Volume: 12 Issue: VI Month of publication: June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63335

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

Securing Logs and Metrics in DevOps Using Blockchain: Enhancing Trust and Integrity in Continuous Deployment

Sahil Dattatray Dhawde
Department of Information Technology, Sathaye College, Mumbai, India

Abstract: In today's constantly evolving DevOps environment, log and metric security is critical to maintaining the trust and integrity of continuous deployment pipelines. When it comes to protecting against tampering and unauthorized access of logs and metrics, traditional methods often fall short. However, traditional approaches frequently encounter challenges related to security, transparency, and reliability. Within this framework, blockchain technology presents itself as an effective way of improving the security and integrity of log and metric data within DevOps processes. This paper explores how to secure logs and metrics in DevOps using blockchain technology, specifically Ethereum with Solidity programming, to improve trust and integrity in continuous deployment processes. We talk about the planning and execution of a blockchain-based system that monitors and logs DevOps pipeline activity. Our method ensures tamper-proof storage, transparent auditing, and precise access control for log and metric data by utilizing smart contracts and decentralized ledger technology. We demonstrate a prototype implementation and evaluate how well it improves accountability, integrity, and security of data. Our research shows how blockchain technology can be used to solve important issues with DevOps logging and monitoring, creating the way to more secure and reliable continuous deployment processes.

Keywords: DevOps, Blockchain, Logs, Metrics, Ethereum.

I. INTRODUCTION

Blockchain technology and DevOps approaches are revolutionizing how organizations build, deploy, and secure software. DevOps emphasizes continuous integration and deployment (CI/CD), which allows for faster development cycles and better communication between of software development (Dev) and IT operations (Ops) teams. It aims to break down traditional techniques by promoting a culture of continuous integration, continuous delivery, and shared responsibility for both tasks. As DevOps evolves, it plays an increasingly important role in enabling technological change across businesses, making it a necessary framework for modern software development and deployment. However, securing the security and integrity of logs and metrics in these rapidly changing DevOps environments presents major issues.

Blockchain technology, with its decentralized, immutable, and transparent nature, provides a strong solution to these problems. By integrating blockchain into DevOps pipelines, organizations may greatly improve the security, transparency, and auditability of their deployment processes. Blockchain was originally designed as the foundational technology for Bitcoin, but it has since evolved to enable a wide range of uses other than cryptocurrencies. It is a decentralized ledger that stores transactions in immutable blocks, providing a secure and transparent system. Its decentralized nature eliminates the need for a single authority, allowing data verification across multiple nodes. This makes blockchain useful in applications like financial services, supply chain management, and DevOps environments. The integration of Blockchain technology in CI/CD pipelines allows organizations to securely record, audit, and protect logs and data, promoting a more trustworthy and integrated software development process.

A. This Importance of Log and Metrics

DevOps practices depend heavily on monitoring and logging, which provides several advantages that are critical for maintaining system flexibility, performance, and reliability. First, maintaining continuous service delivery depends heavily on proactive issue detection. Through monitoring, DevOps teams can keep an eye on important metrics and identify any anomalies or possible bottlenecks before they affect end users. Teams can take preventative measures to reduce risks and guarantee optimal system performance by spotting problems early on. Additionally, logging helps teams maximize productivity and efficiency by offering in-depth insights into system behaviors.

DevOps teams can enhance their application functionality, identify enhancement opportunities, and adjust code and infrastructure accordingly by inspecting logs. This preventive approach to performance optimization improves the system's overall scalability and efficiency.

Also, logs are essential for performing root cause analysis when problems do arise. Logs give teams a detailed sequence of the actions that accompanied an issue, making it easier to locate the problem's origin and implement focused solutions. This enabling faster troubleshooting and increased system dependability. Additionally, in DevOps, both logging and monitoring support data-driven decision-making. Teams can make well-informed decisions about infrastructure scaling, resource allocation, and code improvements by utilizing the insights obtained from monitoring data and logs. As a result, organizations can improve end-user results, streamline operations, and optimize their DevOps processes.

B. *Securing logging and metrics*

Safe and secure logging and metrics gathering are crucial elements in guaranteeing accountability, compliance, and transparency in DevOps procedures. With the help of secure logging, organizations can keep a detailed log of everything that happens in their environments, which promotes stakeholder trust and transparency. Secure logging and metrics gathering also makes it easier to keep up with legal requirements and industry standards, protecting the confidentiality, security, and integrity of data. Establishing strong security protocols to safeguard log and metric data enables organizations to satisfy legal requirements, develop stakeholder trust, and make well-informed decisions grounded in trustworthy and auditable data.

DevOps teams depend significantly on secure logging, monitoring, and data collection practices to achieve proactive problem detection, performance optimization, and data-driven decision-making. These procedures improve user experiences while improving the scalability, efficiency, and reliability of systems. The security and integrity of logs and metrics in DevOps can be strengthened with the help of blockchain technology. Blockchain ensures the immutability of stored data, reducing the risks of tampering or unauthorized alterations. It does this by providing a decentralized, tamper-proof ledger.

The purpose of this study is to explore how blockchain technology can be used to address security and integrity issues that are present in metrics collection and logging processes that are part of DevOps operations. This research seeks to provide light on how blockchain can improve the security, integrity, and reliability of log and metric data in DevOps workflows by examining best practices, mechanisms, and integration strategies.

II. BACKGROUND

A. *Existing System*

A range of tools, techniques, and procedures are used in current DevOps approaches to logging and metrics with the goal of collecting, analyzing, and keeping track of data produced throughout the software development and deployment lifecycle. These tools and techniques are specially designed to meet the demands and difficulties of modern software development and implementation Organizations use centralized logging platforms like Grafana Loki, Splunk, and ELK (Elasticsearch, Logstash, Kibana) Stack extensively to collect, store, and analyses logs from different parts of their DevOps infrastructure. These systems combine logs from different DevOps infrastructure components and offer centralized storage, search, visualization, and alerting features. Teams can use them to troubleshoot problems, monitor performance in real time, and gain insights into the behavior of the system.

In distributed systems and microservices architectures, conventional logging might not offer enough insight into transactions from start to finish. Organizations can track requests across several services with distributed tracing systems like Jaeger, Zipkin, and Open Telemetry, which offer insights into latency, dependencies, and error propagation. likewise, some organizations create custom logging libraries or frameworks to meet needs and specifications. With the help of these libraries, developers can gather more insightful and useful log data. They integrate with the current logging infrastructure and provide extra features like context propagation, structured logging, and log enrichment. Solutions for custom logging offer flexibility and customization to satisfy the unique logging needs of various environments and applications.

With the help of pre-built dashboards, alerting systems, and anomaly detection tools provided by platforms like Prometheus, Grafana, and Datadog are a few examples of metrics collection and monitoring tools that are essential for tracking application health and system performance metrics, which allows for proactive problem identification and resolution. While cloud providers offer managed log management solutions like AWS CloudWatch Logs, Azure Monitor, and Google Cloud Logging, cloud-native environments frequently rely on log shipping and aggregation techniques enabled by tools like Fluentd, Fluent Bit, and Filebeat.

DevOps environments can also benefit from complete monitoring, tracing, and logging solutions offered by observability platforms such as New Relic, Dynatrace, and LightStep. These platforms enable quick issue diagnosis and resolution throughout the entire stack. All things considered, current DevOps approaches to logging and metrics cover a wide range of techniques and tools designed to meet the unique requirements and difficulties of modern software development and deployment.

B. Limitations and Vulnerabilities

While there are many advantages to the current DevOps logging and metrics approaches, there are also some drawbacks and vulnerabilities that companies need to be aware of. First, as data volumes increase, centralized logging platforms and metrics collection tools may face scalability issues. In large-scale or quickly changing environments, this can make it difficult to maintain high ingestion rates, storage capacity, and query performance. Second, implementing and maintaining of these platforms may be difficult and resource-intensive, resulting in high expenditures for licensing, infrastructure provisioning, and continuing maintenance.

There is a chance of single points of failure when using centralized platforms, which can affect incident response and resolution due to errors or failures. Security flaws are a serious concern as well because they allow sensitive data to be accessed or altered by unauthorized parties due to inadequate authentication procedures or ineffective access controls. Root cause analysis and troubleshooting efforts may be impacted by traditional logging approaches' lack of insight and its context, which makes it difficult to connect related events across distributed systems or microservices architectures.

Organizations that depend on specific solutions may also experience vendor lock-in, which restricts their ability to work with other tools and environments and reduces their flexibility. Scalability, data privacy, security, resilience, and interoperability are the top priorities of DevOps logging and monitoring system to meet these challenges. Organizations can optimize the advantages of logging and metrics in their DevOps workflows and minimize risks by carefully weighing these factors and putting strong strategies into place.

C. Need of Blockchain

Blockchain technology offers a decentralized, tamper-proof ledger for storing and managing this important data, which can greatly improve the security and integrity of logs and metrics in DevOps. DevOps teams can reduce common security risks and guarantee the accuracy of the data by using blockchain to create transparent, immutable, and auditable records of log entries and metric data. A secure chain of transactions is created that cannot be changed or removed without the network participants' consensus, with each log entry or metric datapoint being cryptographically hashed and connected to previous entries.

Blockchain's immutability ensures that information entered the ledger cannot be changed or tampered with, offering a trustworthy audit trail for monitoring changes and confirming the authenticity of metrics and logs. Furthermore, by eliminating single points of failure and lowering the possibility of data manipulation or unauthorized access, blockchain's decentralized architecture improves the overall security posture of DevOps environments. Also, only authorized users can view or alter log and metric data due to blockchain technology's precise permission management and access control features. Access control policies can be implemented and data validation procedures can be automated with the help of smart contracts, which are self-executing contracts with predefined rules and conditions. By preventing unauthorized access to or disclosure of sensitive information, this improves data security and confidentiality.

Organizations can improve operational efficiency and stakeholder trust by strengthening data integrity, increasing transparency, and enhancing the security of their DevOps practices through the integration of blockchain technology into logging and metrics collection processes.

D. Blockchain Fundamental

The decentralized ledger system known as blockchain technology completely changes how data is handled and stored. In general, blockchain is defined by multiple fundamental concepts:

1) Decentralization: The core principle of blockchain technology is decentralization. Blockchain functions on a decentralized network of nodes, compared with traditional centralized systems where data is stored and managed by a single authority. Because every node in the network keeps a copy of the whole blockchain, no single entity can control the data. Decentralization lowers the possibility of manipulation or censorship and eliminates single points of failure, which improves security, resilience, and transparency.

2) *Consensus Mechanisms:* Consensus mechanisms are rules that are used in blockchain networks to achieve consensus among nodes about the validity of transactions and the sequence in which they should be added. The addition of new blocks to the blockchain is governed by a number of consensus mechanisms, including Proof of Work (PoW), and Proof of Stake (PoS), Delegated Proof of Stake (DPoS). These rules ensure that, even in the case of a lack of trust between individual nodes, all members of the network come to an agreement regarding the ledger's current state. Game theory is the real source of these consensus mechanisms.
3) *Cryptographic Hashing*: The security and integrity of data stored on the blockchain are crucial depends on cryptographic hashing. The blockchain is made up of a series of sequentially linked, immutable blocks, each of which has a cryptographic hash of the one before it. For each piece of input data, hash functions produce a unique, fixed-length hash, which makes it computationally impossible to decipher the original data from the hash. Any manipulation of the data within a block would result in a change to its hash, which would break the chain and indicate an occurrence of unauthorized modifications.
4) *Smart Contracts:* These are self-executing contracts that are directly encoded into the blockchain with predetermined rules and conditions. These contracts automatically execute and enforce the terms of agreements when predetermined conditions are met. Decentralized applications (DApps) can function independently of intermediaries because of smart contracts, which reduces costs, increases productivity, and minimizes the possibility of fraud or manipulation. Smart contracts facilitate various applications like supply chain management, tokenization, and decentralized finance (DeFi). They are usually written in programming languages like Solidity (for Ethereum).

By providing decentralization, consensus mechanisms, cryptographic hashing, and smart contracts that ensure security, integrity, and transparency across a range of applications and industries, blockchain technology, implemented as a whole, completely changes data management.

E. Blockchain Log Management in Healthcare

A study demonstrates the use of Hyperledger Fabric to build a private Blockchain network for real-time log management in the healthcare industry. A system was created that includes an interface to help with data analysis and a private blockchain network to support the recording of hospital logs. It emphasizes the importance of ensuring data integrity, privacy, and security in healthcare systems. The use of blockchain technology addresses the issue of data immutability, safeguarding patient information from privacy breaches and security threats. By creating a secure and distributed database, blockchain technology offers a solution to the challenges faced by traditional healthcare systems in maintaining the authenticity and security of patient data.

The initiative aims to improve the safety and reliability of healthcare data by utilizing blockchain's decentralized and secure features. The overall goal involved utilizing Hyperledger Fabric technology to build a permissioned private Blockchain network. The Go programming language was used to create a Chaincode that would enable hospital logs to be securely and immutably stored for further analysis and access. The architecture of the solution combined the ledger of Hyperledger Fabric and Chaincode with CouchDB for data storage, in addition to a web interface for real-time dynamic data analysis. The study demonstrates how blockchain technology can improve privacy, confidentiality, and data management procedures in the healthcare sector, which will ultimately result in more reliable and efficient healthcare operations. Utilizing modern technologies to address serious problems in healthcare data management and privacy is demonstrated by the successful implementation of a private Blockchain network for real-time log management.

III.DESIGN AND IMPLEMENTATION

A. Proposed System

The proposed system uses blockchain technology, specifically Ethereum's Solidity programming language, to securely manage and store logs and metrics data generated throughout the DevOps deployment pipeline. The DevOps deployment pipeline generates logs and metrics that include a variety of data, including performance metrics, error logs, deployment events, and resource utilization metrics. APIs or custom scripts will be used to facilitate integration with the DevOps pipeline, enabling automated and smooth data retrieval. Using smart contracts written in Solidity programming, the logs and metrics data will be safely saved on the Ethereum blockchain after being gathered. Because each log entry and metric value is recorded as a transaction on the blockchain, this approach ensures the immutability and transparency of the data.

The logging and metrics data on the blockchain will be carefully managed by smart contracts, which will have features for adding new log entries and metric values as well as retrieving data for analysis and auditing. Access control mechanisms will be implemented to limit data manipulation to authorized users to protect data integrity and prevent tampering.

Encryption techniques will be used to protect sensitive information before log entries and metrics data are stored on the blockchain. Cryptographic hashing algorithms will be used to hash the data before storage, improving data integrity and enhancing against unauthorized alterations. Using smart contracts, the system will implement decentralized governance mechanisms to manage data ownership and access control policies. Transparent voting mechanisms will enable stakeholders to actively participate in decision-making processes, guaranteeing accountability and fairness in data management. Through APIs or custom interfaces, the blockchain-based logging system will be seamlessly integrated with the current DevOps platforms and tools, allowing for effective bidirectional data exchange with other elements of the DevOps ecosystem.

The system will use gas-efficient coding techniques to reduce transaction costs related to storing data on the blockchain, enhance smart contract code, and examine different scaling solutions like layer 2 solutions or sharding to address issues with scalability and performance. To verify the feasibility and functionality of the suggested system, a prototype implementation will be created. This will require the deployment of smart contracts to a test Ethereum network and the simulation of actual scenarios to highlight the system's effectiveness in securely managing DevOps logs and metrics through blockchain technology. Organizations can improve the security, transparency, and integrity of their DevOps processes through this complete system architecture, which will help with decision-making, security, and operational efficiency.

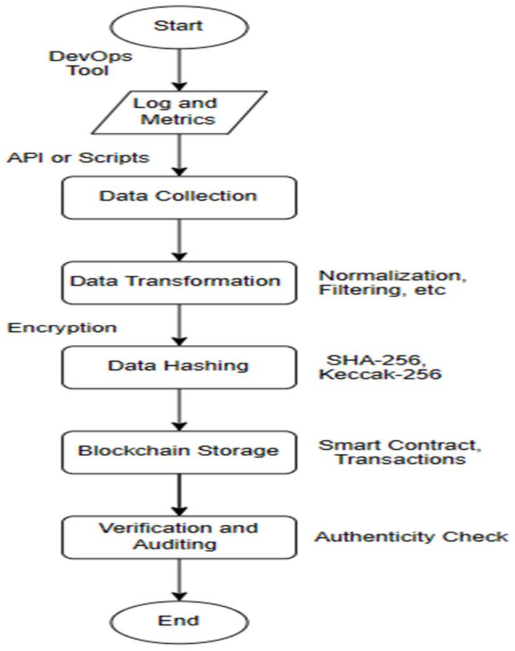


Fig 1: Process of Proposed System.

B. *Process*

The following are the processes that the proposed system includes:

1) *Data Collection:* Throughout the DevOps pipeline, logs and metrics data are gathered from a variety of sources, such as application servers, databases, network devices, and monitoring tools. Deployment events, system logs, performance metrics, error logs, and user activity logs are a few examples of this data. It is possible to collect this data in real-time as events happen using specialized logging agents, custom scripts, or APIs.
2) *Data Transformation: Before storing data on a blockchain, it may be transformed and normalized to ensure consistency* and compatibility with the blockchain data structure. To protect sensitive data, this may require converting data formats, encrypting data, and filtering or aggregating data to reduce redundancy and maximize storage efficiency.
3) *Data Hashing*: After the capture and transformation of the data, unique hash values for every data entry are generated through the application of cryptographic hashing algorithms. Hash values are computed using hashing algorithms like SHA-256 or Keccak-256. These values act like independent digital fingerprints for the original data. By generating consistent hash values for identical data and making it computationally impossible to identify the original data from its hash, as hashing ensures data integrity.

4) *Blockchain Storage:* The hashed data is stored on the blockchain as transactions inside smart contracts, together with any important metadata like timestamps, and transaction identifiers. Each log entry or metric value is recorded as a separate transaction, resulting in an immutable and transparent recording of all data changes. The data structure and logic for storing, retrieving, and updating logs and metrics data on the blockchain are defined by smart contracts written in Solidity programming.
5) *Verification and Auditing:* By examining the computed hash values stored on the blockchain with the hash values computed from the original data, blockchain users can confirm the authenticity and integrity of logs and metrics data. This procedure makes it possible to audit data changes transparently and verifiably, guaranteeing that no illegal tampering or modifications have taken place. Also, the decentralized nature of blockchain promises that data stored on the network is unaffected to manipulation and restrictions, offering a high degree of transparency and trust.

These procedures provide transparent and verifiable methods for hashing and storing logs and metrics data on the blockchain, maintaining data integrity, security, and auditability in DevOps environments.

C. *Prototype Implementation*

This sample code demonstrates how to use blockchain technology to secure logs and metrics in a DevOps environment. This contract makes sure that logs and metrics added are permanent because it makes use of the transparent and immutable Ethereum blockchain. This enhances the data's trustworthiness and integrity.

A basic prototype for securing logs and metrics from a DevOps environment using blockchain technology is demonstrated by the given Solidity code, which defines a smart contract named ‘LogMetrics.’ The two primary structures in this contract are ‘LogEntry’ and ‘MetricEntry’ . Whereas, each ‘LogEntry’ consists of a timestamp and a value and each ‘MetricEntry” consists of a timestamp and a metric value. These entries are stored in mappings that link addresses to arrays of ‘LogEntry’ and ‘MetricEntry’ structs. 'addLog' and 'addMetric' are the functions for adding logs and metrics that are included in the contract. These functions generate new entries and store them in the suitable mappings after receiving the corresponding values and timestamps as inputs. They also generate events LogAdded and MetricAdded to record the additions, which helps the blockchain's tracking of these activities. The 'totalBlocksLog' and 'totalBlocksMetrics' are state variables maintains track of the total number of blocks added and increases with each addition. ‘getLogs’ and ‘getMetrics’ functions are given by the contract to retrieve the stored logs and metrics. Users can view their stored data by using these functions, which accept a user address as input and return arrays of timestamps and values.

While this prototype demonstrates the fundamentals of using blockchain technology to secure logs and metrics, it is important to remember that a real-world deployment will involve various processes and factors. For example, important issues that must be addressed are handling scalability, optimizing gas fees, integrating with existing DevOps tools, securing private keys, and managing access control. The prototype for a real-world DevOps environment involves deploying a smart contract on an Ethereum, integrating it with the CI/CD pipeline, automating execution, managing private keys and sensitive information, and continuously monitoring the blockchain for log and metric entries to enhance trust and integrity. A complete solution would also need advanced features like real-time monitoring, strong auditing mechanisms, and automated data collection from different stages of the DevOps pipeline. Therefore, even though this prototype is a starting point, a fully-fledged system would need a more complex and detailed plan to use blockchain technology to securely store logs and metrics in a DevOps environment.

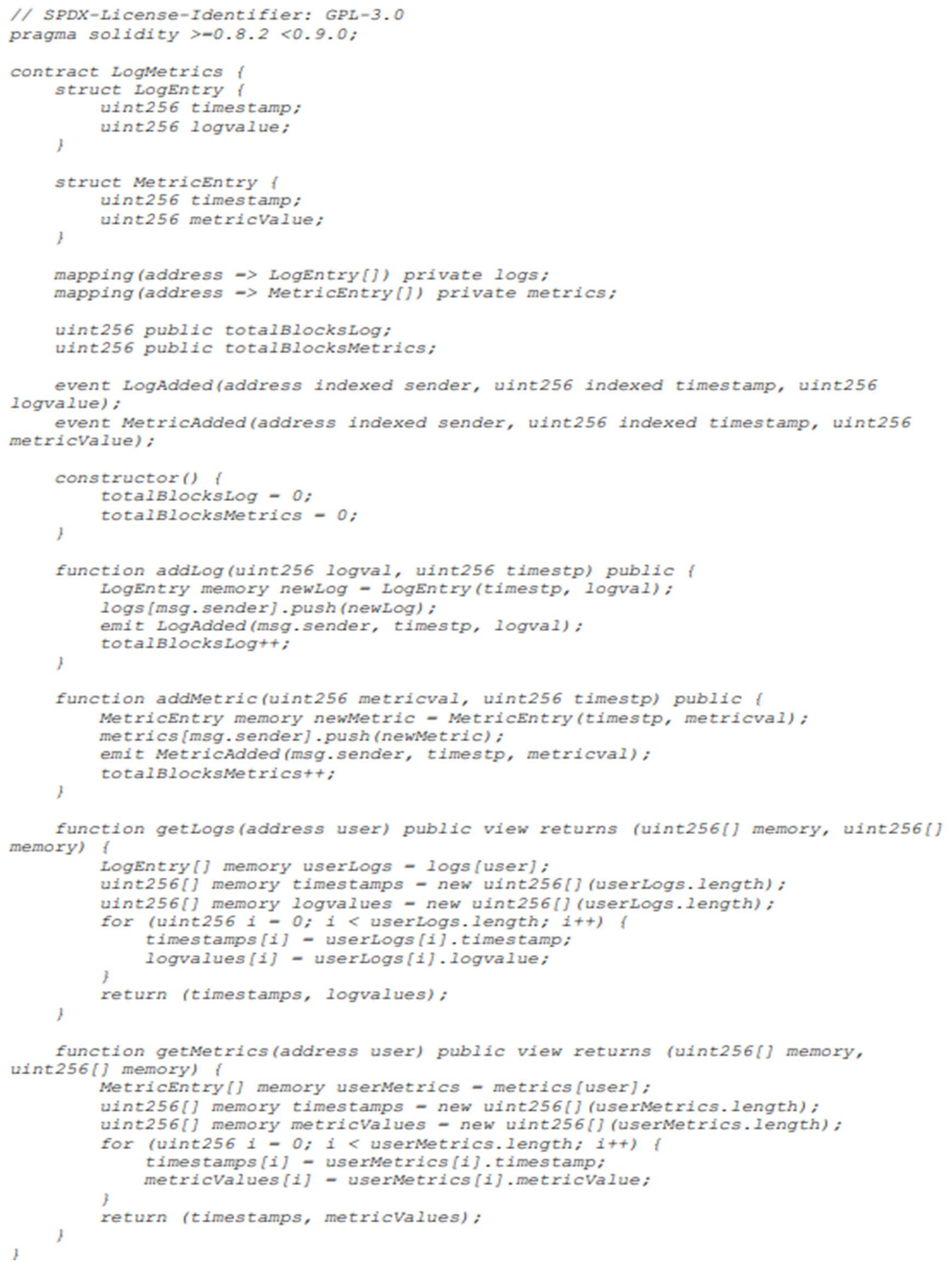
D. *Challenges of Proposed System*

It can be difficult to integrate blockchain technology into DevOps logs and metrics, especially if the pipeline consists of several different processes and tools. In overcome this, organizations can adopt a modular approach, breaking down the integration into smaller components, and establishing clear paths of communication between the blockchain, operations, and development teams to ensure an easier integration. Transaction throughput and latency are two examples of scalability issues that blockchain networks, like Ethereum, may encounter. To address these issues, organizations can look into alternative platforms with higher throughput and scalability solutions, as well as implement optimization techniques for non-critical data storage.

In blockchain technology, data privacy is a critical concern that requires establishing a balance between transparency and sensitive data protection. While technologies like encryption and zero-knowledge proofs can be helpful, it is crucial to ensure regulatory compliance with laws like the GDPR (General Data Protection Regulation) and HIPAA (Health Insurance Portability and Accountability Act). It is essential to have a strong compliance framework that includes audit procedures and encryption standards. It must be regularly verified by audits.

Another issue is user adoption and training, since integrating blockchain technology into the DevOps process might call for guidance and instruction. Users can understand the advantages of blockchain technology and successfully integrate it into their workflows with help of thorough training and ongoing guidance.

We can implement a secure solution for managing logs and metrics data by addressing these challenges strategically and following best practices. Continuous monitoring, feedback, and iteration based on real-world usage can help to improve the solution's effectiveness.



```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.2 <0.9.0;

contract LogMetrics {
    struct LogEntry {
        uint256 timestamp;
        uint256 logvalue;
    }

    struct MetricEntry {
        uint256 timestamp;
        uint256 metricValue;
    }

    mapping(address => LogEntry[]) private logs;
    mapping(address => MetricEntry[]) private metrics;

    uint256 public totalBlocksLog;
    uint256 public totalBlocksMetrics;

    event LogAdded(address indexed sender, uint256 indexed timestamp, uint256
logvalue);
    event MetricAdded(address indexed sender, uint256 indexed timestamp, uint256
metricValue);

    constructor() {
        totalBlocksLog = 0;
        totalBlocksMetrics = 0;
    }

    function addLog(uint256 logval, uint256 timestp) public {
        LogEntry memory newLog = LogEntry(timestp, logval);
        logs[msg.sender].push(newLog);
        emit LogAdded(msg.sender, timestp, logval);
        totalBlocksLog++;
    }

    function addMetric(uint256 metricval, uint256 timestp) public {
        MetricEntry memory newMetric = MetricEntry(timestp, metricval);
        metrics[msg.sender].push(newMetric);
        emit MetricAdded(msg.sender, timestp, metricval);
        totalBlocksMetrics++;
    }

    function getLogs(address user) public view returns (uint256[] memory, uint256[]
memory) {
        LogEntry[] memory userLogs = logs[user];
        uint256[] memory timestamps = new uint256[](userLogs.length);
        uint256[] memory logvalues = new uint256[](userLogs.length);
        for (uint256 i = 0; i < userLogs.length; i++) {
            timestamps[i] = userLogs[i].timestamp;
            logvalues[i] = userLogs[i].logvalue;
        }
        return (timestamps, logvalues);
    }

    function getMetrics(address user) public view returns (uint256[] memory,
uint256[] memory) {
        MetricEntry[] memory userMetrics = metrics[user];
        uint256[] memory timestamps = new uint256[](userMetrics.length);
        uint256[] memory metricValues = new uint256[](userMetrics.length);
        for (uint256 i = 0; i < userMetrics.length; i++) {
            timestamps[i] = userMetrics[i].timestamp;
            metricValues[i] = userMetrics[i].metricValue;
        }
        return (timestamps, metricValues);
    }
}
```

Fig 2: This is a Solidity Code of Proposed System.

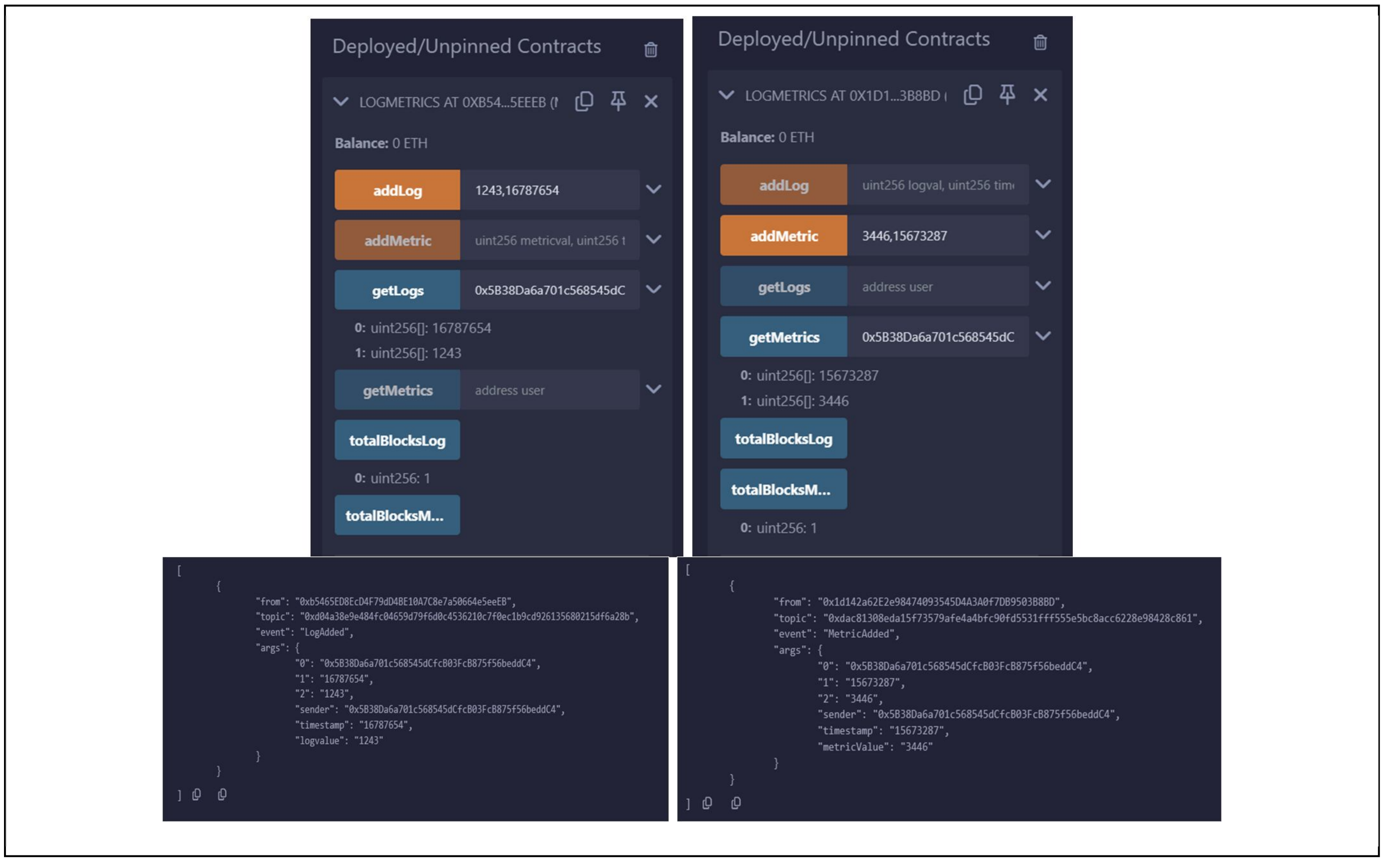
IV. RESULTS AND DISCUSSIONS

A. Results

You can use the Remix Ethereum IDE to deploy and test the 'LogMetrics' smart contract in an actual environment. Deploy the 'LogMetrics' contract first, after deployment, add log entries by calling the 'addLog' function, which will store the log entry and generate the 'LogAdded' event, with the required timestamp and value parameters. Similarly, to add metric entries, call the 'addMetric' function with the necessary timestamp and metric value, then store the entry and generate the 'MetricAdded' event. Use the 'getLogs' and 'getMetrics' functions to retrieve and view the stored logs and metrics for a given address. The total number of log and metric entries added to the system will be reflected in the 'totalBlocksLog' and 'totalBlocksMetrics' variables. You can show that it is possible to use blockchain technology to enhance the trust and integrity of logs and metrics in a DevOps environment by implementing this prototype into practice and testing it. It is important to remember that to handle issues with performance, scalability, and integration, a production-ready system would require extra levels of complexity and optimization.

B. Pros and Cons

The proposed approach of using blockchain to secure logs and metrics in a DevOps environment has several advantages over traditional logging and metric collection methods. The enhanced security and integrity are one of its main advantages. Blockchain provides a high level of trust and authenticity because of its immutable ledger, which makes logs and metrics impossible to tamper with once they are recorded. This immutability is especially useful in settings where compliance and auditability are essential. Also, the decentralized structure of blockchain reduces the risk of a single point of failure, improving the resilience of the system.
However, there are some drawbacks. When compared to traditional systems, the performance costs of blockchain may result in slower log and metric processing times. Another issue is scalability, since blockchains can have trouble effectively managing massive amounts of data. It may also be necessary to make major changes and add complexity to integrate blockchain with existing DevOps pipelines. Finally, the financial cost of blockchain transactions, especially on open networks like Ethereum may avoid them being widely adopted. Despite these difficulties, the suggested method provides a strong answer in situations where data integrity and trust are important.



```
addLog 1243,16787654
getLogs 0x5B38Da6a701c568545dC
0: uint256[]: 16787654
1: uint256[]: 1243
totalBlocksLog
0: uint256: 1
```

```
[
        {
                "from": "0xb5465ED8EcD4F79dD4BE10A7C8e7a50664e5eeEB",
                "topic": "0xd04a38e9e484fc04659d79f6d0c4536210c7f0ec1b9cd926135680215df6a28b",
                "event": "LogAdded",
                "args": {
                        "0": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                        "1": "16787654",
                        "2": "1243",
                        "sender": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                        "timestamp": "16787654",
                        "logvalue": "1243"
                }
        }
]
```

```
addMetric 3446,15673287
getMetrics 0x5B38Da6a701c568545dC
0: uint256[]: 15673287
1: uint256[]: 3446
totalBlocksM...
0: uint256: 1
```

```
[
        {
                "from": "0x1d142a62E2e98474093545D4A3A0f7DB9503B8BD",
                "topic": "0xdac81308eda15f73579afe4a4bfc90fd5531fff555e5bc8acc6228e98428c861",
                "event": "MetricAdded",
                "args": {
                        "0": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                        "1": "15673287",
                        "2": "3446",
                        "sender": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                        "timestamp": "15673287",
                        "metricValue": "3446"
                }
        }
]
```

Fig 3: Output of Proposed System Code.

C. Future work

Future research should focus on scalable blockchain solutions, privacy enhancements, hybrid approaches, standardization and interoperability, and performance optimization. Scalable solutions like sharding and layer-2 protocols can address throughput limitations in log and metric storage. Privacy-preserving techniques like zero-knowledge proofs can protect sensitive data while maintaining transparency. Hybrid models combining blockchain with traditional databases can optimize performance and cost while maintaining data integrity. While research on reducing the performance impact of blockchain transactions on logging and metrics systems can make the approach easier to implement for high-frequency data environments, the integration of blockchain with DevOps tools can improve wider adoption and interoperability.

Blockchain can be used in distributed systems and microservices, security-sensitive environments like finance, healthcare, and government, and compliance and regulatory audits in organizations like GDPR or HIPAA. Its transparency and immutability make audits easier to conduct and can prove legality. Blockchain can offer a trustworthy source of truth for logs and metrics in complex architectures with interdependent services, helping in system monitoring and debugging. Technical expertise, integration with existing systems, financial cost, and unclear regulations can all slow down blockchain adoption. Organizations may struggle with in-house expertise and specialized skills, and integrating blockchain solutions with existing systems can be complicated and require significant changes. The cost of implementing and maintaining blockchain-based logging systems can be expensive.

V. CONCLUSION

In today's dynamic DevOps environment, securing the security and integrity of logs and metrics is essential to maintaining trust in continuous deployment pipelines. The required levels of security, transparency, and trustworthiness are sometimes difficult to achieve using traditional methods. The paper has investigated how blockchain technology, especially Ethereum and Solidity programming may be able to help with these problems. The proposed blockchain-based approach improves the security of log and metric data in DevOps processes by utilizing the immutable and decentralized features of blockchain technology. The solution greatly enhances data integrity and trust by utilizing smart contracts to enable tamper-proof storage, transparent audits, and precise access control. Our prototype implementation demonstrates the potential of this approach and its benefits over traditional methods.

According to the study's findings, using blockchain technology into DevOps processes can reduce the danger of tampering and illegal access, offering a more trustworthy solution for log and metric security. This method not only improves data integrity but also makes the auditing process clear and verifiable, both of which are essential for operational trust and regulatory compliance. The report also identifies a few topics for further investigation and development, such as interoperability, improved security features, performance improvement, scalability, and cost minimization. It will be important to address these issues if blockchain-based DevOps solutions are to be widely adopted. Finally, blockchain technology is an effective choice for DevOps settings to secure logs and data. It provides the way for continuous deployment processes that are more safe, transparent, and trustworthy by getting over the drawbacks of traditional methods. The prototype and insights discussed in this article offer a starting point for more research and development in this area, which will ultimately lead to more reliable and secure DevOps processes.

VI. ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to Mr. Chayan Bhattacharjee for his valuable guidance and support throughout the development of this research paper. His insights and expertise have helped define the scope and quality of this project. I am also grateful to Mrs. Sumedha Barve, the head of the IT Department, for her continuous support and encouragement. Her leadership and vision provided a solid foundation for this research, allowing me to explore and contribute to this interesting topic. Thank you both for your mentoring and believing in this idea.

REFERENCES

[1] Hany Fawzy, A., Wassif, K., & Moussa, H. (2023, March). Framework for automatic detection of anomalies in DevOps. Journal of King Saud University - Computer and Information Sciences, 35(3), 8–19. https://doi.org/10.1016/j.jksuci.2023.02.010

[2] Guimarães, T., Duarte, R., Pinheiro, B., Faria, D., Gomes, P., & Santos, M. F. (2022). Blockchain Analytics - Real-time Log Management in Healthcare. Procedia Computer Science, 201, 702–707. https://doi.org/10.1016/j.procs.2022.03.094

[3] Ahmad, H., & Aujla, G. S. (2023, August). GDPR compliance verification through a user-centric blockchain approach in multi-cloud environment. Computers and Electrical Engineering, 109, 108747. https://doi.org/10.1016/j.compeleceng.2023.108747

[4] Garg, Rishabh. (2023). Blockchain for Decentralized Finance: Impact, Challenges and Remediation. International Journal of Computer Science and Information Technology. 17. 225-237.

[5] Electronic Voting Using Decentralized System Based on Ethereum Blockchain. (2020, March 30). Jurnal Ilmiah Komputasi, 19(1). https://doi.org/10.32409/jikstik.19.1.152

[6] Pathak, Prof & Suradkar, Amol & Kadam, Ajinkya & Ghodeswar, Akansha & Parde, Prashant. (2021). Blockchain Based E-Voting System. International Journal of Scientific Research in Science and Technology. 134-140. 10.32628/IJSRST2182120.

[7] Junaid Nasir Qureshi, Muhammad Shoaib Farooq, Usman Ali et al. Exploring the Integration of Blockchain and Distributed DevOps for Secure, Transparent, and Traceable Software Development, 27 July 2023, PREPRINT (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-3187369/v1]

[8] Muhammad Azeem Akbar, Sajjad Mahmood, and Dominik Siemon. 2022. Toward Effective and Efficient DevOps using Blockchain. In Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering (EASE '22). Association for Computing Machinery, New York, NY, USA, 421–427. https://doi.org/10.1145/3530019.3531344

[9] Singhal, B., Dhameja, G., & Panda, P. S. (2018, July 6). Beginning Blockchain. Apress. http://books.google.ie/books?id=fEpjDwAAQBAJ&printsec=frontcover&dq=Beginning+Blockchain&hl=&cd=1&source=gbs_api

[10] M. Wöhrer and U. Zdun, "DevOps for Ethereum Blockchain Smart Contracts," 2021 IEEE International Conference on Blockchain (Blockchain), Melbourne, Australia, 2021, pp. 244-251, doi: 10.1109/Blockchain53845.2021.00040.

[11] S. Bankar and D. Shah, "Blockchain based framework for Software Development using DevOps," 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), NaviMumbai, India, 2021, pp. 1-6, doi: 10.1109/ICNTE51185.2021.9487760.

[12] Dannen, Chris. (2017). Introducing Ethereum and Solidity. 10.1007/978-1-4842-2535-6.

[13] Walls, M. (2013, April 15). Building a DevOps Culture. "O'Reilly Media, Inc." http://books.google.ie/books?id=L1BZ0w0g-v4C&printsec=frontcover&dq=Building+a+DevOps+culture.&hl=&cd=1&source=gbs_api

[14] Farooq, Muhammad & Ali, Usman. (2023). Harnessing the Potential of Blockchain in DevOps: A Framework for Distributed Integration and Development.

[15] M. H. Rakib, S. Hossain, M. Jahan and U. Kabir, "Towards Blockchain-Driven Network Log Management System," 2020 IEEE 8th International Conference on Smart City and Informatization (iSCI), Guangzhou, China, 2020, pp. 73-80, doi: 10.1109/iSCI50694.2020.00019.

[16] Tenkrooden, Kyle. (2023). Ensuring GDPR Compliance for a Small-Scale Dynamic Web Application Using AWS Cloud Services: Best Practices and Strategies.

[17] Chen, Boyuan & Cui, Jun. (2021). Improving the software logging practices in DevOps.

[18] Shang, Weiyi & Nagappan, Meiyappan & Hassan, Ahmed E. & Jiang, Zhen. (2014). Understanding Log Lines Using Development Knowledge. Proceedings - 30th International Conference on Software Maintenance and Evolution, ICSME 2014. 21-30. 10.1109/ICSME.2014.24.

[19] Ali, & Khan, Abid & Ahmed, Mansoor & Jeon, Gwanggil. (2021). BCALS: Blockchain-based secure log management system for cloud computing. Transactions on Emerging Telecommunications Technologies. 33. 10.1002/ett.4272.

[20] Tripathi, G., Ahad, M. A., & Casalino, G. (2023, December). A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges. Decision Analytics Journal, 9, 100344. https://doi.org/10.1016/j.dajour.2023.100344

[21] Urquhart, A. (2022, June). Under the hood of the Ethereum blockchain. Finance Research Letters, 47, 102628. https://doi.org/10.1016/j.frl.2021.102628

[22] Gowthaman, A. & Manickam, Sumathi. (2015). Performance study of enhanced SHA-256 algorithm. 10. 10921-10932.

[23] Mohammed, Alaa & Abdulateef, Alaa & Abdulateef, Ihsan. (2021). Hyperledger, Ethereum and Blockchain Technology: A Short Overview. 1-6. 10.1109/HORA52670.2021.9461294.

[24] Vujičić, Dejan & Jagodic, Dijana & Ranđić, Siniša. (2018). Blockchain technology, bitcoin, and Ethereum: A brief overview. 1-6. 10.1109/INFOTEH.2018.8345547.

[25] Tandon, A., Kaur, P., Mäntymäki, M., & Dhir, A. (2021, May). Blockchain applications in management: A bibliometric analysis and literature review. Technological Forecasting and Social Change, 166, 120649. https://doi.org/10.1016/j.techfore.2021.120649