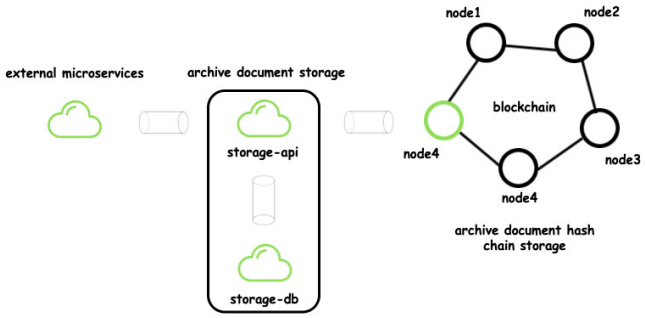

Fig. 1: Lekana architecture. Archive document information stored in storage-db. Archive document hash chain data stored in blockchain.

Lekana is a blockchain-based document archive service. All the archive document information, document payloads, and document hash chain information are maintained in Lekana, which provides APIs to create archive documents, search archive documents and validate hash chain functions. Figure 1 shows the architecture of the Lekana platform. There are two main components in the Lekana platform.

1. Archive document storage (off-chain storage which stored all archive document information and document payloads)
2. Document hash chain (Blockchain storage which stored hash chain of the archive documents)

2.2 Archive document storage

All archive document information and document payloads will be stored in archive document storage. Archive document storage comes with two main components "storage-api" and "storage-db". The archive document information and payloads will be stored in the storage-db(off-chain storage of blockchain) in Lekana platform. It is an off-chain storage built on top of a distributed database. Archive document storage-api exposes API's to create, search and validate the hash chain of the archive documents. External services interact with the API to create, search and validate archive documents. To create archive document, it will send an archive document and create request to storage-api with archive document meta data information and payloads. When the archive document create request is received, it will

save the archive document information and payloads in the storage-db(distributed database). Then it will interact with blockchain (e.g blockchain smart contract) to create the hash chain record for the archive document. All archived document hash chain records are stored in the blockchain.

2.3 Document hash chain

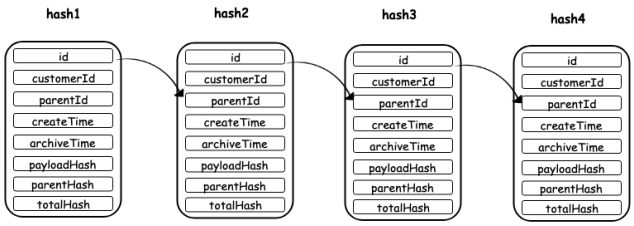

Fig. 2: Lekana archive document hash chain of the customer which stored in blockchain. One hash record refers to its parent hash record.

Lekana platform uses the blockchain to keep the archive document hash records. It gives a decentralized mechanism to check immutability of archive documents. After archiving, the record should be immutable, changes to the recorded documents should not be made, only new records added. By saving archive document hashes in blockchain, it gives a way to track the updates to the document. Blockchain keeps the document hash as a blockchain asset. Figure 4 shows the structure of the asset. The document hash record keeps a reference to its parent document hash, Figure 2. These hashes are stored per user in the blockchain. When a new hash record is created, it searches for the last hash record which corresponds to a given user. Then it adds that hash record id as the `parentHash`. Finally it calculates the total hash based on `parentHash`, `createTime`, `archiveTime` and `payloadHash`. To calculate the total hash, it concatenates `parentHash`, `createTime`, `archiveTime`, `payloadHash` and gets the `SHA256` hash of it. In this way, we build customers' archive document hash chain on top of the blockchain. Each hash record refers to its parent hash record.

The hash chain based archive is designed for integrity such that it is not possible to remove a single document from the archive without breaking a cryptographic chain. The integrity check on the archive hash chain will reveal such an action. The hash chain data in the blockchain can be exposed to outside parties(e.g. customers in "Pageroonline") to verify the integrity of the documents. For example, "Pageroonline" customers can fetch their archive document data from storage-api and hash chain data from the blockchain. Then they can validate the hash chain

with the actual archive document data and check the integrity of the archive documents. By storing document data in the archive-storage(off-chain storage) and hash chain data on the blockchain Lekana has addressed the common issues in cloud based data storage, lack of data privacy, lack of data immutability, lack of traceability, lack of data provenance.

3 Lekana functionality

3.1 Overview

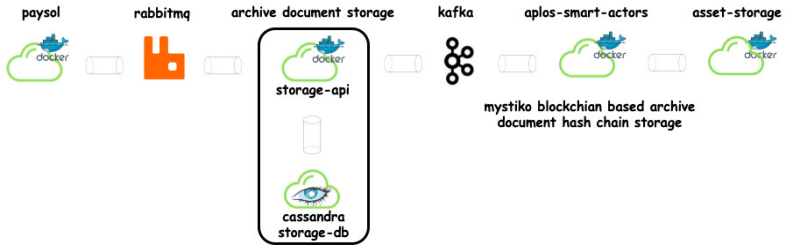

Fig. 3: Lekana architecture in Pageroonline cloud platform. Apache cassandra based storage-db used as the off-chain storage. Mystiko blockchian used to build the document hash chain.

As a use case demonstration, Lekana platform is built on Pageroonline which is a cloud-based e-invoicing platform [13]. Pageroonilne accumulates over one million electronic documents per day. Pageroonline keeps these documents for at most two years. After two years, the document data is archived. Lekana platform is built to store Pagerooinline's archive document information and their payloads(pdf, xml, image byte streams). All two-year-old documents and their payloads saved in the Lekana archive storage are protected in the off-chain storage of the blockchain. The architecture of the Lekana platform implementation of Pageroonline is shown in Figure 3.

We have used Apache Cassandra based distributed database as the storage-db in Pageroonlines' Lekana platform. All archive document information is stored in Cassandra database tables. External microservices(e.g. "Paysol") interacts with storage-api service to create, search, and validate the archive documents. Pageroonline uses Rabbitmq [17] as the microservice message broker. The service communication protocols is defined with Google Protobuf messages [18]. Protobuf API is exposed with storage-api to create, search, and validate the hash chain of the archive documents. External microservices in pageroonline interact with the API through Rabbitmq to create, search, and validate archive documents. For example,

the microservice named “Paysol” sends an archive document create Protobuf message(with archive document information, payloads) to storage-api. Then it will save archive document information and payloads in the Apache Cassandra based storage-api(off-chain storage). After that it interacts with Mystiko’s blockchain smart contract and create the hash chain record for the archive document. Mystiko blockchain exposes Apache Kafka [19] based asynchronous API to communicate with smart contracts.

The Lekana platform document hash chain is built on top of Mystiko blockchain, which is highly scalable in terms of storage capability. Mystiko blockchain is built with microservice based distributed system architecture [20]. All services are dockerized [21] and available for deployment using Kubernetes [22]. Since Pageroonlines services are also built as a microservice architecture, by deploying with docker/kubernetes we are able to easily integrate Mystiko blockchain with Pageroonline platform. In Lekana, Mytiko blockchain maintains the hash chain of the archived documents. This hash chain is exposed to customers in Pageroonline to verify the integrity of the documents and guarantee the data provenance without revealing actual document payloads or metadata information in the archive storage.

3.2 Hash chain functions

```
{
  "id": "<hash id>",
  "customerId": "<document owner customer id>",
  "parentId": "<last archive document id of the customer>",
  "createTime": "<document create time>",
  "archiveTime": "<document archive time>",
  "payloadHash": "<document payload hash>",
  "parentHash": "<parent document hash>",
  "totalHash": "<total hash>"
}
```

Fig. 4: Lekana Hash assert structure in Mystiko blockchain.

```
{
  "id": "<transaction id>",
  "execer": "<transaction executing user>",
  "messageType": "create",
  "customerId": "<document owner customer id>",
  "documentId": "<document id>",
  "parentId": "<last archive document id of the customer>",
  "createTime": "<document create time>",
  "archiveTime": "<document archive time>",
  "docHash": "<document payload hash>",
  "digsig": "<digital signature of the message>"
}
```

Fig. 5: Lekana CreateHash message in Mystiko blockchain.

Hash chain create, validate and search functions are implemented as Smart contracts. Smart contracts are built using the Scala functional programming language [23–25] based Akka actors [26–28] in Mystiko blockchain. Aplos Smart Actors in Mystiko blockchain consume transaction messages via Kafka message broker [19] with a reactive streams approach [29, 30]. There is a smart actor named `HashChainActor` which manages the hash chain functions. This actor defines three smart contract functions, `CreateHash`, `ValidateHashChain`, `SearchHashChain`. These functions are invoked through the transaction messages which come to the `HashChainActor` via Apache Kafka.

`CreateHash` corresponds to the creating of a new hash chain record for the customer and document. Transaction messages correspond with `CreateHash` de-

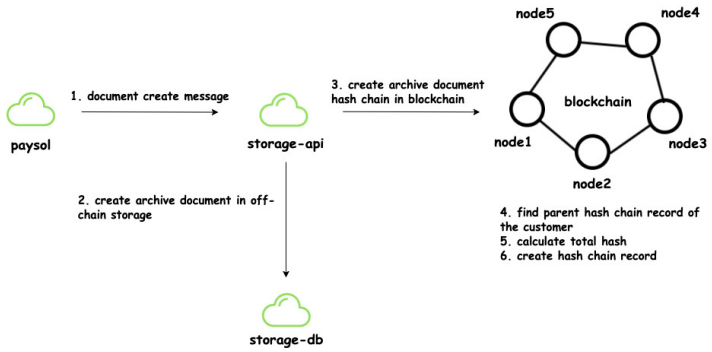

Fig. 6: Lekana create archive document flow. First create archive document in off-chain storage. Then add hash chain record in the blockchain.

scribed in Figure 5. When this function is invoked, it first finds the last hash chain record corresponding to the given customer(parent hash), then calculates the total hash with parent hash to create a new hash record in the Mystiko storage(ledger), Figure 6. `ValidateHashChain` to do the hash chain validation of a given customer. It retrieves the hash chain of a given customer with `fromDate` to `toDate` and validates against the actual document payloads stored in document storage service. If any alteration is made to the document payload, it can be detected in `ValidateHashChain` function, Figure 7. `SearchHashChain` facilitates the Hash chain search function. It searches the hash chain records of the given customer with `fromDate` to `toDate`. `SearchHashChain` and `ValidateHashChain` functions are exposed to third party customers via HTTP REST API. By using this API any customer can verify their archive documents status and hash chain, such as whether hash chain and documents have been altered or not.

3.3 Analytics and machine learning

Mystiko blockchain keeps all transactions, blocks, asset information on Apache Cassandra based Elassandra Storage [31]. It exposes Apache Lucene index [32] based Elasticsearch APIs [33] for transactions, blocks and assets on the blockchain. We have integrated Kibana analytic dashboards [34] with Elasticsearch API to visualize real-time and historical data on Mystiko blockchain. Mentioned above, Mystiko blockchain comes with Mystiko-Ml, an Apache Spark-based machine learning and analytics service. It establishes supervised or unsupervised machine learning models with the existing data on Mystiko Cassandra storage(both on-chain and off-chain). These models can be used to do predictions of real-time data. We have integrated Mystiko-Ml service into the Lekana platform to build an isolation forest unsupervised [16] model with the off-chain storage data.

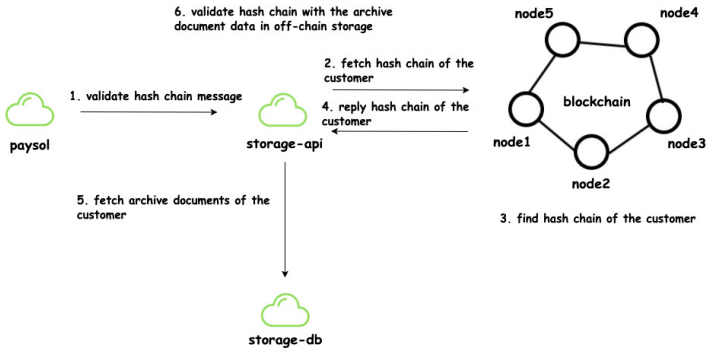

Fig. 7: Lekana validate customer hash chain flow. First fetch hash chain of the customer from the blockchain. Then fetch archive document data from the off-chain storage. Finally validate hash chain with the archive document data in off-chain storage.

4 Performance Evaluation

Performance is evaluated for the underlying blockchain storage in Lekana platform. The evaluation shows the scalability and transaction throughput features in Lekana platform when using different blockchain storages. To obtain the results, we have deployed Lekana blockchain on multi-node Mystiko cluster(4 nodes) and multi-node Hypeledger fabric cluster. Hyperledger fabric runs with a Kafka based consensus with 3 Orderer nodes, 4 Kafka nodes, 3 Zookeeper nodes and LevelDB [7] as the state database. Mystiko blockchain runs with 4 Kafka nodes, 3 Zookeeper nodes and Apache Cassandra [14] as the state database. We use Apache Cassandra [14] as the storage-db(off-chain storage) in the archive document storage service. The evaluation results are obtained based on the following five metrics.

1. Invoke Transaction throughput
2. Query Transaction throughput
3. Transaction scalability
4. Transaction latency
5. Search performance

4.1 Invoke transaction throughput

For this evaluation, we recorded the number of invoke transactions that can be executed in the underlying blockchain ledger. Invoke transaction creates a record in the ledger and updates the status of the assets. We flood invoke transactions for each blockchain peer and recorded the number of executed transactions. As

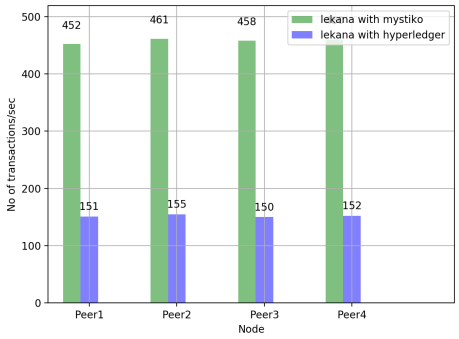

Fig. 8: Invoke transaction throughput of Lekana platform in different blockchain ledgers.

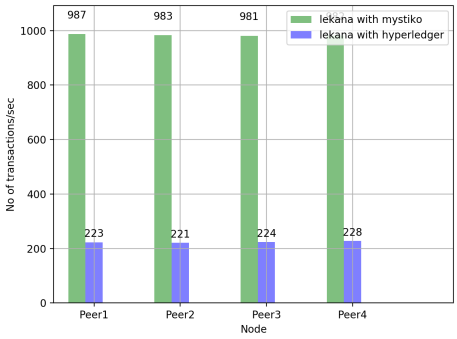

Fig. 9: Queary transaction throughput of Lekana platform in different blockchain ledgers.

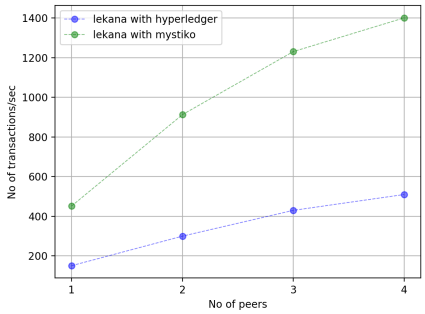

Fig. 10: Transaction scalability of Lekana platform in different blockchain ledgers.

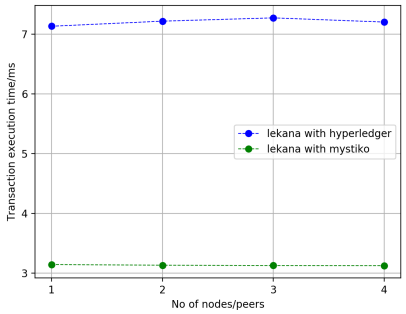

Fig. 11: Transaction latency of Lekana platform in different blockchain ledgers.

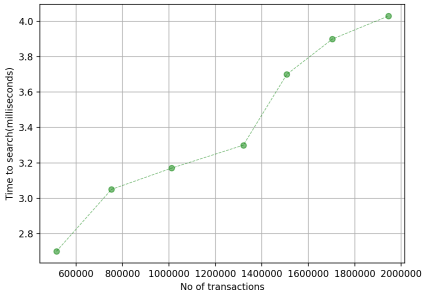

Fig. 12: Search performance of Lekana storage.

shown in Figure 8, we have obtained consistent transaction throughput in different blockchain peers in Lekana platform.

4.2 Query transaction throughput

For this evaluation, we recorded the number of query transactions that can be executed in the underlying blockchain ledger of the Lekana platform. Query transactions just query the status from the ledger. They neither create a transaction in the ledger nor update the asset status. We flooded query transactions for each blockchain peer and recorded the number of completed transactions. As shown in Figure 9, we compare the query transaction throughput of Lekana platform in different blockchain environments. Since query transactions are not updating the ledger status, it has higher throughput compared to invoke transactions. For example, search transaction throughput of Mystiko ledger-based Lekana platform is 2 times higher than invoke transaction throughput, and search transaction throughput of Hyperledger Fabric ledger-based Lekana platform is 1.5 times higher than invoke throughput.

4.3 Transaction scalability

For this evaluation, we record the number of invoke transactions (per second) over the number of blockchain peers in the Lekana platform. We flood concurrent transactions in each blockchain peer and record the number of executed transactions. Figure 10 shows transaction scalability results for the Lekana platform. When adding blockchain nodes to the cluster it linearly increase the transaction throughput.

4.4 Transaction latency

Next, we evaluate the transaction latency of the underlying blockchain ledger in the Lekana platform. We flood concurrent transactions in each blockchain peer and calculate the average transaction latency. Figure 11 shows the transaction latency results of Lekana platform in different blockchain environments. When adding nodes to the cluster a consistent latency is maintained in Lekana platform.

4.5 Search performance

Finally, we evaluate the search performance of the Lekana storage service. Lekana platform provides ability to search data in its storage using Elasticsearch [33]. For this evaluation, we issue concurrent search queries to the Lekana platform and compute the search time. As shown in Figure 12, to search 2 million records, it takes only 4 milliseconds. The Apache Lucene index-based Elasticsearch storage is the main reason yielding a fast search in the Lekana platform.

5 Related Work

Table 1: Blockchain based storage platform comparison

Platform	Implemented Blockchain	Implemented architecture	Consensus	Scalability	Smart contract support	Deduplication support	Off-chain support	Full-text search support
Lekana	Mystiko	Microservices	Paxos	High	Yes	No	Yes	Yes
Archain	Archain	Monolith	PoW	Low	No	No	Yes	No
Archangel	Ethereum	Monolith	PoS	Low	Yes	No	Yes	No
Yugala	Mystiko	Microservices	Paxos	High	Yes	Yes	Yes	Yes
Sia	Sia	Monolith	PoW	Low	Yes	No	Yes	No
Filecoin	IPFS/Filecoin	Monolith	PoS/PoR	Mid	Yes	Yes	No	No
Storj	Ethereum	Monolith	PoS	Low	Yes	Yes	No	No
Swarm	Ethereum	Monolith	PoS	Low	Yes	No	No	No

Existing research has been conducted to build decentralized storage systems on top of the blockchain. In this section, we outline the main features and architecture of these research projects.

Archain [35] is a blockchain-based document archive system, developed for the state archive-keeping committee of the Republic of Tatarstan (Russia). It keeps document information in blockchain transactions. Each accepted document corresponds to one transaction record in the blockchain. The system can be described as an interaction of participants of three roles: Administrator, Expert, and User. Roles are selected and assigned to members through the Certification Authority. Users create and upload documents to the network. Administrators select an expert for each of the created documents and add them to the archive after the expert's approval. Experts make decisions on documents – if the document is improperly formalized, has some metadata missing, or doesn't comply with local legislation, then it should be denied from transferring to the archive.

Archangel [36] is a blockchain-based decentralized platform for ensuring the long-term integrity of digital documents stored within public archives. It uses blockchain as a basis for ensuring the provenance and integrity of documents during the process of preserving the records (curation) and upon release (presentation). Archangel uses the Ethereum blockchain to record digital signatures(derived from either scanned digital or born-digital archival images) of the documents. The process works by creating a hash of the original digital document, recording that on the blockchain, preserving that document in the archives, and then subsequently checking that the document has not been altered by comparing the hash of the preserved document with the hash originally recorded on the blockchain.

Yugala [37] Yugala is encrypted cloud storage for IoT data/Big data. It proposes a blockchain-based lightweight, encrypted cloud storage architecture, which maintains file confidentiality. Yugala removes the traditional centralized data dedu-

plication and increases file integrity by using a decentralized blockchain. In particular, it discusses two approaches for file confidentiality with data deduplication: one uses double hashing and the other symmetric encryption. Yugala storage is built on top Mystiko blockchain. All data deduplication handling functions(with double hashing and symmetric encryption) have been implemented with Aplos smart contract on Mystiko blockchain.

Sia [38] is a decentralized cloud storage platform that intends to compete with existing storage solutions, at both the P2P and enterprise level. Instead of renting storage from a centralized provider, peers on Sia rent storage from each other. Sia itself stores only the storage contracts formed between parties, defining the terms of their arrangement. These smart contracts are stored in their public blockchain systems like bitcoin. They have a preference for the Proof-of-Work(PoW) consensus and use ASIC chips for Siacoin mining. Their Proof of Storage algorithm is utilized to further protect and validate proofs and file contracts on the network.

Filecoin [39] is an open-source project designed to create a permanent, decentralized method of data storage and sharing. It is an advanced IPFS [40] version with a blockchain incentive mechanism and even the off-chain trading market for file storage. The network provides a decentralized hub on which people who have excess storage capacity can offer it to those in need of capacity. Individuals and businesses pay to store data on the storage provider's hardware. Filecoin utilizes a proof of storage concept (similar to PoW) to determine if a miner has conducted his storing/retrieving duties. The concept has two elements: Proof-of-replication and Proof-of-spacetime. Proof-of-replication is used by storage providers to show that they have stored a unique set of data to the space it owns. Proof-of-spacetime enables storage providers to show that data has been stored over a specific period. This is done by requiring a storage provider to show sequential instances of Proof-of-replication.

Storj [41] is an open-source platform that leverages Ethereum blockchain to provide end-to-end encrypted cloud storage services. Instead of maintaining its own data centers, the Storj platform relies on a peer-to-peer network of individuals or entities sharing their storage space. Their technology revolves file sharing, similarly to how torrents work and separate parts of the files to users in the network. When a user requests the file, Storj uses a distributed hash table to locate all the shards and pieces them together. These files are also encrypted before sharing and the person uploading it has their private key to validate ownership. Storj uses private verification, which means the data owner is supposed to do the auditing job here with pre-generated nonces/salts and the classic Hash(block, nonce/salt). Unlike Filecoin and Sia, Storej does not support smart contracts on the blockchain that set the rules and requirements for storage. Instead, Storj users pay for what they use.

Swarm [42] is a distributed storage platform and content distribution service on the Ethereum web3 stack. The primary objective of Swarm is to provide a decentralized and redundant store for dapp code and data as well as blockchain and state data. Swarm is also set out to provide various base layer services for web3, including node-to-node messaging, media streaming, decentralized database services and scalable state-channel infrastructure for decentralized service economies. It splits the data into blocks called chunks, which have a maximum size limit of 4K bytes and distribute among the nodes. It provides ENS(Ethereum Name System), which is implemented as a smart contract on the Ethereum network. It can be considered as the equivalent of the domain name service (DNS) that facilitates content naming in traditional internet services.

The comparison summary of these storage platforms and the Lekana platform is presented in Table 1. It compares Running blockchain platform, Implementation architecture, Consensus, Scalability, Smart contract support, Deduplication support, Off-chain storage support, Full-text search support details.

6 Conclusions and Future works

With Lekana we have introduced a blockchain-based, scalable, decentralized document archive storage platform. Lekana has addressed the common issues in centralized cloud-based storage platforms(e.g. lack of data privacy, lack of data immutability, lack of traceability) while supporting data provenance in the cloud. By using Mystiko blockchain to build the Lekana platform we were able to align the Lekana platform with high transaction load in Pageroonline. We have presented the scalability and transaction throughput features of the platform with empirical evaluations.

We also introduced a blockchain-based novel approach to keeping an immutable hash chain of archiving data for customers. This chain can be accessed by outside parties(e.g. customers in Pageroonline) to verify the integrity of the documents without revealing actual document payloads or metadata information. Most recently we have integrated Lekana version 1.0 with pagerooinline cloud platform, following the agile continuous delivery approach when building and releasing the product every month.

Acknowledgements

This work was funded by the Department of Energy (DOE) Office of Fossil Energy (FE) (Federal Grant #DE-FE0031744).

References

1. “Amzon s3.” [Online]. Available: https://aws.amazon.com/s3/

2. S. Krishnan and J. L. U. Gonzalez, *Building Your Next Big Thing with Google Cloud Platform: A Guide for Developers and Enterprise Architects*. Springer, 2015.
3. R. Jennings, *Cloud computing with the Windows Azure platform*. John Wiley & Sons, 2010.
4. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
5. V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.
6. T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: a scalable blockchain database," *white paper, BigChainDB*, 2016.
7. E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. ACM, 2018, p. 30.
8. E. Bandara, W. K. NG, K. DE Zoysa, N. Fernando, S. Tharaka, P. Maurakirinathan, and N. Jayasuriya, "Mystiko—blockchain meets big data," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 3024–3032.
9. "Solidity." [Online]. Available: https://solidity.readthedocs.io/en/develop/
10. S. Popejoy, "The pact smart contract language," *June-2017.[Online]. Available: http://kadena.io/docs/Kadena-PactWhitepaper. pdf*, 2016.
11. E. Eykholt, L. G. Meredith, and J. Denman, "Rchain architecture documentation," 2017.
12. E. Bandara, W. K. NG, K. De Zoysa, and N. Ranasinghe, "Aplos: Smart contracts made smart," *BlockSys'2019*, 2019.
13. "Pageroonline." [Online]. Available: https://www.pagero.com/about-pagero/
14. A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2010.
15. X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen *et al.*, "Mllib: Machine learning in apache spark," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1235–1241, 2016.
16. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.
17. A. Videla and J. J. Williams, *RabbitMQ in action: distributed messaging for everyone*. Manning, 2012.
18. K. Varda, "Protocol buffers: Google's data interchange format," *Google Open Source Blog, Available at least as early as Jul*, vol. 72, 2008.
19. J. Kreps, N. Narkhede, J. Rao *et al.*, "Kafka: A distributed messaging system for log processing," in *Proceedings of the NetDB*, 2011, pp. 1–7.
20. J. Thönes, "Microservices," *IEEE software*, vol. 32, no. 1, pp. 116–116, 2015.
21. "Docker documentation," Aug 2018. [Online]. Available: https://docs.docker.com/
22. "Kubernetes documentation." [Online]. Available: https://kubernetes.io/docs/home/?path=users&persona=app-developer&level=foundational
23. M. Odersky, P. Altherr, V. Cremet, B. Emir, S. Maneth, S. Micheloud, N. Mihaylov, M. Schinz, E. Stenman, and M. Zenger, "An overview of the scala programming language," Tech. Rep., 2004.
24. "The scala programming language." [Online]. Available: https://www.scala-lang.org/
25. J. Hughes, "Why functional programming matters," *The computer journal*, vol. 32, no. 2, pp. 98–107, 1989.
26. "Akka documentation." [Online]. Available: https://doc.akka.io/docs/akka/2.5/actors.html
27. C. Hewitt, "Actor model of computation: scalable robust information systems," *arXiv preprint arXiv:1008.1459*, 2010.
28. C. A. R. Hoare, "Communicating sequential processes," *Communications of the ACM*, vol. 21, no. 8, pp. 666–677, 1978.
29. "Akka documentation." [Online]. Available: https://doc.akka.io/docs/akka/2.5/stream/

30. A. Destounis, G. S. Paschos, and I. Koutsopoulos, "Streaming big data meets backpressure in distributed network computation," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.
31. Strapdata, "strapdata/elassandra," Jul 2018. [Online]. Available: https://github.com/strapdata/elassandra
32. "Welcome to apache lucene." [Online]. Available: http://lucene.apache.org/
33. "Elastic stack and product documentation — elastic." [Online]. Available: https://www.elastic.co/guide/index.html
34. "Kibana product documentation." [Online]. Available: https://www.elastic.co/products/kibana
35. A. Galiev, N. Prokopyev, S. Ishmukhametov, E. Stolov, R. Latypov, and I. Vlasov, "Archain: A novel blockchain based archival system," in *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2018, pp. 84–89.
36. J. Collomosse, T. Bui, A. Brown, J. Sheridan, A. Green, M. Bell, J. Fawcett, J. Higgins, and O. Thereaux, "Archangel: Trusted archives of digital public documents," in *Proceedings of the ACM Symposium on Document Engineering 2018*. ACM, 2018, p. 31.
37. S. S. P. F. Sarada Prasad Gochhayat, Eranga Herath, "Yugala: Blockchain based encrypted cloud storage for iot data," *IEEE Blockchain*, 2019.
38. D. Vorick and L. Champine, "Sia: Simple decentralized storage," *Nebulous Inc*, 2014.
39. J. Benet and N. Greco, "Filecoin: A decentralized storage network," *Protoc. Labs*, 2018.
40. J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.
41. S. Wilkinson, J. Lowry, and T. Boshevski, "Metadisk a blockchain-based decentralized file storage application," *Tech. Rep.*, 2014.
42. J. H. Hartman, I. Murdock, and T. Spalink, "The swarm scalable storage system," in *Proceedings. 19th IEEE International Conference on Distributed Computing Systems (Cat. No. 99CB37003)*. IEEE, 1999, pp. 74–81.