

SECURE AND TRANSPARENT CLOUD TASK SCHEDULING USING BLOCKCHAIN TECHNOLOGY

DR. GADE VENKATA VARA PRASAD¹, T. SHASHI KUMAR², V. MANIKANTA³, T. HARI KRISHNA⁴, CH. SHIVA DURGA
SURYA PRABHAS⁵

ASSOCIATE PROFESSOR¹, UG SCHOLAR^{2,3,4&5}

DEPARTMENT OF AI&ML, NARSIMHA REDDY ENGINEERING COLLEGE (UGC- AUTONOMOUS) MAISAMMAGUDA
(V), KOMPALLY, SECUNDERABAD, TELANGANA-500100

ABSTRACT

Cloud computing has revolutionized data storage and computational processes, but it still faces challenges in secure task scheduling and data integrity. Existing approaches, as discussed in the base paper, suffer from issues such as inefficient scheduling algorithms, high computational overhead, and vulnerabilities in data security. To address these limitations, we propose an improved blockchain-integrated task scheduling system that ensures **tamper-proof data recording, enhanced verification mechanisms, and automated document validation**. Our approach leverages **smart contracts** to securely store task submissions on the blockchain, providing transparency and immutability. By integrating blockchain verification, we eliminate unauthorized modifications and enhance trust in the scheduling process. Additionally, our system automates **task validation and document conversion**, reducing manual overhead. When an admin verifies a task, the associated document is automatically converted into a **PDF**, updating the task status while maintaining a decentralized and auditable record. Compared to existing methods, our system offers **greater security, improved efficiency, and seamless task traceability**. Through this novel blockchain-based framework, we successfully mitigate the drawbacks highlighted in the base paper, enhancing the security and reliability of task scheduling in cloud environments.

INTRODUCTION

Background and Motivation

Cloud computing has become an essential technology for handling large-scale data storage, computation, and service provisioning. It provides organizations and users with scalable, cost-effective, and on-demand access to computational resources. However, with the increasing adoption of cloud-based services, several challenges have emerged, particularly in **task scheduling, security, and data integrity**. The dynamic nature of cloud environments, coupled with the need for efficient task execution, necessitates the development of robust scheduling mechanisms.

The base paper highlights several task scheduling approaches but also identifies key **limitations in security, efficiency, and trust management**. Traditional scheduling algorithms often fail to provide **reliable verification mechanisms**, making them vulnerable to **tampering, unauthorized access, and inefficient resource allocation**. Additionally, the lack of **transparency in task scheduling** and execution has led to issues such as **data manipulation, unauthorized modifications, and inefficient tracking**.

To overcome these challenges, we introduce a **blockchain-based task scheduling system** that leverages the **immutability, transparency, and decentralized nature of blockchain technology**. This approach ensures **tamper-proof task submissions**, enhances trust through **blockchain verification mechanisms**, and automates document validation through **smart contract integration**. By addressing the weaknesses of the existing methodologies, our proposed system provides a **more secure and efficient** solution for task scheduling in cloud environments.

Challenges in Traditional Task Scheduling

Task scheduling in cloud computing refers to the process of **assigning computational tasks to available resources** based on specific constraints such as **time, cost, energy efficiency, and security**. Several scheduling techniques, including heuristic-based, genetic algorithms, and swarm intelligence, have been developed to optimize task execution. However, these approaches suffer from inherent **security vulnerabilities and inefficiencies**, as outlined below:

1. **Lack of Data Integrity:** Traditional scheduling models store task data in centralized databases, making them susceptible to **unauthorized modifications and data breaches**. If a malicious actor gains access to the system, they can **alter or delete task records**, compromising the reliability of scheduling.
2. **Absence of Transparent Verification:** Task completion and validation mechanisms often **lack**

transparency, making it difficult for stakeholders to verify whether a task has been executed correctly or manipulated.

3. **Security Threats in Cloud Environments:** The transmission of sensitive task-related data over cloud networks exposes it to **cyber threats, including unauthorized access, man-in-the-middle attacks, and data leaks.**
4. **Inefficient Tracking and Auditing:** Current task scheduling systems do not provide an **immutable and auditable record of task execution**, leading to **trust issues and accountability concerns.**
5. **Manual Task Verification Overhead:** In many cloud environments, task validation is a **manual process** that requires administrative intervention, increasing the risk of **human errors and delays.**

The limitations of these traditional approaches necessitate the integration of a **trustless and verifiable** system that can securely store, track, and validate task execution. Blockchain technology provides a **decentralized and tamper-proof** alternative that effectively addresses these issues.

The Role of Blockchain in Task Scheduling

Blockchain technology is a **decentralized, distributed ledger** that records transactions in an immutable and transparent manner. It eliminates the **need for intermediaries** and ensures **data integrity** by storing information across multiple nodes in a network. The key features of blockchain that enhance task scheduling include:

1. **Immutability:** Once a transaction is recorded on the blockchain, it **cannot be altered or deleted**, ensuring the integrity of task submissions.
2. **Transparency:** All blockchain transactions are publicly accessible (or permissioned in private blockchains), allowing **verifiable and auditable task execution.**
3. **Smart Contracts:** These self-executing contracts automate task validation and **eliminate the need for manual verification**, reducing administrative overhead.
4. **Decentralization:** By removing central authorities, blockchain prevents **unauthorized modifications and data breaches**, ensuring **secure task execution.**

5. **Trust and Security:** Blockchain employs **cryptographic hashing and consensus mechanisms**, making it highly resistant to **fraud and unauthorized alterations.**

Our approach leverages **blockchain technology to securely store task data**, verify task execution through **blockchain transactions**, and automate document validation using **smart contracts**. This ensures **end-to-end security, transparency, and reliability** in cloud task scheduling.

Proposed Blockchain-Based Task Scheduling System

The proposed system introduces a **novel blockchain-integrated task scheduling framework** that overcomes the limitations of traditional models. It consists of the following key components:

1. Task Submission and Storage:

- Users submit tasks (Word documents) to the system.
- The **task details, including user ID, document metadata, and timestamp**, are recorded in both the **Django database and blockchain.**
- Blockchain ensures that **task records are tamper-proof and permanently stored.**

2. Blockchain Transaction Verification:

- Each task submission generates a **unique blockchain transaction ID (blockchain_tx).**
- The transaction is **stored on the blockchain**, ensuring that task records remain immutable.
- Users and admins can **verify task authenticity** using blockchain explorers.

3. Automated Task Validation and Document Conversion:

- When an admin **clicks the "Verify" button**, the system:
 - **Fetches the blockchain transaction** to validate task authenticity.

- Converts the uploaded Word document to a PDF using automated conversion tools.
- Updates the task status to "Completed", allowing users to download the final document.

4. User and Admin Interfaces:

- Users can submit tasks and download completed PDFs.
- Admins can verify transactions, process tasks, and ensure authenticity through blockchain validation.

Key Advantages of Our Approach

Our blockchain-integrated task scheduling system provides several advantages over traditional scheduling models:

1. Enhanced Security and Integrity:

- Blockchain ensures **tamper-proof task records**, eliminating the risk of **unauthorized data modifications**.

2. Automated and Efficient Verification:

- Smart contracts automate **task validation**, reducing the need for **manual administrative oversight**.

3. Transparent and Auditable Transactions:

- Every task submission and verification step is **recorded on the blockchain**, allowing **stakeholders to audit and verify transactions**.

4. Seamless Document Processing:

- The system **automatically converts Word documents to PDFs** once verified, streamlining workflow and improving efficiency.

5. User-Friendly Task Tracking and Downloading:

- Users can **track the status of their tasks** and **download completed documents without additional requests**.

6. Decentralized Trust Model:

- By leveraging blockchain, the system **removes the need for central trust authorities**, ensuring a **trustless and reliable task scheduling mechanism**.

Conclusion

This work introduces a **secure, transparent, and efficient task scheduling system** that integrates **blockchain verification and automated document processing**. By leveraging **smart contracts and decentralized storage**, our approach **addresses the shortcomings of existing scheduling models** and enhances **trust, security, and usability**. Compared to the base paper, our solution provides **greater transparency, automated task validation, and seamless task tracking**, making it a superior alternative for secure cloud-based task scheduling.

This research lays the foundation for **future improvements in decentralized cloud task management**, where blockchain **revolutionizes trust and efficiency** in scheduling mechanisms.

LITERATURE SURVEY

Cloud task scheduling is a crucial component in distributed computing, ensuring efficient allocation of resources while maintaining security, cost efficiency, and performance optimization. Various approaches have been explored in the literature to address the challenges in scheduling, including heuristic methods, evolutionary algorithms, and blockchain-based security enhancements. However, these methods have limitations in terms of **security, transparency, and verifiability**, which we aim to overcome in our proposed blockchain-integrated task scheduling system. This section reviews existing approaches, their limitations, and how our proposed solution improves upon them.

Traditional Task Scheduling Approaches

Task scheduling in cloud environments has traditionally relied on heuristic and metaheuristic optimization techniques. These methods aim to balance resource allocation, execution time, and cost-effectiveness. However, they often fall short in providing **security, transparency, and immutability**.

Heuristic-Based Scheduling

Heuristic algorithms are widely used in task scheduling due to their ability to find near-optimal solutions efficiently. Common heuristics include:

- **First Come First Serve (FCFS):** Tasks are executed in the order of arrival. It is simple but does not consider task priority or resource availability, leading to inefficient utilization.
- **Round Robin (RR):** Tasks are assigned to available resources in a cyclic manner, ensuring fairness but often causing delays in high-priority tasks.
- **Min-Min and Max-Min:** These approaches select tasks based on their execution time, aiming to optimize overall completion time. However, they suffer from **high scheduling costs** and inefficiency in dynamic environments.

Limitations of Heuristic Approaches:

- ✓ Lack of security mechanisms to prevent unauthorized task modifications.
- ✓ No transparency or auditability in task execution records.
- ✓ Inefficient in dynamic cloud environments where task requirements frequently change.

Metaheuristic and Evolutionary Scheduling

To improve upon heuristic scheduling, metaheuristic techniques such as **Genetic Algorithms (GA)**, **Particle Swarm Optimization (PSO)**, and **Ant Colony Optimization (ACO)** have been applied in cloud task scheduling.

- **Genetic Algorithm (GA):** Inspired by natural selection, GA optimizes task scheduling by evolving solutions through selection, crossover, and mutation. While effective, **GA requires high computation time**.
- **Particle Swarm Optimization (PSO):** Based on swarm intelligence, PSO optimizes scheduling by improving resource allocation. However, **PSO struggles with local optima issues**.
- **Ant Colony Optimization (ACO):** Mimics the foraging behavior of ants to optimize task distribution. ACO provides **good load balancing but is slow in large-scale scheduling**.

Limitations of Metaheuristic Approaches:

- ✓ High computational complexity leads to **increased execution time**.

- ✓ No **built-in security mechanisms** to protect task data.

- ✓ Lack of **verifiable and immutable task records**, making auditing difficult.

Our approach **integrates blockchain technology** to provide **secure, tamper-proof task records**, ensuring **transparent and efficient scheduling**.

Blockchain-Based Scheduling and Security Mechanisms

Blockchain has been explored as a security-enhancing technology in cloud task scheduling due to its **decentralized, immutable, and trustless nature**. Several studies have proposed integrating blockchain with scheduling, focusing primarily on **security and trust management**.

Blockchain for Secure Task Scheduling

Recent studies have highlighted the role of blockchain in improving **data integrity and trust** in cloud computing:

- **F. Zhang et al. (2019) - Trusted Routing Scheme Using Blockchain:**
 - Proposed a routing scheme where blockchain records route information to **detect malicious nodes**.
 - While improving **security**, this method introduced **high overhead in transaction processing**.
- **Wei She (2019) - Blockchain Trust Model (BTM) for Malicious Node Detection:**
 - Introduced a trust model based on blockchain for **detecting malicious nodes in wireless sensor networks**.
 - The method showed high accuracy but **relied on voting mechanisms prone to false positives**.
- **Xiaolong Xu et al. (2021) - PBFT Consensus for Cloud Task Scheduling:**
 - Applied **Practical Byzantine Fault Tolerance (PBFT)** to improve **trust and consensus in cloud scheduling**.

- While securing transactions, PBFT suffers from **higher computational costs**.

Limitations of Blockchain-Based Approaches:

- ✓ High processing overhead due to **complex consensus mechanisms**.
- ✓ Many approaches **focus only on security** but do not integrate **task automation** or **verification**.
- ✓ Lack of seamless user interaction for task submission and document processing.

Our solution **overcomes these issues** by:

- ✓ Using **lightweight smart contracts** to store **task submissions** without high computational overhead.
- ✓ Providing **automatic task verification and document conversion**, reducing **manual effort**.
- ✓ Ensuring **task immutability and trust** using **blockchain-stored transactions**.

Comparative Analysis of Existing and Proposed Approaches

The table below compares traditional scheduling, blockchain-based scheduling, and our **proposed blockchain-integrated task scheduling system** in terms of security, efficiency, and usability.

Feature	Traditional Scheduling	Blockchain-Based Scheduling	Our Proposed System
Security	Low (Data stored centrally, risk of tampering)	High (Immutable blockchain storage)	High (Blockchain + Smart Contracts)
Task Verification	Manual	Partial (Limited to blockchain logs)	Automated through smart contracts
Data Integrity	Moderate (No immutability)	High (Tamper-proof transactions)	High (Secure and immutable records)
User-Friendly Interface	Limited	Moderate (Mainly admin-focused)	Yes (Users & admins interact via Django-based UI)

Feature	Traditional Scheduling	Blockchain-Based Scheduling	Our Proposed System
Transparency	Low (No public logs)	Moderate (Limited blockchain logs)	High (Real-time task tracking & verification)
Task Processing Automation	No	Partial	Yes (Blockchain verification + PDF conversion)

Our approach **not only ensures security and immutability** but also provides **automated verification, document processing, and enhanced transparency**, making it superior to both traditional and blockchain-based scheduling models.

How Our Approach Overcomes the Limitations

By analyzing the limitations of existing scheduling techniques, we have designed an improved blockchain-integrated task scheduling framework with the following key improvements:

1. Immutability & Security:

- Unlike traditional scheduling models, **task data and verification records are stored on the blockchain**, preventing tampering and unauthorized modifications.

2. Automated Verification:

- Instead of relying on manual verification, **smart contracts automatically validate whether a task's blockchain transaction ID exists**, reducing human error.

3. Efficient Task Processing:

- When an **admin verifies a task**, the system **automatically converts** the Word document into a **PDF**, updates the **task status**, and enables **user downloads**.

4. Seamless User Interaction:

- Our system provides an **easy-to-use web interface** where users can **upload, track, and download documents**, enhancing

usability compared to traditional
blockchain-based scheduling.

Conclusion

This literature survey analyzed various **task scheduling techniques** and their **security implications**, highlighting the **gaps in efficiency, transparency, and automation**. While **traditional scheduling techniques** suffer from **security vulnerabilities**, **blockchain-based approaches** often introduce **high processing overhead without full automation**.

Our proposed system **addresses these challenges** by integrating **blockchain verification, automated task processing, and user-friendly tracking**, enhancing **security, efficiency, and usability** in cloud task scheduling.

This research sets the foundation for **trustless, efficient, and secure task execution**, **eliminating the risks of tampering and inefficiency** faced in existing methodologies.

SYSTEM ANALYSIS

EXISTING SYSTEM VS. PROPOSED SYSTEM IN BLOCKCHAIN-BASED TASK SCHEDULING

Introduction

In recent years, cloud-based task scheduling has been widely used for handling various computational workloads. However, security concerns, centralized control, and inefficient resource management pose significant challenges. The existing systems primarily rely on centralized task management and cloud infrastructure, making them vulnerable to security breaches and tampering. This document presents a comparison between the **existing system** and our **proposed blockchain-integrated system** using **Django and Ganache** for secure task scheduling.

Existing System

Overview

The traditional cloud task scheduling system follows a centralized architecture where all tasks are handled by a central cloud service provider. The major components include:

- **User Interface:** Users submit tasks via a web application.
- **Cloud Computing Resources:** The system schedules tasks on virtual machines (VMs) or dedicated servers.

- **Task Scheduling Algorithm:** Various scheduling strategies (e.g., FCFS, Round Robin, ACO) are used.
- **Centralized Database:** All transaction and task logs are stored in a single database.

Limitations

Despite its advantages, the existing system has several drawbacks:

1. **Single Point of Failure:** If the central cloud service fails, the entire system becomes inoperable.
2. **Security Vulnerabilities:** Data manipulation, unauthorized access, and task mismanagement can occur.
3. **Lack of Transparency:** Users cannot independently verify whether their task was executed correctly.
4. **Trust Issues:** Users must completely trust the service provider with their data and processing integrity.
5. **Resource Management Issues:** Inefficient allocation of computing power leads to delays and cost inefficiencies.

Proposed System

Overview

The proposed system integrates **blockchain technology (Ganache)** into **Django's cloud task scheduling system**, ensuring **transparency, security, and decentralized verification**. It eliminates the risks associated with centralized task scheduling by leveraging blockchain's immutable ledger.

Key Components

1. **User Interface (Bootstrap + Django Traditional Views):**
 - Users can log in, submit tasks, track status, and verify blockchain transactions.
2. **Blockchain Integration (Ganache + Web3.py):**
 - Every task submitted is recorded on the **Ganache blockchain**, ensuring immutability.
 - Users receive a **Blockchain Transaction ID** to verify task execution.

3. MySQL as Cloud Database:

- Stores task metadata, user details, and execution results.

4. Task Scheduling & Execution:

- Admin assigns computing resources for task execution.
- Task results are stored in MySQL and linked to blockchain records.

5. Admin Dashboard:

- Admin can activate/deactivate users, manage cloud resources, and verify blockchain transactions before approving tasks for execution.

Advantages Over Existing System

- Decentralized Security:** Transactions are logged in **Ganache blockchain**, preventing manipulation.
- Tamper-Proof Record Keeping:** Each task's lifecycle is recorded immutably.
- User Verification of Execution:** Users can independently verify if their task was processed by checking its blockchain record.
- Resource Efficiency:** Admins allocate tasks based on real-time compute resource availability, optimizing scheduling performance.
- Improved Transparency:** All transactions and approvals are verifiable, reducing trust dependency on a centralized entity.
- No Third-Party Services:** Fully self-contained system with **Django, MySQL, and Ganache**.

System Workflow

User Workflow

- User logs in.**
- User submits a task** (e.g., converting Word to PDF, file storage).
- Task is stored in MySQL and recorded on the blockchain (Ganache).**

4. User receives a Blockchain Transaction ID.

- User verifies the transaction** via a blockchain lookup feature.
- Once the task is processed**, the result is available for download.
- User logs out.**

Admin Workflow

- Admin logs in.**
- Admin views blockchain transaction history.**
- Admin verifies blockchain transactions before approving tasks.**
- Admin assigns computing resources for task execution.**
- Admin monitors system activity and logs out.**

Comparison Table

Feature	Existing System	Proposed System
Security	Vulnerable to attacks	Tamper-proof with blockchain
Transparency	Opaque task processing	Users can verify via blockchain
Trust Model	Centralized, trust provider	Decentralized, trust in blockchain
Data Integrity	Data can be altered	Immutable blockchain records
Resource Allocation	Fixed scheduling	Dynamic, based on availability
Fault Tolerance	Single point of failure	Distributed ledger resilience
User Control	Limited visibility	User-verifiable execution
Cost Efficiency	& High due to resource mismanagement	Optimized using real-time data

Conclusion

The proposed **blockchain-powered task scheduling system** enhances **security, transparency, and efficiency** compared to traditional cloud-based solutions. By leveraging **Ganache blockchain for transaction logging and verification**, users and admins gain **trustless interaction and improved security**. This solution ensures **tamper-proof task execution**, decentralizes control, and provides **end-to-end task traceability**, making it a **next-generation cloud task management solution**.

Future Enhancements

While the current system ensures security and transparency, future improvements may include:

- **Smart Contract Automation:** Automate task execution approvals.
- **Multi-Chain Support:** Deploy on public blockchains (Ethereum, Hyperledger).
- **AI-based Resource Optimization:** Use ML models to predict task execution times.
- **Decentralized Compute Nodes:** Allow distributed nodes to execute tasks.

By adopting blockchain-based scheduling, organizations can **significantly reduce operational risks and improve cloud task efficiency**.

IMPLEMENTATION

The **Secure and Transparent Cloud Task Scheduling System using Blockchain Technology** is designed to ensure reliable, decentralized, and tamper-proof task scheduling in cloud computing environments. The system integrates **cloud computing infrastructure with blockchain technology** to improve security, transparency, and trust in task allocation and execution. The architecture is divided into several functional modules.

User Request Module

This module handles task requests submitted by users or applications to the cloud system.

Functions:

- Accept task requests from users.
- Define task requirements such as CPU, memory, and storage.
- Authenticate users before task submission.

The task request is forwarded to the scheduling system for processing.

Task Management Module

This module manages the tasks submitted to the cloud environment.

Functions:

- Store and organize tasks in the task queue.
- Classify tasks based on priority and resource requirements.
- Prepare tasks for scheduling and resource allocation.

This module ensures efficient management of multiple tasks.

Blockchain Network Module

This module implements the blockchain infrastructure for secure task scheduling.

Functions:

- Maintain a decentralized distributed ledger.
- Record task scheduling activities as blockchain transactions.
- Ensure immutability and transparency of scheduling records.

The blockchain prevents tampering or manipulation of scheduling decisions.

Smart Contract Module

Smart contracts automate the scheduling process and enforce predefined rules.

Functions:

- Define scheduling policies and rules.
- Automatically allocate tasks to appropriate cloud resources.
- Execute task scheduling without human intervention.

Smart contracts ensure **fair and secure task allocation**.

Resource Allocation Module

This module assigns available cloud resources to tasks.

Functions:

- Monitor available computing resources.
- Allocate virtual machines or containers to tasks.
- Optimize resource utilization and reduce scheduling delays.

This module improves overall system efficiency.

Task Execution Module

This module executes tasks on allocated cloud resources.

Functions:

- Monitor task execution status.
- Ensure tasks are completed successfully.
- Record execution results in the blockchain for verification.

This provides transparency in the execution process.

Monitoring and Verification Module

This module continuously monitors the system and verifies task scheduling records.

Functions:

- Track task progress and resource usage.
- Verify scheduling decisions stored on the blockchain.
- Detect anomalies or malicious activities.

This module ensures system reliability and security.

Reporting and User Interface Module

This module provides dashboards and reports for administrators and users.

Functions:

- Display task scheduling information.
- Show resource utilization statistics.
- Provide transparency through blockchain transaction records.

This helps administrators monitor system performance.

CONCLUSION

Cloud computing has become an essential platform for storing data and executing large-scale computational tasks. However, traditional cloud task scheduling systems often suffer from security vulnerabilities, lack of transparency, and dependence on centralized control mechanisms. These limitations may lead to issues such as data tampering, unauthorized access, and inefficient resource allocation.

The proposed **blockchain-based cloud task scheduling framework** provides a secure and transparent solution for managing task execution in cloud environments. By utilizing the decentralized and immutable nature of blockchain technology, the system ensures that task scheduling information is securely recorded and cannot be altered by unauthorized entities. Smart contracts further automate the scheduling process, improving efficiency while reducing the need for manual intervention.

The integration of blockchain enhances **data integrity, trust, and accountability** within the cloud ecosystem. It enables transparent monitoring of task allocation and execution while protecting

sensitive data from potential cyber threats. Additionally, the decentralized architecture minimizes reliance on a single authority, thereby reducing the risks associated with centralized systems.

Overall, the proposed framework demonstrates significant potential in improving the **security, reliability, and transparency** of cloud task scheduling systems. Future research may focus on optimizing blockchain performance, reducing computational overhead, and integrating advanced technologies such as **artificial intelligence and edge computing** to further enhance the efficiency and scalability of secure cloud environments.

REFERENCES

1. Zhang, F., Wang, J., & Liu, Y. (2019). *Trusted Routing Scheme Using Blockchain for Cloud Security*. IEEE Transactions on Cloud Computing, 7(3), 458-469.
2. She, W., & Liu, X. (2019). *Blockchain-Based Trust Model for Malicious Node Detection in Cloud Computing*. International Journal of Computer Applications, 182(45), 12-20.
3. Xu, X., Zhang, Q., & Yang, W. (2021). *Enhancing Cloud Task Scheduling Using Blockchain and PBFT Consensus*. Future Generation Computer Systems, 124, 67-79.
4. Singh, A., & Chana, I. (2016). *A Survey of Task Scheduling Techniques in Cloud Computing*. Journal of Supercomputing, 72(5), 1423-1457.
5. Sharma, R., & Patel, D. (2020). *Blockchain Integration in Cloud Computing: A Review on Security and Privacy Challenges*. ACM Computing Surveys, 52(6), 112-130.
6. Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from <https://bitcoin.org/bitcoin.pdf>.
7. Wood, G. (2014). *Ethereum: A Secure Decentralized Generalized Transaction Ledger*. Ethereum Project Yellow Paper.
8. Meng, W., & Li, J. (2022). *Smart Contract-Based Secure Scheduling in Cloud Computing*. IEEE Access, 10, 9832-9845.
9. Kshetri, N. (2021). *Blockchain and Trust in Cloud Computing: A Comprehensive Study*. Journal of Cloud Computing, 9(1), 32-48.
10. Banerjee, S., & Roy, D. (2020). *Task Scheduling in Cloud Computing: Security Challenges and Solutions*. Proceedings of the International Conference on Cloud Security, 201-213.
11. Gupta, H., & Kumar, P. (2021). *Artificial Intelligence-Driven Optimization in Blockchain-Based Task Scheduling*. Future Internet, 13(5), 102-117.

-
12. Patel, A., & Shah, R. (2022). *Enhancing Cloud Task Management with Decentralized Smart Contracts and AI Models*. Journal of Cloud Security Research, 7(4), 78-94.
 13. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). *Fog Computing and Its Role in the Internet of Things*. Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 13-16.
 14. Hyperledger Foundation (2022). *Hyperledger Fabric Documentation*. Retrieved from <https://hyperledger.org/fabric>.
 15. Vitalik Buterin (2013). *Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform*. Retrieved from <https://ethereum.org/en/whitepaper/>