International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I6P107

Volume 74 Issue 6, 104-119, June 2026

Original Article

BLOCKVAL: A Hybrid Consensus Algorithm for Document Validation in Private Blockchain Networks

Shibron Arby Azizy[1], Aditya Kurniawan[2]

[1,2]*Department of Computer Science, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia.*

[1]*Corresponding Author : shibron.azizy@gmail.com*

Received: 23 May 2025 Revised: 04 April 2026 Accepted: 20 April 2026 Published: 27 June 2026

Abstract - *The process of document validation is important in government agencies to support integrity and ensure the reliability of administrative and legal documents. However, classical validation systems that rely on centralized databases and manual auditing tend to be inefficient, insecure, and subject to tampering. Based on the literature, this article proposes BLOCKVAL, a multi-disciplinary, government-specific hybrid blockchain-based validation system. BLOCKVAL has a two-layer consensus structure with Proof of Authority (PoA) at the second layer for quick and efficient document review, and Proof of Work (PoW) at the first layer for immutable data and improved security. The hybrid structure increases integrity, transparency, and tamper resistance of the data while preserving the power of PoA consensus. The system framework features hierarchical role-based DApps, digital signatures with CAs, and a dual-ledger integrated with the system that facilitates concurrent validation and mining. BLOCKVAL also provides an automated means of scanning for tampered or inconsistent ledgers and restoring them; it allows selective repair or complete replacement of ledgers depending on how severe the harm is. Experimental results indicate that the average transaction time of the hybrid model is constant, 4.31 seconds, similar to the PoA-only model. The average PoA block creation time for the hybrid model is 0.371 seconds, and it is 0.677 seconds for PoW on average over 4 slave nodes. For 200 instances of simulated tampering, the system achieved 100% successful recovery, with an average repair time by the system of 1.50 seconds. These results show that BLOCKVAL can achieve an optimal trade-off between efficiency and security. The suggested model could be employed to securely verify documents in a regulated environment, such as in the Ministry of Public Works, Republic of Indonesia.*

Keywords - *Private Blockchain, Document Validation, Proof of Authority, Proof of Work, Government Systems.*

1. Introduction

1.1. Background

Document validation is one of the most important activities of government institutions to guarantee the validity and correctness of projects authorized by the government. Thousands of contracts document each year are managed by the Ministry of Public Works in Indonesia, and this validation is needed in order to support monitoring of compliance and keep audit errors at bay. However, so far, the validation system is centralized and manual, and it introduces inefficiencies, increases the cost of production, and also poses security risks. The safeguarding of data, particularly in the information security world, is generally framed in the domain of information security using the core ten principles of the CIA, which are confidentiality, integrity, and availability [1]. Among these three concepts, preserving the file integrity is particularly important as it acts to maintain the form of written documents as full, unchanged documents, so people can be dependable when they have files [2]. The integrity of existing files has emerged as an important element of information security in the cloud and decentralized environments [3]. Pinheiro et al.'s study establishes that file integrity can be preserved well with blockchain, as the immutability and traceability of project data are maintained intact [4]. The validating system is still an issue despite its value. Conventional validation systems are typically centralized and human-centric, relying heavily on manual workflows that lack automation and transparency [5]. One of these reasons is that the required physical copies of various documents lead to a high operational overhead, which is primarily related to verification, file management (i.e., physical file), and audits [6]. Even after lengthy and costly certification processes, the final attested documents can still be forged digitally, highlighting the insecurity of traditional validation processes [7]. To address these challenges, blockchain technology has now emerged as a promising alternative. Blockchain represents a solution for document verification that is decentralized and resistant to tampering. Public blockchain based on Ethereum is expected to modernize the processes of approval and lower the dependence on physical documents. It

could also bring transparency for users and document validators more easily [5]. Public blockchains promote transparency but expose high transaction costs and latency problems. This constraint renders public blockchain solutions less adaptable in cost-sensitive environments, including government institutions.

1.2. Industrial and Regulatory Context

Document validation system is applied in the financial and administrative areas due to tight financial and administrative regulations in the government context. Usually, public blockchain platforms are based on gas-based transaction models in which transaction fees can vary according to network demand. This diversity of costs makes it hard to manage budget planning and financial accountability, especially when the cost of such expenditure must be legitimate and auditable. Government information systems should follow strict governance over participation through networks, beyond financial reasons. Administratively sensitive data cannot be processed on networks based on any free-for-all or anonymous access. Therefore, permissions-based blockchain architectures with nodes first identified and then authorized to join the network tend to be more suitable for an institutional deployment. For instance, Pericàs-Gornals et al. reported on a private blockchain system that could ensure the secure and transparent verification of COVID-19 certificates using the Ethereum Virtual Machine and PoA consensus. Their system focuses on privacy, immutability, and regulation compliance [8]. Similarly, Aldwairi et al. proposed using blockchain for their nostrification system to create digital certificates in awarding institutions [7]. They had established their own private blockchain network, built on the Proof of Existence mechanism through the Python v3 language, Flask web server, with Ganache as a test blockchain environment.

1.3. Research Gap and Contributions

In addition to permissioned architectures, a hybrid consensus model has emerged to build on the best of the numerous blockchain consensus tools. Cash and Bassiouni also introduced the two-tier blockchain architecture that used Proof of Work (PoW) and Proof of Authority (PoA) to promote the secure use of information [9]. PoW, natively implemented in Bitcoin, is secure for compliance since reconfiguring existing confirmed blocks is extremely computationally expensive [10]. On the contrary, PoA belongs to the Byzantine Fault Tolerant (BFT) family and enables the concord of a group of well-known validators at an earlier stage [11, 12]. Although these works demonstrate the potential applicability of hybrid consensus architectures, the incorporation of these architectures into document validation systems in government domains has not been widely investigated or explored. However, existing implementations primarily employ one consensus mechanism or hybrid consensus models aimed at the sharing of data rather than document validation. This work aims to fill the void with BLOCKVAL, a hybrid blockchain document validation system implemented for permissioned networks. According to the proposed architecture, PoA is a fast document validation software, and PoW then helps ensure the integrity of the ledger. The idea is to provide the system with greater resistance to tampering in the ledger, while at the same time allowing the system to continue its work of enforcing the credibility and efficiency that it requires to work for government organisations. In BLOCKVAL, based on what is already seen in cryptographic hashing, we apply a shared ledger for validation of documents over time to further make it more difficult than at some point in the future to have verification. The system also comprises digital signatures from Certificate Authorities (CAs) to add to evidence when the user attempts to edit a file. A QR-based stamp is also added to it, mainly to tie each authenticated copy to someone's personal identification and to make intrusive changes easier to verify.

2. Literature Review

There have been several research studies into blockchain-based solutions to secure the validation of documents. The public Ethereum blockchain has been suggested as a technique to maintain the integrity of academic credentials by offering an immutable record of stored information [13]. The problem is that public blockchains have high transaction costs, slow processing times, and are prone to network congestion, which makes them less suitable for government systems. Other methods have covered private blockchain implementations, including employing the Geth client on OpenStack or a combination of Geth and Parity clients to provide better security and control [6, 14]. Meanwhile, a custom private blockchain prototype was developed in Python, Flask, and Ganache, in which an extra Merkle Tree hashing layer was added to improve the validation of foreign-issued diplomas [7]. In the health sector, a blockchain-based system for COVID-19 certificate verification was introduced by leveraging the Rinkeby Test Network along with IPFS storage and Proxy Re-Encryption (PRE) to preserve privacy while maintaining the integrity of data [8]. Although this approach means that data privacy is preserved and forgery is avoided, the use of IPFS itself may cause data availability problems if nodes become inactive. Another study proposed a blockchain tax compliance framework based on asymmetric encryption on top of a public Ethereum network [15]. However, we know that gas fees for each transaction introduce additional financial costs, restricting applicability in cost-sensitive environments. In a different area, blockchain technology has been explored as a means of preserving forensic evidence and maintaining the fidelity and integrity of collected data [16]. Other research explores blockchain's possible role for government work, becoming more open and simpler to verify, so that institutions can be more accountable in their work [17]. Blockchain has also been used to protect digital evidence, such as marking crime scene photos with a cryptographic watermark and storing the associated information on a private blockchain to

prevent modification or loss of data [18]. Viewing the two reports as a whole, blockchain might seem like a tool that would bolster both security and trust in government work. A few of the findings are promising, particularly in fields that process important or sensitive information. However, employing blockchain in actual government offices is not necessarily simple. There are still concerns about regulations, the condition of the systems that are already running, and whether test results, performed in smaller tests, will work as well as they can when deployed on a larger scale. Blockchain may also be utilized in construction documents. M. Das et al. [19] provide for better handling of multiple types of files with off-chain procedures using Hyperledger Fabric and Merkle-Patricia Trie (MPT). Other studies implemented Ethereum smart contracts combined with IPFS to support digital approvals and minimize paperwork [5]. This method can illustrate a clearer, faster, and more transparent workflow. That said, there are challenges associated with public blockchains, such as transaction fees that do not fit well in a government atmosphere and the potential for delays when the public network is overloaded. Overall, previous studies show that blockchain can improve the security, transparency, and traceability of document validation systems. However, public blockchains still face challenges regarding volatile gas fees, rendering them unsuitable for use in a government environment. On the other hand, private blockchain providers running on third-party platforms are susceptible to service unavailability and the risk of unknown document storage. Unlike existing work that only examined one type of consensus, the BLOCKVAL model utilizes two consensus models, PoA and PoW, but separated into two interconnected layers. PoA makes use of the speed for document validation, while the PoW strengthens the security of the validation results. However, it is well known that this dual-consensus model has not been fully implemented in the government sector for document validation.

3. Proposed System and Methodology

3.1. Proposed Blockchain System

BLOCKVAL is a document validation system based on blockchain using PoA and PoW to secure data while still achieving speed in document processing. Three DApps work on each role on the blockchain. The validator DApps contain three nodes that verify document authenticity. The uploader DApps is a node used by document owners to submit files, check validation progress, and view tokens (CAs) to verify data integrity. When a document owner uploads a document, a hash is created using SHA-256 and stored on the chain. At last, the function of the Master DApps is quite different. Master DApps function more as a network administrator without participating in the document validation process. An overview of how all these parts are interrelated can be seen in Figure 1.

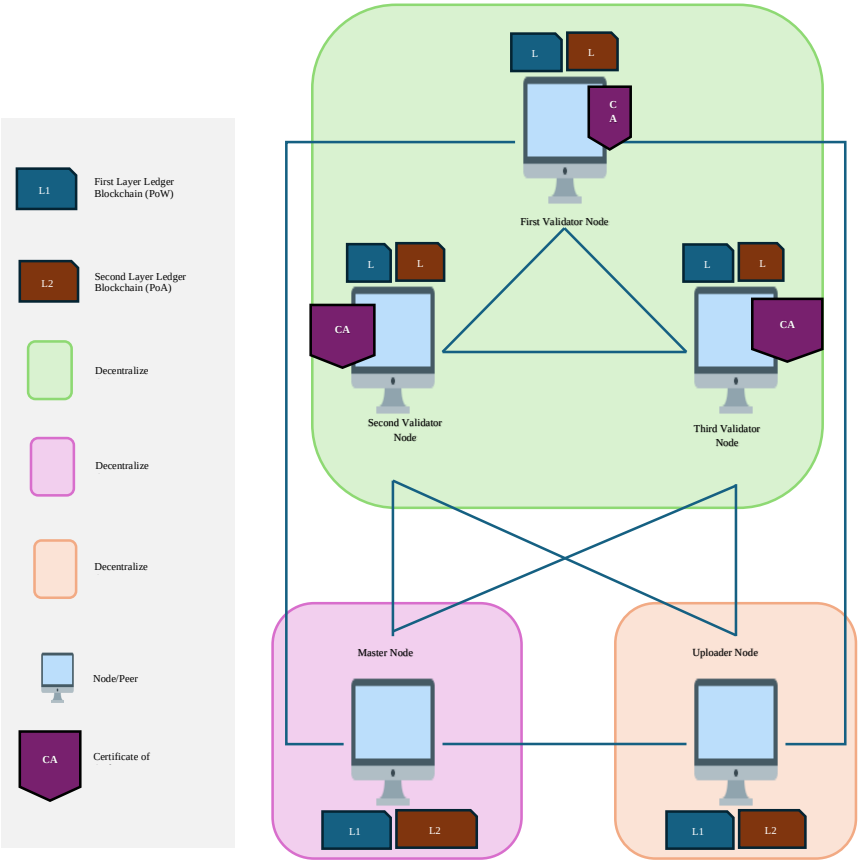

Fig. 1 Private blockchain architecture

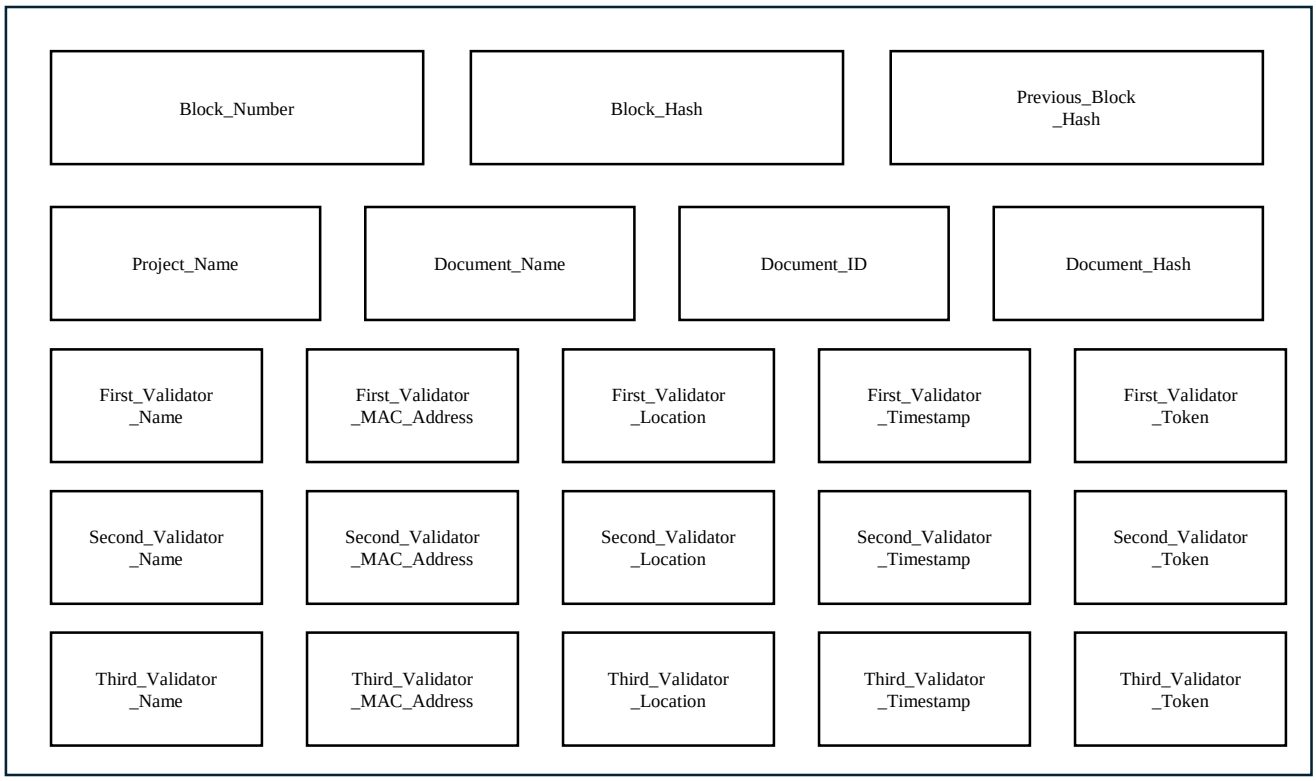

Fig. 2 Block structure on first layer ledger (PoW)

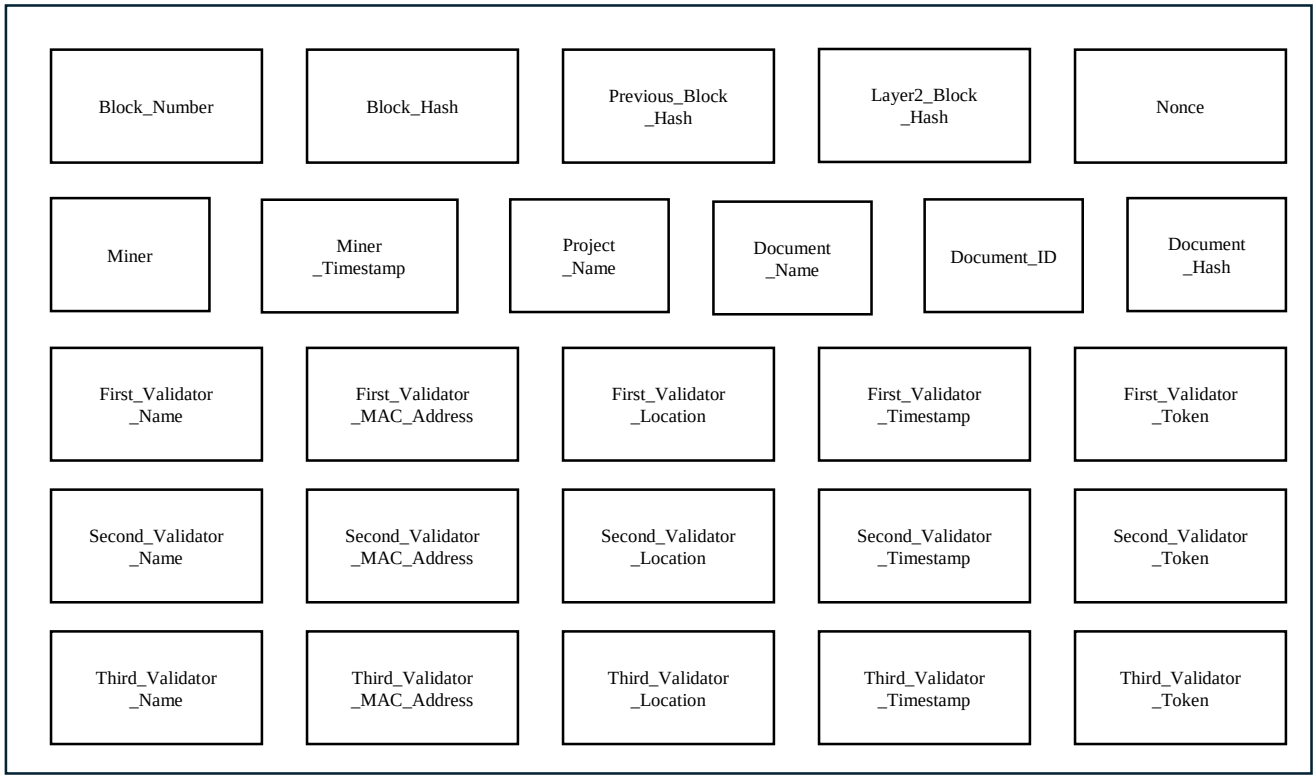

Fig. 3 Block structure on second layer ledger (PoA)

Figure 1 shows that each node stores its records on two ledgers. Document validation is handled first, and the results are stored in the PoA ledger on the second layer. Once this part is finished, the same information is rewritten to the PoW

ledger with some additional information added. This process takes a bit longer, but in the end, it provides another layer of protection for stored data, making it more difficult to tamper with later. These steps are designed to keep the validation process fast while still maintaining a more secure copy of the data on the other layer. The amount of information that a block in PoA and PoW ledgers is slightly different. Figure 2 shows that in the first layer, some information was included, like nonce, hash linking the first layer with the second layer, the name of any miner who did the work, and the time of the miner's step completing the miner. A PoA layer depicted in Figure 3 does not contain this information. This nonce is utilized in PoW, and the Layer2_Block_Hash makes the stored data more resistant to tampering. Miner information, on the other hand, only stores the node that finished the PoW process at the moment.

In Figure 4, PoA is with the second layer and PoW with the first layer. The PoA half is utilized initially as it provides a quicker way for the validators to prove documents. We then write their decisions to a smaller ledger, and after all validators agree that something is true, they write it down as an output to the second layer, to which the other nodes share it. This record, later, is hashed and copied again, saving it to the first layer for protection in the long run. We need more computing at the PoW step, and stored data is better protected. In this configuration, speed is the important aspect at the second layer to which the second layer is primarily interested, and it takes precedence for speed-based processing. The first layer provides a more secure record for any approved information. The process starts when the uploader node sends a document. The database would then store the file as an SHA-256 hash, and the hash would then be the document ID.

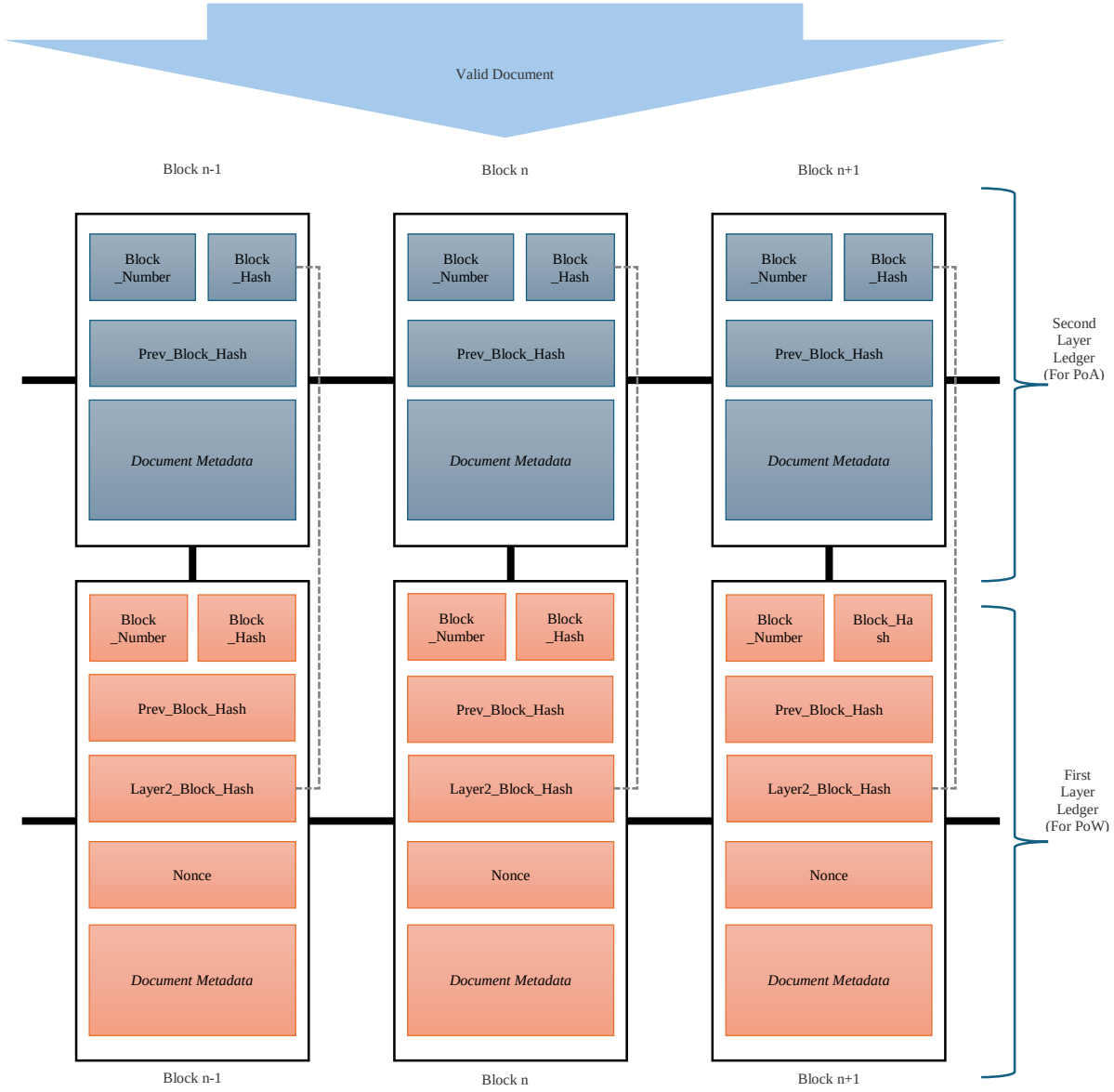

Fig. 4 Hybrid ledger PoA-PoW

Figure 6 summarizes just what takes place once a file is uploaded. Validators check if the document presented in the Validator DApp appears to be valid. If the document is accepted, each validator sends a digital certificate. The certificate constitutes proof for the verification and only holds the necessary information, such as the name of the validator, the MAC address and location, when and at what time it validates, as shown in Figure 5. All certificates are saved as Base64 on a disk for retrieval. When the verification is performed by all three validators, a new block is constructed, it is briefly verified, and it is fed to the ledger, which is in the second layer.

In case all validators agreed that the ledger was correct, the ledger is shared with every node on the network. At the same time, each node (except the Master Node) solves a PoW task for the second-layer ledger to add some security. Thus, if one node completes the work, it then sends its result on to the Master Node. As a result, the Master Node looks over what it finds and selects one to proceed with, and this decision is sent back again for verification by other nodes. Once the nodes have come to an agreement for use, the ledger is updated with a new block along with it and spread on the network.

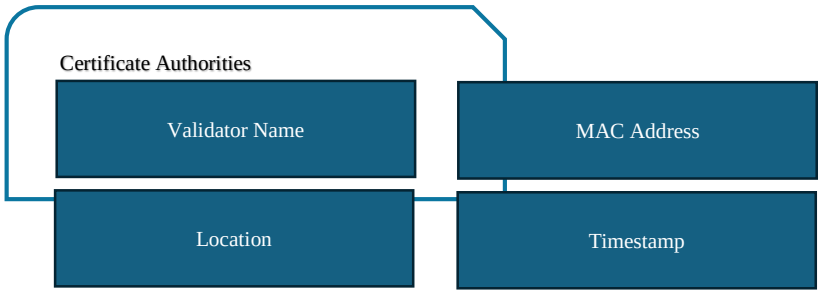

Fig. 5 Structure of Certificate Authorities (CAs)

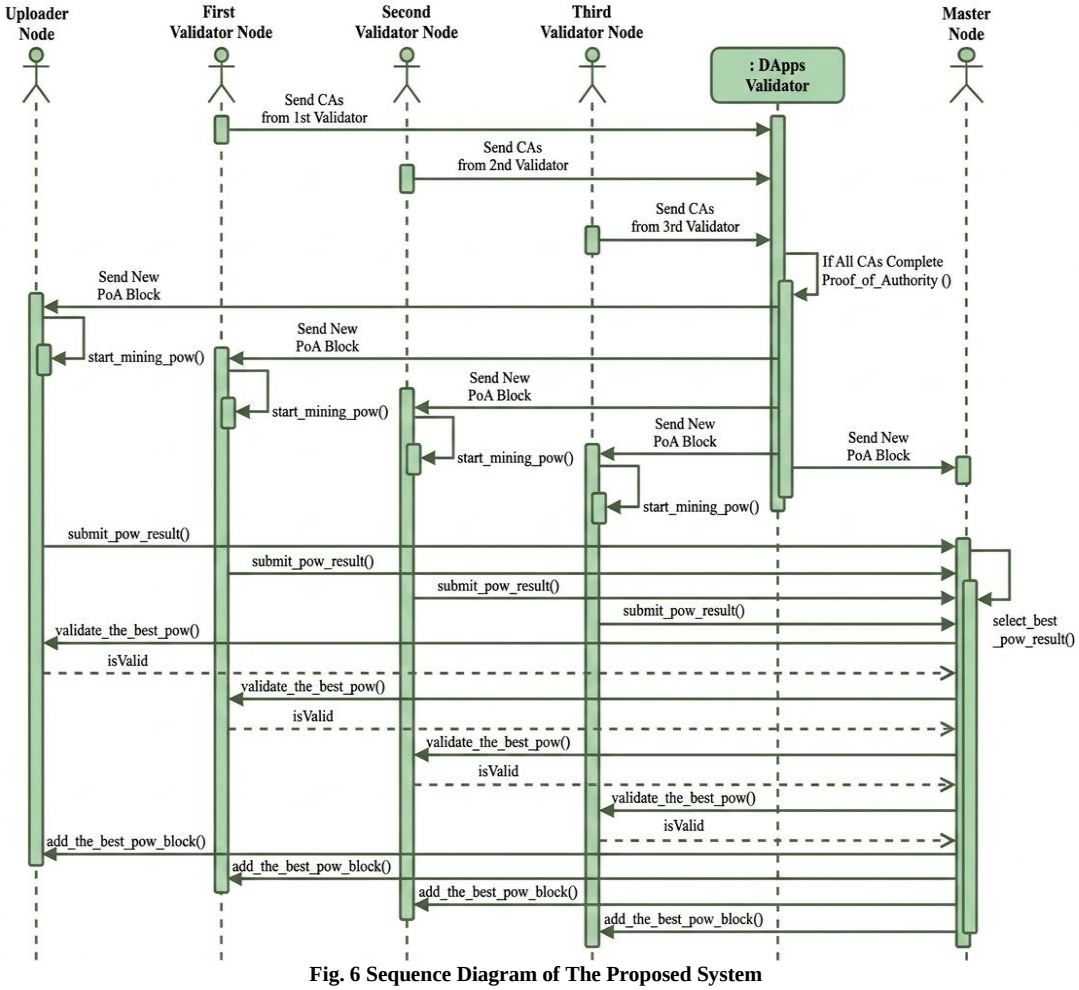

Fig. 6 Sequence Diagram of The Proposed System

3.2. Forking Issues

On public blockchains, there are situations where two miners end up finishing a block at almost the same time. When this happens, the system briefly holds two different versions of the chain, because both blocks are technically valid. [20]. This split is usually called a fork. After some time, the PoW mechanism will continue with whichever chain grows longer, since it represents the most accumulated work. The shorter chain is eventually dropped, and the network moves forward with the longest one as the main record.

3.2.1. Forking Issues in PoA Consensus Algorithm

In government systems, where rules are strict and data must be handled carefully, this branching issue needs to be considered because it can lead to missing or ignored data. To deal with this, the model introduces a simple queue for the validators. In the second layer (PoA), when two or more validators try to validate documents at the same time, they are asked to wait in line. Each validator will only continue after the one before it has finished creating its block, as shown in Algorithm 1.

Algorithm 1: Proof of Authority

```
FUNCTION validate_document(file_id):
  IF all three tokens are not empty AND not equal to
        'invalid':

    WAIT until the master node is idle
    granted = False
    WHILE not granted:
      SEND request TO master with
            in_progress=True
      IF response OK:
        granted = True
        BREAK
      WAIT 2 seconds

    // Validate the PoA ledger before writing a new block
    IF ledger_poa IS valid:
      CALL proof_of_authority(document)
      ATTACH QRCode TO PDF
      UPDATE document status TO 'isValid'=true
    ELSE:
      RETURN "PoA ledger invalid."
    SEND request TO master with in_progress=False

  ELSE IF one of the tokens is 'invalid':
    UPDATE document status TO 'isValid' = false

  RETURN "Validation process completed."
```

3.2.2. Forking Issues in PoW Consensus Algorithm

In the first layer, the PoW process works a bit differently from PoA. Here, the master node collects the mining results from the other nodes before a new block is added to the ledger. When one of the slave nodes finishes its mining task, the master node waits for a short moment to give other nodes a chance to submit their results if they finish around the same time. As shown in Algorithm 2, any node that completes the mining process will call the *submit_pow_result_to_master()* function. The idea is similar to the mechanism used in the PoA layer, which is to avoid forks in the PoW process so that all information is properly written into the ledger.

Algorithm 2: Proof of Work

```
FUNCTION start_mining_pow():
  IF mining_in_progress:
    RETURN
  IF new_pow_block_exists_in_other_node():
    RETURN

  mining_in_progress ← True
  block_data ← extract_data_from_last_poa_block()
  nonce, pow_hash ←
        perform_proof_of_work(block_data)
  new_block ← build_pow_block(block_data, nonce,
        pow_hash)

  IF new_pow_block_exists_in_other_node():
    RETURN
  IF local_ledger_pow IS valid:
    SEND new_block TO
          submit_pow_result_to_master()
  mining_in_progress ← False

FUNCTION submit_pow_result_to_master(block):
  IF block_number == last_finalized_block_number
    RETURN "Already added."

  IF block_number EXISTS in buffer:
    REPLACE with a block with the earliest timestamp
  ELSE:
    ADD block TO pending_pow_blocks buffer

  IF pow_selection_timer IS not running:
    START timer TO trigger
      select_best_pow_block() in 2 seconds

FUNCTION select_best_pow_block():
  best_block ← block IN pending_pow_blocks WITH
        earliest timestamp
  votes ← 0

  FOR EACH node IN network EXCLUDING master:
    SEND best_block TO node for
          validate_candidate_block
    IF node returns valid:
      votes += 1

  IF votes ≥ majority:
    IF best_block NOT EXISTS IN ledger_pow:
```

```
        INSERT best_block INTO ledger_pow
        BROADCAST best_block TO all nodes TO
                    /finalize_pow_block
    CLEAR pending_pow_blocks
```

3.3. Ledger Integrity and Repair Mechanism

To keep the ledger the same on all nodes, BLOCKVAL uses PoW not to validate documents but to check whether each node's ledger is still correct. If the system notices problems like broken hashes or blocks that do not match, it compares the ledger with others in the network.

When a difference is confirmed, the faulty ledger is replaced with a correct version so that every node stays in sync and protected from unwanted changes. BLOCKVAL uses two ways to fix a corrupted ledger.

When the damage is more than 30%, the system chooses the replace option, which removes the whole ledger and downloads a clean one from another node. If the corruption is 30% or less, it uses the repair method, where only the incorrect blocks are updated. This process is handled inside the validate_ledger() function, which is shown in Algorithm 3.

Algorithm 3: Validate the Ledger

```
FUNCTION is_valid_ledger(ledger_type):

    SET ledger ← fetch local ledger (PoA or PoW)

    IF validate_ledger(ledger):
       RETURN True  // Ledger valid

    SET invalid_blocks ← []
    FOR EACH block IN ledger (starting from index 1):
       If the hash or linkage is broken:
          ADD block. number TO invalid_blocks

    SET total_blocks ← length of ledger
    SET broken_ratio ← length of invalid_blocks /
           total_blocks

    IF broken_ratio > 0.3:
       longest_ledger ← longest valid ledger from other
                 node
       IF longest_ledger EXISTS:
          replace_ledger(longest_ledger)
          RETURN True
       ELSE:
          RETURN False
    ELSE:
       repair_ledger_blocks(invalid_blocks)
       RETURN True IF repairs succeed
```

```
FUNCTION validate_ledger(ledger):

    FOR EACH block FROM index 1 TO end:
       SET prev_block ← ledger[i-1]
       SET current_block ← ledger[i]

       IF current_block.previous_hash ≠ prev_block.hash:
          RETURN False  // Broken chain

       IF ledger_type == "poa":
          IF hash(current_block fields) ≠
                       current_block.hash:
             RETURN False  // Invalid PoA hash
       ELSE IF ledger_type == "pow":
          IF hash with nonce is invalid OR does not start
                       with "0000":
             RETURN False
          IF current_block.poa_hash ≠
                corresponds to the PoA block. hash:
             RETURN False

    RETURN True
```

4. Experimental Setup and Evaluation Metrics

4.1. Hardware and Network Environment

The system is developed using Python version 3 and the Flask web framework.

For the testing environment, MongoDB is utilized to store document data, while SQLite is used to maintain the blockchain ledger on each node. We use 5 nodes, including the Master Node, Uploader Node, and 3 validator nodes. The device used is:

1. Master Node: Windows 11, Intel Core i7-1165G7, 16GB RAM, 1TB SSD;
2. Uploader Node: Windows 11, Intel Core i5-8250U, 8GB RAM, 256GB SSD;
3. Validator 1 Node: VM Kali Linux 2024.1 with 4 processors and 2GB RAM allocation;
4. Validator 2 Node: VM Windows 10 with 2 processors and 2GB RAM allocation;
5. Validator 3 Node: VM Ubuntu 24.04.2 with 2 processors and 4GB RAM allocation,

All three virtual machines run on the master node at the same time.

4.2. Node Configuration

The evaluation was conducted in a controlled environment using five interconnected nodes, each assigned specific roles to simulate a realistic private blockchain network. The node configuration can be seen in Table 1.

Table 1. Node Roles and Functions in the Evaluation

Node Name	Type	Functions
Master Node as Master Node	Master Node	• Receiving mining results from other nodes. • Selecting the best PoW block • Broadcasting the finalized block to all nodes • Coordinating ledger synchronization and PoA validation
Uploader Node as Node Slave 1	Sleve Node	• Upload the Document • Start Mining Process • Validate the best PoW Block
First Validator Node as Node Slave 2	Sleve Node	• Validate the document and create the CAs • Start Mining Process • Validate the best PoW Block
Second Validator Node as Node Slave 3	Sleve Node	• Validate the document and create the CAs • Start Mining Process • Validate the best PoW Block
Third Validator Node as Node Slave 3	Sleve Node	• Validate the document and create the CAs • Start the PoA Process as the last validator • Start Mining Process • Validate the best PoW Block

4.3. Evaluation Plan and Metrics

We tested the hybrid BLOCKVAL technique both during the run, side by side with a PoA-only system, and with test performance and its conditions. In the PoA-only regime, only the PoA consensus is considered in the validation of transactions; hence, if validators are granted authorizations, they validate the entire transaction. On the other hand, the hybrid architecture adds one more PoW layer after PoA validation to each validated block in the same permissioned network, providing a dual consensus validation for each validated block. An additional tampering scenario was added to ensure system robustness. The tampering was made automatically from a particular endpoint that changed certain data in the ledger, imitating unauthorized modifications. A detailed description of this procedure is provided in the "*Security Integrity Evaluation*" section.

To study performance in a consistent environment, the experiment was repeated ten times, since the latency of the network could result directly in the processing time and block generation behavior. 200 test documents were sent in each run of experiments, giving rise to 200 related blocks in a sequence of 200 in each experiment. Through the repetition of the experiment over several iterations, it is possible to find the performance stability and consistency of the two configurations of the tested structures. For the most part, from our observations, we could see four issues that appeared crucial to pay attention to.

4.3.1. Security Integrity Evaluation

This section of the work was focused on tracking what occurs to the ledger once some areas have been purposefully altered. The idea was for the system to pick up small alterations to a block, or some extra data that should not be there. The larger question is whether the hybrid setup keeps the ledgers in the same state across nodes even when a few blocks no longer exist.

To perform this, several hand edits were performed on some blocks within the local ledger. They changed some of the hash values, and a couple of other blocks had the validator information updated as well. Once these changes are made to the ledger, the system will look at the ledger again by invoking the is_valid ledger() function.

Depending on the function produced, some of the impacted blocks may be repaired, or perhaps if cases were thought to be worse, the ledger could be swapped altogether. These actions follow the method that was written earlier in Algorithm 3.

4.3.2. Performance Evaluation

Transaction Time

This measurement looks at how long the whole transaction takes, starting from the moment a document is uploaded until it receives its QR stamp and is saved in the system. The time spent by the three validators during the checking process is included. The block for the PoA layer is then created by the last validator, and after that, PoW mining is carried out, and the ledger is synchronized across the nodes.

Block Creation Time

Here, the evaluation mostly tries to see how long the system needs to form a block, and it is not the same as the full transaction time. The interest is only in the moment when a block is actually created in both layers.

In the PoA part, we count the time from when the last validator begins the block-building step after all tokens are

collected. The time stops when the system can produce a hash that the block can use. This step runs fairly quickly in most cases, but the time taken can still depend on how quickly the validators complete their own work.

On PoW, that is different. The nodes start putting together a block based on the PoA result, and measurement will continue until a valid hash and nonce are found. For some cases, this can take longer, depending on how hard a particular mining operation actually is at that time.

Only the essential hashing and block-building operations are considered. Any additional delays caused by validation or node synchronization are not counted.

CPU Usage

In this section of the test, we only check how much CPU seems to be utilized by the system as a block is being formed in PoA and PoW. This uses Python's psutil.cpu_percent() for the readings.

For PoA, we take the CPU value during Proof_of_Authority() (Algorithm 1). At that time, the system is busy producing the block hash and inserting the block into the ledger. Only the last validator, in this instance Node Slave 4, is measured because it is the one that actually performs the PoA block.

The way PoW is checked is somewhat different. Here, the CPU usage is extracted from the slave nodes immediately after they finish the mining routine, where the hash and nonce are calculated.

his is due to the fact that the master node does not participate in this step and therefore is not used for its measurement.

5. Comparative and Security Analysis

5.1. Comparison with Existing Models

Table 2 compares the existing blockchain-based document validation systems among network type, cost dependency of transaction, model of consensus, and their influence on the system, and mechanisms through which their integrity is reinforced. Gas-based transaction models for public blockchain solutions based on the public network consensus are used for secure public network consensus secure transactions [5, 16]. Although computationally robust, such designs add cost variability that could not easily fit with government budget limitations. Many articles utilize permissioned architectures to remove gas fees and maintain institutional control. They deploy authority-based or BFT-based consensus-level rules in controlled networks [7, 8, 18, 19]. Moreover, while such approaches deliver predictable operational costs and governance control, their integrity model is based on validation agreement after blocks have been confirmed.

While building blocks and enforcing their integrity are the same, the proposed BLOCKVAL architecture separates them. Blocks are initially verified via PoA and then reinforced internally using internal PoW layers. In comparison with current permissioned solutions, we add computational resistance post-validation without public-chain gas mechanisms. By providing permissioned governance, cost predictability, and post-validation computing reinforcement, BLOCKVAL meets the operational and security needs pertaining to the operations of government spaces. Enterprise blockchain frameworks like Corda, Quorum, and others provide robust permissioned ecosystems, for example. However, they are more of general-purpose enterprise blockchain platforms than dedicated document validation platforms and thus do not meet the criteria for this study's comparative solutions.

Table 2. Comparative analysis of blockchain-based document validation systems

Reference	Network Type	Gas Fee Dependency	Consensus Model	Integrity Enforcement Basis	Computational Reinforcement
[5]	Public	Yes	PoS	Public PoW consensus	Not separate
[16]	Public	Yes	Hybrid	Public-chain anchoring	Yes (public chain)
[7]	Permissioned	No	Authority-based	Hash verification	No
[18]	Permissioned	No	Authority-based	Watermark + hash	No
[19]	Permissioned	No	BFT-based	Validator agreement	No
[8]	Permissioned	No	PoA	Authority trust	No
Proposed BLOCKVAL	Permissioned	No	PoA (block creation) + PoW (reinforcement)	Authority + computational anchoring	Yes (internal PoW layer)

5.2. Security Analysis and Threat Modeling

Allowing blockchain systems to minimise the ability to attack on an anonymous basis, although it still relies on the integrity of validators. In PoA-oriented environments, the block creation is delegated to authorized validators, and there are compromised or colluding validators who may approve

corrupt data if governance controls break down [21]. PoA also allows efficiency. However, since this is entirely trust-based, security guarantees are limited. Consensus design has traditionally been framed in terms of a trade-off between performance and security [22].

BLOCKVAL defines block creation rules by PoA, and PoW is added after block creation to improve the integrity of its ledger. This dual-layer nature of the system makes it more challenging to modify established records since any change would not only have to be performed without proper validator approval but also requires recomputing the PoW layer. Adding a nonce into the PoW stage introduces computational overhead, which results in significant resource utilization, since unauthorized replacement of the ledger data, as described in the security rules in Proof of Work systems [10]. The threat model investigated here covers validator compromise, unauthorized alterations to the ledger, and a partially corrupted ledger. For the purpose of assessing the resilience, tampering was simulated through an automated endpoint that changed specific ledger entries. BLOCKVAL also introduces an integrity reconciliation and restoration method to normalize in the event of anomalies, adding to the robustness of a protected system.

5.3. Scalability and Robustness Considerations

The scalability of BLOCKVAL derives from the PoA validation layer; it is at this layer that authorized validators determine the logic to create blocks. Moreover, in PoA, computation does not require much, hence transaction processing is efficient in permissioned networks of small to medium size. With the increasing number of validators, the coordination overhead and consensus communication could drive an increase in latency, and hence, are common drawbacks of authority-based systems [22].

PoW is applied only after the block is formed. It does not affect validation throughput in the architecture proposed. It imposes some additional computational overhead only on the reinforcement layer but allows the system to maintain a sensible processing pace while enhancing the ledger integrity.

From the standpoint of robustness, BLOCKVAL is designed for a controlled institutional environment, being able to handle partial node-to-node failure and inconsistencies. As block approval is centralized with the known validators, the network can continue to function as long as a sufficient number of authorized nodes remain active. With PoW reinforcement and integrity checks in addition, this PoW strengthening makes blocking unauthorized blocks harder and also allows restoration if corruption has been discovered.

Although this current assessment had been performed on a few nodes, the nested distinction between validation and reinforcement suggested an architecture that could scale horizontally with usual governmental deployments without sacrificing operational continuity.

6. Results and Discussion

6.1. Security Integrity Evaluation

Figure 7 shows the real way that the system reacts deliberately when the blockchain ledger is amended. Several updates were made to the local ledger in this evaluation for an integrity issue, such as altering hash values or interrupting the block order, in order to simulate an integrity problem. After adding these changes, the system invoked the is_valid_ledger() function. Depending on what it found, the function attempted to fix the affected portions or replace the ledger entirely. The choice between these two actions was based on the rules mentioned previously in Algorithm 3.

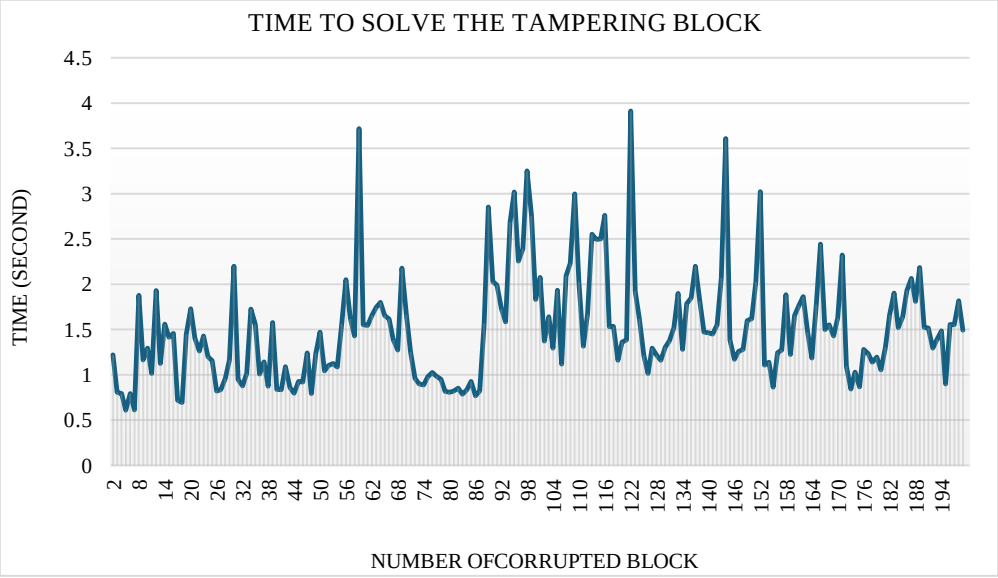

Fig. 7 Time to Solve Tampering Block

Through all 200 intentionally altered blocks, all of them were ultimately repaired. According to the average recovery time of ~1.50 seconds. In a handful of cases, the time was longer, and the longest was 3.91 seconds. This indicates that the recovery time may vary over time and is dependent on what has changed.

All 200 blocks were fixed, and the ledger was restored to the intended state across the nodes. The proposed system can still repair altered blocks when viewing the results. There are no manual adjustments in restoring the ledger. This is helpful for storing data with accuracy.

For verification of government documents, the restoration of the ledger itself can be very useful, in that one cannot leave the data corrupt for an inordinate duration of time.

6.2. Performance Evaluation

6.2.1. Transaction Time

Figure 8 shows how long each block takes to finish in both the PoA-only setup and the hybrid one. In the PoA setup, they travel over the length of a unit pretty far. Some blocks take almost 1.41 seconds, while others can be more than 22 seconds. The hybrid setup still has time range changes, but the change of time range here is smaller, around 1.98-15.59 seconds. Differences in these times can also come from a few factors. The network can also slow down, or the validators sometimes do not talk at the same time. In other examples, there are nodes that are just busy with other things. The hybrid configuration also maintains the delay to a manageable level after adding PoW, as the mining portion works in the background and does not halt the document validation.

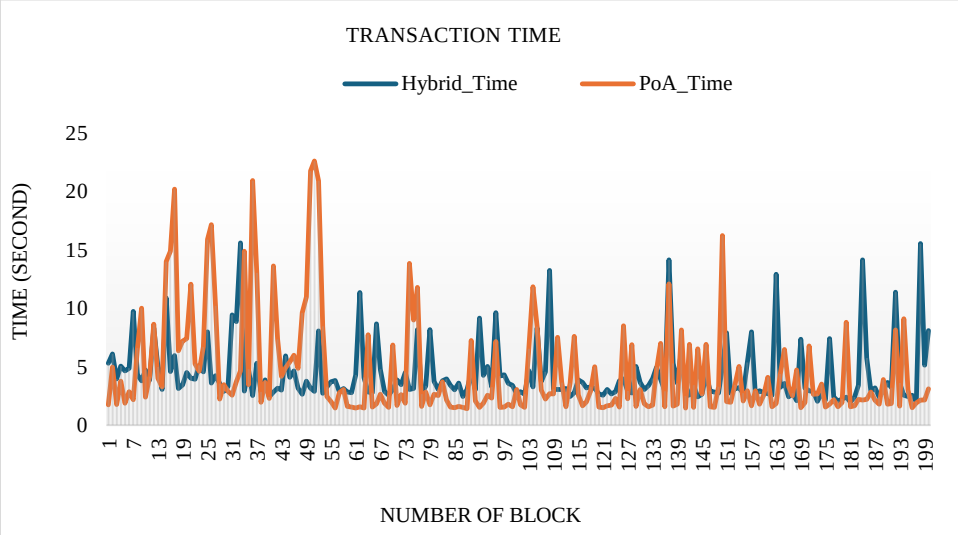

Fig. 8 Actual Transaction Time

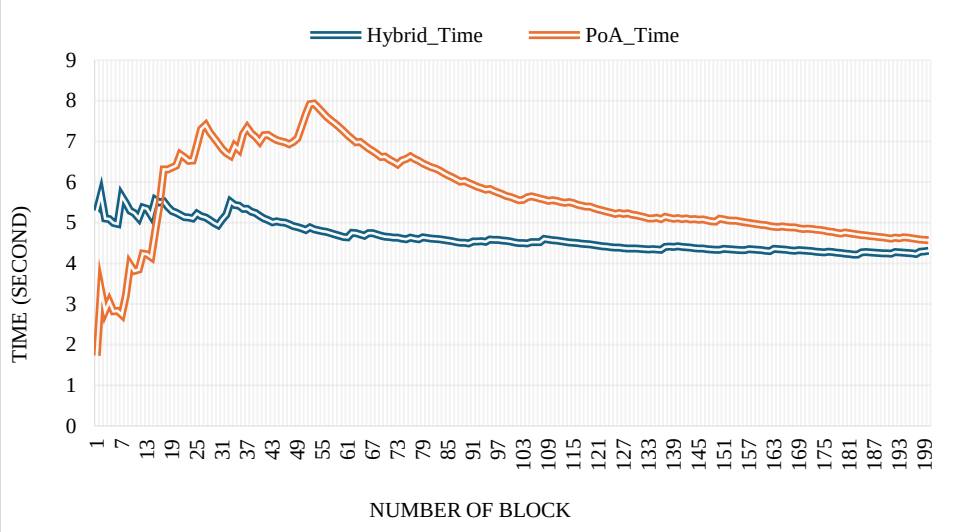

Fig. 9 Preprocessed Transaction Time

Figure 9 returns the same transaction results, but this time the values seem smoother, as they were averaged step by step. The PoA-only setup seems faster initially in the 200-document test.

After a few more documents have been applied, the figures of the hybrid model settle to a value of around 4.31 seconds on average. The PoA-only setup stands slightly above that, at about 4.57 seconds.

Even though you add a layer of security, the hybrid version is slightly further along the average, which overall holds for longer periods. The hybrid approach looks great, at least for performance, and is a great extra layer of protection.

6.2.2. Block Creation Time

Figures 10 and 11 allow a clearer view of the time period required to create a block after installing both configurations. The PoA block for Node 4 in the hybrid system, in turn, takes 0.371 seconds on average, which is a little slower compared to 0.339 seconds for Node 4 from the PoA-only setup. This slight increase could be due to the system starting the start_mining_pow() thread immediately after the PoA block is completed, and extra background work that may be happening. For PoW, each node is not equal in time. In the other direction and on the other side, the average time is 0.677 seconds, with the four slave nodes taken to finish. Node 3 is faster than the other three nodes (around 0.534 seconds), and Node 4 takes roughly 0.814 seconds to complete things.

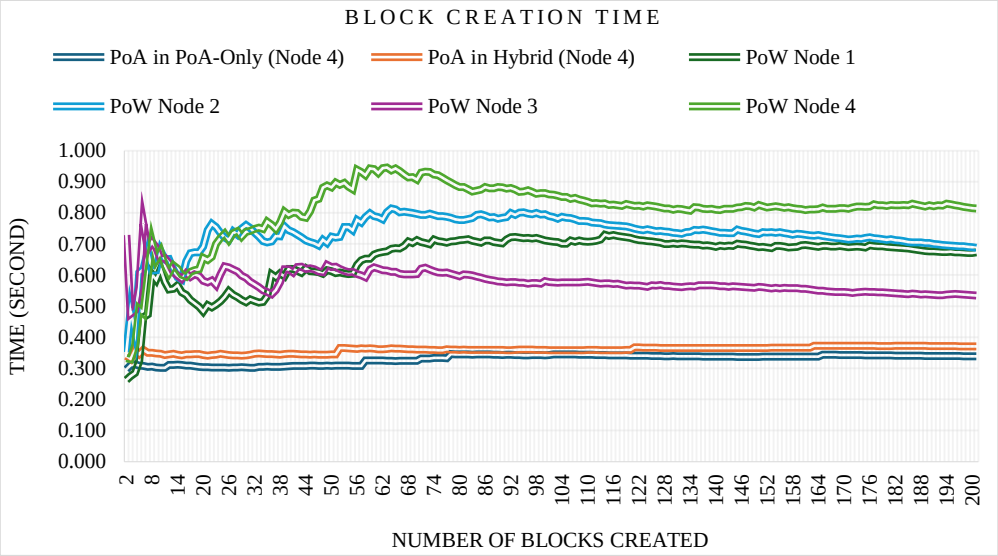

Fig. 10 Block creation time

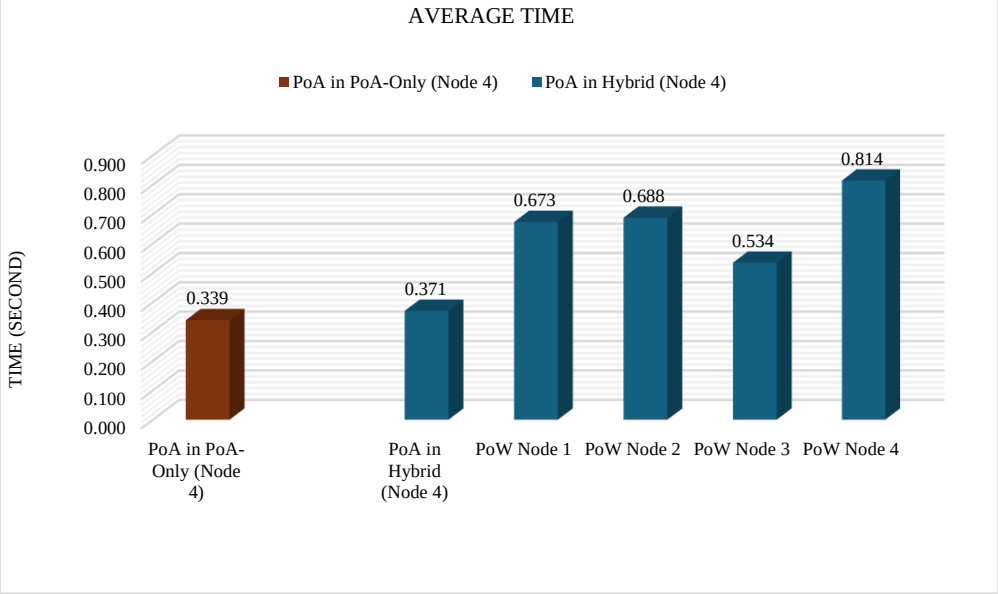

Fig. 11 The average of block creation time

This reason should be anticipated, as the PoW takes a lot more computing power to find the correct hash and nonce, and is therefore one and a half, if not two, times slower than PoA. Despite this additional delay, PoW is still a means to ensure the data is more secure because every block has to traverse a distributed agreement process. While the hybrid setup does seem to preserve the speed advantage associated with PoA, the better protection associated with PoW is thus made available. Since the PoW work runs independently in the background, it does not slow down the main validation flow, as illustrated by the stable results presented earlier in Figure 8 and Figure 9.

6.2.3. CPU Usage

Figure 12 shows how the CPU usage changes over time when blocks are created in both PoA and PoW. From the trend, PoA seems to keep its CPU usage low throughout the 200 blocks, which suggests that the process does not put much strain on the machine. PoW, on the other hand, uses noticeably more CPU because the nodes need to search for the right hash and nonce during mining. This general pattern is also mentioned in a study by Blanco et al., where PoW systems were found to require more computation than PoA [23].

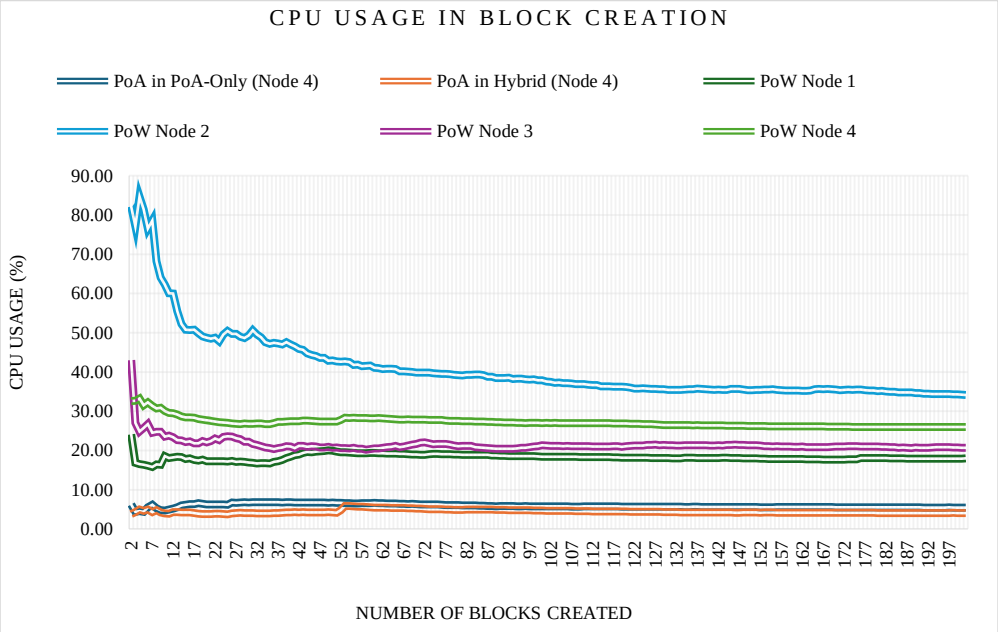

Fig. 12 CPU Usage in Block Creation Process

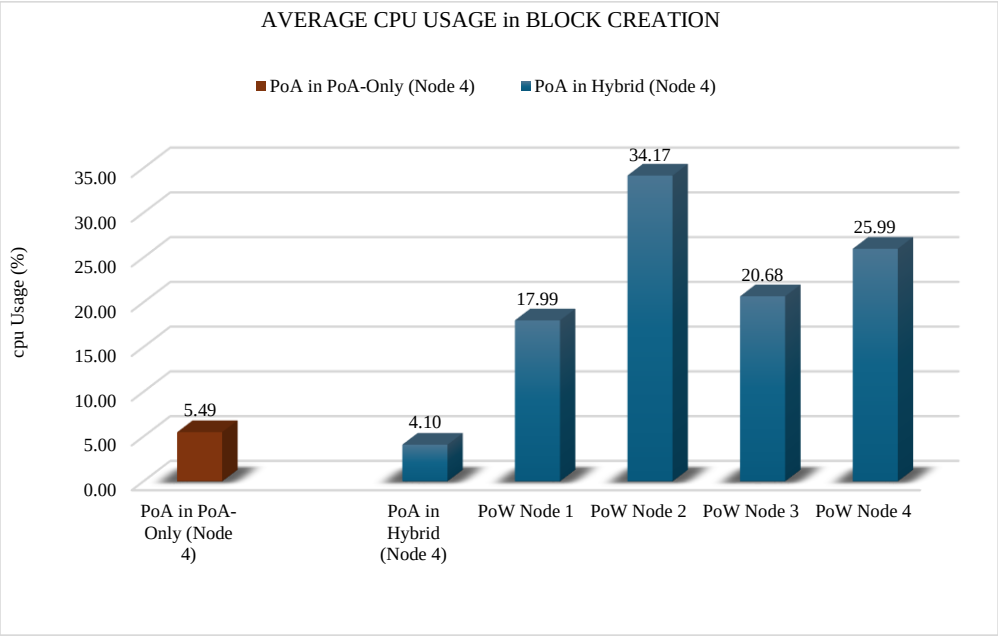

Fig. 13 Average CPU Usage in Block Creation Process

For easier comparison, Figure 13 illustrates the average CPU consumption of each node through the 200 blocks. In the PoA-only setup, when Node 4 created PoA blocks, the average usage was 5.49%. Now in the hybrid setup, the same PoA step uses slightly less CPU, less than 4.10%. However, during the PoW stage, the same node became higher, about 25.99%. This shows PoA is yet a lighter process. However, PoW loads an extra burden on top of it because this process requires mining. Figure 13 further demonstrates that the hybrid model exhibits another pattern for node 1, which results in an average performance 17.99% better than the other nodes. So, one possible reason is that node 1 executes on physical hardware, and the other three nodes run on virtual machines. This difference implies that virtualization increases CPU usage during mining, and hardware type can directly influence consensus performance.

7. Conclusion and Future Work

Two-layer blockchain-based system: fast validation using PoA, stronger protection using PoW afterward. The idea is to split the processes in such a way that the system will not slow down while maintaining a better ledger. The results matched our expectations, with an average transaction time of 4.31 seconds and block creation times of 0.371 and 0.677 seconds for PoA and PoW, respectively. The next main test revealed that all 200 ledger tampering schemes could be recovered in an average time of 1.50 seconds. In our opinion, the addition of PoW to PoA significantly increased its ability to detect a manipulation without compromising the speed of normal validation. Despite these promising results, several limitations remain. The current implementation is designed for controlled institutional environments, typically within local office networks or securely interconnected offices via VPN. It is not intended for large-scale public blockchain deployment. Future work may therefore explore performance optimization and scalability evaluation across geographically distributed institutional networks. Other improvements that can be made to upgrade this model are more optimized workloads, which can minimize computing resource consumption in the mining operations. Furthermore, version control can be implemented in order to perform document editing within the system.

Acknowledgments

The authors would like to express our gratitude to the Ministry of Public Works of the Republic of Indonesia for the information and insights provided regarding the document validation process and their support for the testing process used in this research. We also express our gratitude to the BINUS Postgraduate Program for facilitating the research and providing academic supervision.

Author Contributorship

Shibron Arby Azizy led the conceptual design, system development, experimentation, data analysis, and manuscript writing. Aditya Kurniawan contributed to research supervision, methodology refinement, technical validation, and manuscript revision.

Data Availability

All source code, experiment scripts, and sample datasets used in this study are publicly available at the following GitHub repository: https://github.com/sazizy/blockval. The repository is provided exclusively for academic and research purposes. The use of any part of the code or data for commercial applications is not permitted without prior written consent from the authors.

References

[1] Fatimah Alkhudhayr et al., "Information Security: A Review of Information Security Issues and Techniques," *2019 2nd International Conference on Computer Applications and Information Security (ICCAIS)*, Riyadh, Saudi Arabia, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Chai Kar Yee, and Mohamad Fadli Zolkipli, "Review on Confidentiality, Integrity and Availability in Information Security," *Journal of ICT in Education*, vol. 8, no. 2, pp. 34-42, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Bin Liu et al., "Blockchain based Data Integrity Service Framework for IoT Data," *2017 IEEE International Conference on Web Services (ICWS)*, Honolulu, HI, USA, pp. 468-475, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[4] Alexandre Pinheiro et al., "Monitoring File Integrity using Blockchain and Smart Contracts," *IEEE Access*, vol. 8, pp. 198548-198579, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Sanil Gandhi et al., "A Blockchain-based Data-Driven Trustworthy Approval Process System," *International Journal of Information Management Data Insights*, vol. 3, no. 1, pp. 1-10, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Pavitra Haveri et al., "EduBlock: Securing Educational Documents using Blockchain Technology," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, pp. 1-7, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Monther Aldwairi, Mohamad Badra, and Rouba Borghol, "DocCert: Nostrification, Document Verification and Authenticity Blockchain Solution," *2023 Fifth International Conference on Blockchain Computing and Applications (BCCA)*, Kuwait, Kuwait, pp. 652-657, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Rosa Pericàs-Gornals, Macià Mut-Puigserver, and M. Magdalena Payeras-Capellà, "Highly Private Blockchain-based Management System for Digital COVID-19 Certificates," *International Journal of Information Security*, vol. 21, no. 5, pp. 1069-1090, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Michael Cash, and Mostafa Bassiouni, "Two-Tier Permission-ed and Permission-Less Blockchain for Secure Data Sharing," *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, New York, NY, USA, pp. 138-144, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *SSRN*, pp. 1-9, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[11] Xuefeng Liu et al., "MDP-based Quantitative Analysis Framework for Proof of Authority," *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Guilin, China, pp. 227-236, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Shashank Joshi, "Feasibility of Proof of Authority as a Consensus Protocol Model," *arXiv preprint*, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Shahriar Karim Shawon et al., "DIUcerts DApp: A Blockchain-based Solution for Verification of Educational Certificates," *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, pp. 1-10, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] M. Dhulavvagol Praveen et al., "Scalable Blockchain Architecture using off-Chain IPFS for Marks Card Validation," *Procedia Computer Science*, vol. 215, pp. 370-379, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Filip Fatz, Philip Hake, and Peter Fettke, "Towards Tax Compliance by Design: A Decentralized Validation of Tax Processes using Blockchain Technology," *2019 IEEE 21st Conference on Business Informatics (CBI)*, Moscow, Russia, pp. 559-568, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Sumit Kumar Rana et al., "Decentralized Model to Protect Digital Evidence via Smart Contracts using Layer 2 Polygon Blockchain," *IEEE Access*, vol. 11, pp. 83289-83300, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ibrahim Ramadan Abdelhamid et al., "Redefining Governmental Services Through Blockchain and Smart Contracts," *Mathematical Modelling of Engineering Problems*, vol. 10, no. 5, pp. 1515-1528, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Isaac Baffour Senkyire, and Quist-Aphetsi Kester, "Validation of Forensic Crime Scene Images using Watermarking and Cryptographic Blockchain," *2019 International Conference on Computer, Data Science and Applications (ICDSA)*, Accra, Ghana, pp. 1-4, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] Moumita Das et al., "A Blockchain-based Integrated Document Management Framework for Construction Applications," *Automation in Construction*, vol. 133, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Zibin Zheng et al., "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," *2017 IEEE International Congress on Big Data (BigData Congress)*, Honolulu, HI, USA, pp. 557-564, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[21] Norshakinah Md Nasir, Suhaidi Hassan, and Khuzairi Mohd Zaini, "Securing Permissioned Blockchain-based Systems: An Analysis on the Significance of Consensus Mechanisms," *IEEE Access*, vol. 12, pp. 138211-138238, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Muhammad Muntasir Yakubu et al., "A Systematic Literature Review on Blockchain Consensus Mechanisms' Security: Applications and Open Challenges," *Computer Systems Science and Engineering*, vol. 48, no. 6, pp. 1437-1481, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Gabriel Fernández-Blanco et al., "Design, Implementation and Practical Energy-Efficiency Evaluation of a Blockchain based Academic Credential Verification System for Low-Power Nodes," *Applied Science*, vol. 15, no. 12, pp. 1-42, 2025. [CrossRef] [Google Scholar] [Publisher Link]