LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

RESEARCH ARTICLE

Achieving Greater Decentralization with Atomic Ownership Blockchains

Zhuo Liu†‡

Abstract. This paper proposes Atomic Ownership Blockchains (AOB), a novel blockchain architecture designed to address scalability and decentralization challenges in distributed ledger systems. AOB introduces an approach where each atomic object is represented by an independent blockchain, potentially allowing for horizontal scaling and enhanced security. The system stores only ownership transfer records, which may enable parallel transaction processing and improved throughput. By eliminating traditional mining and voting mechanisms, AOB aims to mitigate certain security risks while proposing an implicit consensus mechanism for resolving forks. The AOB architecture could potentially support the digitization of real-world assets and enable decentralized applications involving shared or fractional ownership. This paper presents the theoretical framework of AOB, discussing its potential advantages and outlining areas for future research and empirical validation. Practical implementation and rigorous testing are necessary to fully assess its viability and impact on digital ownership paradigms.

1. Introduction

The Atomic Ownership Blockchains (AOB) framework represents a breakthrough in decentralized technology, addressing the critical limitations of scalability, energy consumption, and security that have plagued earlier blockchain systems like Bitcoin. By leveraging a unique atomic design, AOB manages to deliver high levels of security while maintaining simplicity. This innovation aims to address the "impossible triangle" of blockchain technology, enabling it to serve not only as a decentralized cryptocurrency for the world's 8 billion people but also as a versatile platform for recording various types of transactions beyond currency.

Bitcoin,[1] often celebrated as the gold standard of decentralization,[2] has been instrumental in providing a trustless and censorship-resistant payments network.[3] However, a deeper analysis reveals substantial limitations in its decentralization claims. The proof-of-work (PoW) consensus mechanism, which Bitcoin relies on, can be compared to a competition where miners bet computational resources to win rewards. This system prioritizes computational power over the integrity of the blockchain, allowing branches with more computational power to prevail, regardless of their "rightness."[4]

Originally, Bitcoin's design envisioned a decentralized network where individuals could mine blocks using their personal computers. At that time, the hashing power was relatively evenly distributed, giving some legitimacy to the concept of "voting by computational power."[5] However, the advent of specialized mining hardware concentrated hashing power in the hands of a few, professional mining pools. This centralization undermines the decentralized ethos of

† Z. Liu (liuzhuo2011@gmail.com) is an independent researcher from Beijing, China.
‡ 1J9swFhj76E7D1hoacusEA5wBK6ypVSUHL

Bitcoin, as decisions about blockchain forks are effectively made by a small number of powerful entities rather than a distributed network of independent participants.[6]

Another critical issue with PoW is its inherent randomness. The principle of the longest chain, which guides miners to consolidate their efforts on the most extended branch, can lead to arbitrary outcomes. This randomness means that the voting process in Bitcoin is unprincipled and directionless, failing to reflect the collective views or conscience of the network's participants.[7]

Moreover, Bitcoin's reliance on miners to record transactions introduces a potential conflict between public power and private rights. Users must depend on miners to validate their transactions, which can lead to situations where miners, wielding public power, can influence or restrict users' private rights. This dependency exposes users to potential interference and undermines the trustless nature of the system.[8]

The well-known issue of double-spending attacks further complicates Bitcoin's security landscape.[9] While supporters argue that attackers with significant hashing power would avoid such attacks to protect their investments, this assumption overlooks the possibility of non-economic motivations.[10] Ideological motives, vandalism, revenge, and even mental illness could drive attackers to undermine the network, highlighting the need for a robust security mechanism that does not rely solely on economic deterrents.[11, 12]

The true value of blockchain technology lies in its ability to elevate security from an economic to a cryptographic level. Bitcoin can only try to strive towards economic-level security,[13] and even if it achieves that, it will still be vulnerable to attacks from irrational actors. To truly achieve decentralization, blockchain systems must provide cryptographic security that is immune to human unpredictability.[14]

AOB addresses these challenges by adopting a fundamentally different approach. Unlike traditional public blockchains, AOB operates through a public network of private blockchains. Each private blockchain is owned by an individual and can only be modified by its owner, while remaining visible to the entire network. This model allows for the transfer of ownership through blockchain transactions, maintaining a transparent and immutable record of ownership history.

This decentralized model is further enhanced by the ability to horizontally scale the network by increasing the number of blockchains. Each blockchain can represent a different asset or piece of data, enabling diverse business applications and enhancing the system's capacity without compromising decentralization.

AOB achieves a higher level of decentralization by focusing on atomic objects (indivisible, unique units) rather than a single macroscopic blockchain (for the whole system). Each blockchain in the AOB network represents a discrete unit, and the collective distribution of these units ensures a decentralized system. This approach eliminates the need for consensus algorithms, reduces the influence of random factors, and provides robust security against attacks.

By solving the fundamental issues of decentralization, scalability, and security, AOB stands as a potentially advantageous alternative to previous blockchain technologies. It offers a truly decentralized infrastructure that can support a wide range of applications, marking a significant advancement in the evolution of blockchain technology.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

2. Theoretical Framework

2.1. Basic Model—This study explores a unique type of private blockchain that, unlike typical private blockchains, operates in the public domain and is visible to the entire network, akin to a public blockchain. However, each private blockchain has a designated owner who is the sole entity authorized to add blocks, while others can only read.

A significant distinction of this private blockchain model is the ability to transfer ownership. When the current owner adds a block indicating a transfer to another individual (*e.g.*, "I transfer this blockchain to Bob"), the network recognizes the new owner, who then gains the right to add subsequent blocks and can further transfer ownership (see Figure 1).

In contrast to public blockchains that employ explicit consensus algorithms involving multiple nodes, the proposed AOB system aims to achieve consensus through analysis of broadcast timing and order, potentially reducing the need for explicit node agreement. The single owner, possessing the private key, can sign and add blocks without the need for validation by others, enhancing efficiency and speed.

In the proposed AOB system, the designated owner of a private blockchain is granted the ability to transfer ownership without requiring consensus from other participants, potentially offering enhanced control over the blockchain's future state. This transfer is legitimate as long as it is recorded in a block, recognized by all nodes in the network. In a multi-chain system, multiple private blockchains can circulate independently among participants, allowing for decentralized operations. Each participant manages their own blockchain without affecting others, maintaining equal rights and preventing any single point of control. This system scales horizontally by increasing the number of blockchains, limited only by hardware capabilities, thus offering high scalability.

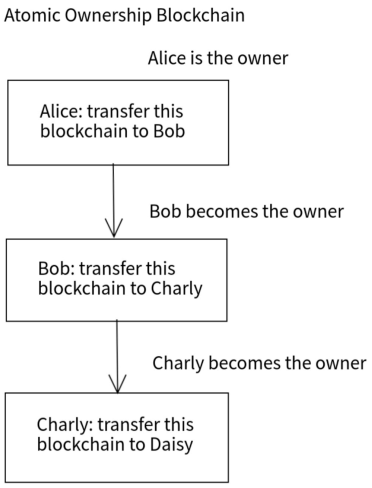

Fig. 1. AOB ownership transfer.

The AOB system allows for the assignment of specific meanings to different blockchains for various applications. One potential application is the representation of fixed-value units, similar to banknotes, which could serve as a basis for a cryptocurrency system. Unlike Bitcoin's single global ledger, these private blockchains record micro-objects, focusing on ownership transfers of individual units.

The proposed AOBs are designed to record ownership transfer history, potentially offering a mechanism for representing atomic ownership. The AOB system proposes a distributed approach to ownership tracking, potentially offering faster responses to ownership changes compared to global ledger systems.

Compared to UTXOs, AOB's atomic design offers a more streamlined approach to ownership transfer. While UTXOs require complex operations like splitting and consolidation during transactions, AOB banknotes maintain fixed denominations and operate independently, eliminating the need for horizontal connections between units. This simplified structure enables more efficient management and tracking of individual ownership transfers.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

This study suggests that private blockchains focused on microscopic objects, where each blockchain represents an atomic object, may offer an alternative approach to decentralization. This approach avoids the centralization issues inherent in macroscopic public blockchains, thus enhancing overall decentralization. The AOB system utilizes user devices as network nodes, with servers primarily functioning as connection facilitators, thereby establishing a decentralized network architecture. Logically, all nodes carry equal weight within the network. The system is designed for high flexibility, allowing nodes to join or leave the network at any time. However, users are encouraged to maintain long-term listening nodes online to ensure network stability and continuity. This approach aims to enhance decentralization through minimizing dependence on centralized infrastructure and utilizing distributed user device resources.

The server's role is to assist user devices in connecting and monitor message timing. Servers are stackable and replaceable. By distributing network responsibilities across user devices, AOB achieves a robust, scalable, and truly decentralized architecture that aligns with its core principles of atomic ownership and decentralized control.

Each AOB can represent a Non-Fungible Token (NFT). NFTs are inherently atomic, representing single, indivisible units of ownership, making them well-suited for AOB representation. An NFT based on AOB is native to the system, unifying AOB and NFT, and any business using AOB starts with NFTs.

Banknotes, though typically fungible tokens (FTs) due to their interchangeable nature, also have unique serial numbers, making them potential NFTs. If specific numbered banknotes are required as proof of rights, they function as NFTs. Similarly, NFTs can be used as FTs and restored to NFTs when differentiation is needed.

AOBs can record any atomic object with ownership, including financial instruments, commodities, and rights. These objects, commonly used as FTs, meet NFT requirements by adding unique serial numbers. AOB has the potential to be applied to cryptocurrencies, CBDCs, commodities, financial instruments, metaverse assets, NFTs, and more.

2.2. Security.

2.2.1. Elimination Voting—First, the focus should be on defending against Sybil attacks. Sybil attacks specifically aim at manipulating the voting process. Without any voting taking place, there would be no opportunity for Sybil attacks. In the case of AOB, being a private blockchain, the current owner has complete control over the state changes without considering others' opinions. Therefore, there is no need for voting in this scenario; in subsequent functions, it is crucial to exclude any voting operations as well.

However, eliminating double-spending attacks is not as straightforward. For instance, if Alice sends one of her AOBs to Bob and then adds another block in the same position, giving this altered blockchain to Charly, how should this situation be addressed?

2.2.2. Punishing the Account—In *private blockchains*, each position has a specific *owner* who alone can add blocks, preventing forks if the *owner* operates compliantly. If a fork occurs, it indicates cheating, as evidenced by the addition of multiple blocks at the same position. In such cases, the cheater (*e.g.*, Alice) is blacklisted, and others cease trading with her.

Blacklisting alone may not suffice as punishment, especially in anonymous decentralized systems where the cheater can create new accounts. To mitigate potential fraudulent activities, the system proposes implementing economic disincentives for account creation, such as

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

activation fees or minimum balance requirements. Additionally, limiting payments to one blockchain at a time reduces the potential gain from cheating.

While economic-level security can be achieved, not all attacks are economically motivated. Despite blacklisting, forks remain, posing the challenge of determining rightful ownership (*e.g.*, should the AOB belong to Bob or Charly?). This remains a critical issue to address.

2.2.3. Choosing the Branch—The detrimental effects of forks manifest in two primary aspects: firstly, they undermine the acceptability of the fork received by the initial recipient; secondly, the concurrent circulation of multiple forks can lead to monetary inflation. An effective solution leverages temporal order, stipulating that among conflicting blocks, the one broadcast earliest is deemed valid. Through the recording of broadcast times, nodes achieve implicit consensus (observation and inference of facts) on the validity of the first-broadcasted block without requiring explicit declarations from individual nodes.

Each node independently arrives at its own determination, which serves solely as its internal decision and does not influence the choices of other nodes. There is no necessity to communicate this conclusion to other nodes, and consequently, malicious nodes are unable to sway the judgments of others by transmitting erroneous information, thereby effectively preventing Sybil attacks. AOB eliminates any explicit or implicit voting mechanisms, thus obviating the requirement for a low-latency synchronous network.

The outcomes of fork selection do not necessitate strict network-wide consensus. As each blockchain represents merely a single atomic object, the impact on system operation is negligible, provided the proportion of disagreements among nodes remains sufficiently small. AOB ensures that attacks are economically disadvantageous by identifying and penalizing attackers, thereby preventing the occurrence of large-scale forks.

If an attacker broadcasts conflicting blocks almost simultaneously, nodes may receive them in different orders, which can hinder reaching consensus. When any node detects conflicting blocks, it broadcasts an alert with high priority to ensure all nodes are notified. Given this, the recipient can resolve the issue of indistinguishable block order by implementing a waiting period.

The recipient, as the primary party adversely affected by a fork, bears direct responsibility for security. In contrast, the threat posed by a fork to other nodes is considerably diminished, as they may never encounter this specific forked blockchain; furthermore, holding a divergent recognition of the fork would reduce their likelihood of accepting it. Thus, the recipient has a compelling rationale to wait for a duration sufficient to allow the network to acknowledge their accepted fork as the earliest broadcast. A waiting period equivalent to four times the network-wide broadcast time ($4t_0$) can ensure that security reaches a satisfactory level, analogous to awaiting six block confirmations after a Bitcoin transaction is recorded on the blockchain. Recipients may also elect to extend this period to attain greater confidence in security.

It is feasible to further concentrate the associated risks squarely upon the recipient. This can be accomplished by stipulating a minimum time interval, denoted as *tl*—on the order of several hours or potentially longer—that must elapse between consecutive transfer blocks. The recipient is then mandated to observe this *tl* interval before being permitted to append a subsequent transfer block to the blockchain. During this *tl* period, any risk associated with a fork becomes comprehensively apparent, and crucially, this risk cannot be offloaded by the recipient to another party. Considering that a user's assets may predominantly exist in the form of these AOBs, which are analogous to banknotes, the sheer volume of individual units held would likely

ledgerjournal.org
ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

be considerable. Consequently, there is typically no pressing imperative for the user to expend a specific, recently acquired AOB unit in the immediate aftermath of its reception.

Each node can estimate t_0. Assume the network parameters are:

Nodes	$N = 10^{11}$
Random connections per node	500
Connection Availability	95%
Transmission delay	$\tau \sim U(20\text{ ms}, 1000\text{ ms})$

We define t_0 as the time for a message to reach 99.99% of nodes with 99.99% probability. Effective degree $k\_eff = 500 \times 0.95 = 475$.

For a random network with N nodes and effective degree k_eff, the average distance (in hops) between nodes can be estimated using:

$d \approx \log(N) / \log(k\_eff) = 11 / \log(475)$ 4≈0.1 hops

To ensure 99.99% reliability, we add additional hops as a safety margin:

- For 99% coverage: d + 1 hops
- For 99.99% coverage: d + 3 hops

Therefore, critical path length 4≈0.1 + 3 7≈ hops.

With transmission delays summed across these 7 hops, and accounting for the upper tail of the delay distribution, t_0 is estimated to be approximately 7 seconds.

The maximum reception time difference between any two nodes for the same message is t_0. Therefore, when a recipient gets messages A and B at times Ta and Tb, where $Tb > Ta + 4t_0$, we can determine that:

- Any other node will receive message A no later than $Ta + t_0$
- Any other node will receive message B no earlier than $Tb - t_0$

Since $Tb > Ta + 4t_0$, we can establish that: $Tb - t_0 > Ta + 3t_0$

This means any node will receive message B at least $2t_0$ after receiving message A, ensuring a minimum interval of $2t_0$ between receiving the two messages. The nodes can be sure that almost all nodes receive A first, and are confident in its acceptability when they receive the fork announced in A later. This also eliminates the risk of inflation caused by forks. Given that t_0 is calculated with 99.99% confidence, the probability that any node will receive messages A and B with an interval exceeding $2t_0$ is greater than 99.98% (99.99% × 99.99%).

Malicious manipulation of broadcast timing is inherently difficult to achieve. Given that all blocks are distributed via broadcast and each user typically deploys multiple nodes across

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

diverse network domains, an attacker would need to control virtually all nodes to cause some nodes to receive a later-broadcast block prior to an earlier-broadcast one, particularly when the broadcast time difference exceeds t_0.

Consequently, attackers cannot create valid forks. If conflicting blocks are broadcast with a significant temporal interval, nodes will confirm the valid fork based on the broadcast sequence. Otherwise, if broadcasts are nearly simultaneous, the attack will be detected by the recipient during the waiting period and subsequently rejected.

The sole novel network security requirement introduced by AOB is the expectation that nodes maintain long-term online status. Users can readily fulfill this by deploying nodes on network servers. This requirement is not overly stringent; a user's offline status primarily creates difficulty for that user in selecting forks that occurred during their offline period, potentially leading to some inconvenience when subsequently receiving these blockchains, but without causing disruption to other users.

The AOB system aligns user actions with their consequences. This is achieved through several key mechanisms: perpetrators of double-spending attacks face direct penalties, ensuring accountability; recipients serve as the primary implementers of security measures, autonomously determining their desired level of protection while bearing the associated risks; and nodes that fail to remain online may struggle to select the correct fork during an attack, incentivizing consistent participation.

Consequently, when users deviate from established guidelines, they primarily incur significant trouble or losses for themselves, with minimal impact on others, reinforcing the system's self-regulating design.

2.2.4. Accomplices—We cannot trust Bob and Charly unconditionally. If Charly is Alice's accomplice and receives the forked block through a non-broadcast channel without timely broadcasting, then the block received by Bob will naturally gain recognition from the entire network. But what if Bob is also an accomplice? Bob and Charly might accept payment blocks received through non-broadcast channels.

If neither of them broadcasts, other nodes remain unaware of these two payments. When one of them eventually pays their fork to an honest person, one of the fork blocks is finally broadcast. The principle of time order still applies, which can help resolve the issue. Furthermore, the recipient realizes that they did not receive the previous block in time, so they can refuse the current block. When everyone refuses, Bob and Charly will find it difficult to pay out, equivalent to destroying their own banknotes.

2.2.5. Network Issues—This ideal situation relies on two assumptions: nodes are always online and the network is always connected. The reality may be more complex.

The proposed system relies on node connectivity to establish the order of potentially conflicting blocks. Consequently, it is important to remain online. Users can deploy listening nodes on servers to record block order, sharing them among trusted parties. This cost is minimal.

For disconnected users or new users, there are two methods to handle unrecognizable forking blockchains:

- Avoid unrecognizable forking blockchains by requiring the sender to switch to a recognizable AOB of the same value.
- Ask trusted parties, like local shop owners, if they recognize the fork. If they do, the AOB can be accepted and used for future payments.

ledgerjournal.org
ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

Network disconnections can lead to discrepancies among nodes regarding which block was broadcast first in the event of a fork. Imagine Alice initially broadcasts a payment block transferring to Bob, but the network splits immediately afterward, accidentally isolating Bob and other nodes that had already received the broadcast from Alice. Then Alice accidentally broadcasts a second block, resulting in two non-overlapping broadcast zones. Upon network reconnection, nodes in different zones may have conflicting information about which block was originally broadcast.

Although an attacker may not be able to control, predict, or even detect such network disconnections, these occurrences are plausible and must be considered. Therefore, recipients should exercise extra caution and only consider this scenario when dealing with critical blockchains, such as those representing high-value banknotes.

Extending the waiting time to the maximum network partitioning time plus $3t_0$ can solve these issues. During this period, the recipient should connect to the network through multiple means and test network connections. If all major websites are reachable, any disconnection is inconsequential.

The recipient of an AOB should wait a substantial amount of time—longer than any expected network split duration—before transferring it to others. This waiting period helps prevent effective forks from being rejected, thus avoiding potential losses for the initial recipient.

Broadcast time and maximum network partitioning time are estimated values; larger values increase security. For high denomination banknotes, waiting time can be extended.

Over time, even low probability events may occur, but the system will not collapse. This is because AOBs are microscopic blockchains (each for one atomic object), and consensus on an individual atom is not critical. If there is disagreement between two users about a banknote's fork, they can resolve it by trading with another banknote instead.

2.2.6. Security Example— An attacker, determined to undermine AOB even at personal cost, faces significant challenges. Before initiating an attack, they realize that profiting from double-spending is impossible and their account will certainly be lost, ensuring the attack results in financial loss. Despite this, they proceed, aiming to disrupt AOB.

They add two payment blocks at the same position on a blockchain, each sent to different recipients, and contemplates broadcasting. Their options are limited:

- If the broadcast interval is too long, allowing network consensus on the order, the later block becomes invalid.
- If the interval is too short, recipients will reject the blocks due to insufficient waiting time for security confirmation.
- The only scenario for successful payments is if both recipients are accomplices who do not broadcast. However, this prevents them from further transferring the received blockchains, as normal recipients would reject transactions with unknown previous payment blocks.

The attacker's sole remaining hope relies on an extremely improbable event: a network split occurring precisely during the attack, isolating the area where the first broadcast arrives, and lasting unexpectedly long. This could potentially cause a small portion of normal nodes to have a different understanding of a blockchain's state compared to other nodes.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

The probability of such an event is so minuscule that even if the attacker attempts this every second and accepts the losses, they might not encounter success in a lifetime. Moreover, they likely do not possess enough blockchains to sustain such frequent attacks.

In the event of a successful attack, the potential impact may be limited to a single atomic object, which theoretically should not significantly affect the overall system integrity. Furthermore, this small problem could potentially be rectified later.

This scenario aims to demonstrate the potential resilience of the proposed AOB security model, though empirical testing would be necessary for validation. It demonstrates how the system's design makes attacks not only unprofitable but also extremely unlikely to succeed, even when an attacker is willing to incur losses. The combination of broadcasting, waiting periods, and the atomic nature of the blockchains creates multiple layers of security, making AOB highly resistant to malicious activities.

2.3. Scalability—The key question of scalability is whether a cryptocurrency based on AOB can meet the daily usage needs of 8 billion people worldwide. If each person makes 10 AOB payments per day, this would generate 80 billion blocks per day, averaging nearly 1 million blocks per second.

2.3.1. Grouping—While hardware improvements may contribute to system performance over time, this study proposes software-based solutions to address immediate scalability challenges. We can divide AOBs into multiple groups for processing. For example, dividing AOBs into 65,536 groups based on the first two bytes of the ID can ensure balanced distribution. Each node focuses on a few groups, processing and storing only the data changes within these groups, significantly reducing the workload.

In real-world scenarios, individuals may engage in economic transactions with multiple social circles, requiring multiple groups:

- Family group: Transactions and financial sharing with family members.
- Company group: Business-related transactions with colleagues.
- Local group: Economic activities within a local area.
- Friend groups: Financial exchanges with different friend circles.

Global groups can be preset for ad hoc payments when no common focused group exists. These groups allow all nodes to process payments through AOBs within them. The number of AOBs in global groups is kept small, used only when necessary.

The proposed grouping mechanism utilizes the properties of microscopic blockchains to enable a form of vertical system partitioning, potentially enhancing scalability.

Theoretical analysis of time complexity suggests that the grouping of AOB chains may offer improved scalability, though empirical validation is necessary to confirm this hypothesis. Adding nodes within a group has a complexity of O(n2), where n is the number of nodes in the group, as each new node needs to communicate with all other nodes in the group. However, adding new groups only has a complexity of O(n), where n is the number of groups, since each new group only needs to focus on transactions within that group, independent of other groups. This linear scalability by increasing the number of groups is a key advantage of the AOB grouping algorithm.

2.3.2. Speedy Channel—The proposed Speedy Channel mechanism in the AOB architecture is designed to potentially enhance system scalability. Empirical studies would be required to quantify the extent of this improvement. It allows two parties to establish a temporary payment

ledgerjournal.org
ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

channel and conduct multiple transfers efficiently, without broadcasting every transaction to the entire network. This significantly reduces network load and improves throughput.

The Speedy Channel mechanism works similar to the Lightning Network for Bitcoin. Alice and Bob first establish a Speedy Channel on an AOB and pledge some tokens as the initial channel balance. They can then rapidly transfer balances back and forth within the channel by simply adding new blocks to the channel AOB, without broadcasting these transfers to the entire network. When they wish to settle and close out the channel, the final state is broadcast across the network.[15, 16, 17]

The proposed Speedy Channel mechanism aims to enhance scalability and potentially mitigate issues related to long waiting times caused by network partition risks. Since transfers happen within a private domain between the participating parties, recipients do not need to wait for an extended period to confirm transactions.

Moreover, Speedy Channels support cascading transfers and multiple participants and can facilitate data transfer by encrypting data with recipients' public keys. They provide a flexible payment network for various needs.[18]

While the core AOB protocol aims to provide decentralization and security benefits, the integration of Speedy Channels is intended to enhance its potential practicality and usability for real-world applications. Practical implementations and user studies would be necessary to evaluate these claims.

2.4. Banknote Generation—The AOB system employs a novel mechanism for banknote generation. Within this framework, any participant can create banknotes by computing hash values that satisfy predefined criteria, with the denomination of these banknotes being determined by the computational difficulty of the corresponding hash. This non-competitive Proof-of-Work paradigm differs significantly from systems such as Bitcoin, with its core distinguishing feature being isolation: the mining activities of individual participants do not influence one another.

Consequently, should the market value of these banknotes exceed the electricity costs incurred in their production, participants are incentivized to generate additional banknotes, thereby exerting a stabilizing influence on the notes' price. Conversely, when production becomes unprofitable, the output of notes is expected to decrease. This dynamic process intrinsically links the currency's value to the prevailing hash computing power. Prior to significant advancements in hardware efficiency, this mechanism effectively anchors the currency's value to electricity costs, as the requisite hash computations translate to a relatively consistent level of energy consumption.

2.5. Formal Definition.

2.5.1. Atomic Ownership Blockchain—An Atomic Ownership Blockchain is a tuple (*RB, (TB, RJB?)**), where:

- (*RB*) is a Root Block, representing the genesis block of the blockchain.
- (*TB, RJB*?)* is a (possibly empty) sequence of Transfer Blocks and Reject Blocks. There could be 0 or 1 Reject Block following each Transfer Block.
- (*TB*) is a Transfer Block, representing an ownership transfer.
- (*RJB*) is a Reject Block, representing a rejection to the previous ownership transfer.

ledgerjournal.org
ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

2.5.2. Root Block—A Root Block is a tuple (*ID, CB, MD, IO, CT*), where:

- (*ID*): A hash value computed over all other data fields of the Root Block, serving as the unique identifier of both the Root Block and the Atomic Ownership Blockchain.
- (*CB*): The hash of the whitepaper provides the creation basis for the blockchain.
- (*MD*): Meta-data specific to the blockchain, defined in the whitepaper, varying across different blockchains.
- (*IO*): The public key of the initial owner of the blockchain.
- (*CT*): The timestamp of the Root Block's creation.

Constraint:

- *ID* := Hash(*CB, MD, IO, CT*), where Hash is a cryptographic hash function.

2.5.3. Transfer Block—A Transfer Block is a tuple (*PH, TO, AT, SIG*), where:

- (*PH*): The hash of the immediately preceding block (either the Root Block, another Transfer Block or a Reject Block).
- (*TO*): The public key of the new owner to whom ownership is transferred.
- (*AT*): The timestamp of the Transfer Block's creation.
- (*SIG*): The digital signature of the current owner (the owner prior to this transfer block) over all preceding data in the block, also serving as the unique identifier (*ID*) of the Transfer Block.

Constraints:

- *SIG* := Sign_{SK}(*PH, TO, AT*), where (*SK*) is the private key of the current owner, and Sign is a cryptographic signature function.
- The Transfer Block's ID is (*SIG*).
- (*PH*) must correspond to the ID of the immediately preceding block in the blockchain.

2.5.4. Reject Block—A Reject Block is a tuple (*PH, TO, AT, SIG*), which is used by a recipient to reject a transfer due to personal configuration, where:

- (*PH*): The hash of the immediately preceding Transfer Block.
- (*TO*): The public key of the sender of the previous Transfer Block.
- (*AT*): The timestamp in the previous Transfer Block.
- (*SIG*): The digital signature of the current owner (the owner prior to this Reject block) over all preceding data in the block, also serving as the unique identifier (ID) of the Reject Block.

ledgerjournal.org
ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

Constraints:

- *SIG* := Sign_{SK}(*PH, TO, AT*), where (*SK*) is the private key of the current owner, and Sign is a cryptographic signature function.
- The Reject Block's ID is (*SIG*).
- (*PH*) must correspond to the ID of the immediately preceding Transfer Block in the blockchain (the prior (*TB.SIG*)).

2.5.5. Process: Receive New Transfer Block

Input:

- A Transfer Block, *TB* = (*PH, TO, AT, SIG*), received from a source node (*S*) via a broadcast network.
- The Atomic Ownership Blockchain, *BC* = (*RB*, (*TB*, *RJB*?)*), where (*RB*) is the Root Block and (*TB*, *RJB*?)* is the sequence of existing Transfer Blocks and Reject Blocks.
- t_0: The network-wide broadcast time.
- *tl*: The minimum time interval allowed between a Transfer Block and the previous block.
- (*CurrentOwner*): The public key of the current owner of the blockchain, determined as:
  - (*RB.IO*) if (*TB, RJB*?)* is empty, or Bn.TO, where Bn is the last Transfer Block or Reject Block in (*TB*, *RJB*?)*.
- (*LocalUser*): The public key of the user associated with the receiving node.

Output:

- The blockchain (*BC*) is updated (if the block is valid and no conflicts arise).
- Messages or alerts are broadcast to the network (*e.g.*, failure notifications, conflict alerts).
- A Reject Block is broadcast to reject the previous transfer.
- The source node connection is managed (*e.g.*, terminated on failure).
- A local decision is made if the new owner is the local node's user.

Process:

- Receive Transfer Block:
  - Receive *TB* = (*PH, TO, AT, SIG*) from source node (*S*) via the broadcast network.
- Check Parent Exists:
  - If not found B_n in Known Blocks:
    - Request B_n and trigger another Receive Process for B_n.
    - Wait till Termination of the Receive Process for B_n.
    - If cannot get B_n, Return and halt the process.

ledgerjournal.org
ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

        - Set *ParentDelayed* := *True*.
- Validate Standard Conditions:
  Validate the following conditions:
    - ID Validity:
        - Set *TB.ID* := *SIG*.
        - Verify that (*SIG*) is a valid signature over (*PH, TO, AT*) using the public key (*CurrentOwner*).
        - $\text{Verify}_{\text{CurrentOwner}}$(*SIG*,(*PH, TO, AT*)) = True
    - Current Owner Validity:
        - Verify that (*CurrentOwner*) is not blacklisted.
    - Parent Hash Validity:
        - If (*TB, RJB*?)* is empty, verify *PH* = *RB.ID*.
        - Otherwise, verify $PH = B_n.SIG$, where B_n is the last Transfer Block or Reject Block in (*TB, RJB*?)*.
    - TimeStamp:
        - Verify $B_n.AT < AT$

  If any condition fails:
    - Discard (*TB*).
    - Send a failure message to (*S*), containing the specific validation error (*e.g.*, “Invalid signature”, “Invalid parent hash”).
    - Terminate the connection with (*S*).
    - Return and halt the process.
- Check for Conflicting Blocks:
    - Check if there exists a prior Transfer Block TB_{prior} in (*TB, RJB*?)* at the same position (*i.e.*, with the same (*PH*)).
    - If a prior block TB_{prior} exists:
        - Consider (*TB*) invalid.
        - Blacklist the Signer:
            - Add the public key (*CurrentOwner*) (who signed (*TB*)) to a blacklist.
        - Broadcast Conflict Alert:
            - Construct an alert message containing at least (*TB*) and TB_{prior}.
            - Broadcast the alert to all nodes with high priority.
        - Evaluate Prior Block Validity:
            - Compute Δt, the time elapsed since TB_{prior} was received:
            - If $\Delta t < 2t_0$:
                - Consider the blockchain (*BC*) invalid (*e.g.*, mark it as forked and compromised).
        - Return and halt the process.
- Broadcast and Store the Block:
    - Broadcast (*TB*) to all connected nodes.
    - Append (*TB*) to (*TB, RJB*?)* in (*BC*).
    - Record the reception timestamp of (*TB*) as $t_{\text{receive}} := t_{\text{now}}$. Note it is not (*AT*).

ledgerjournal.org
ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

- Handle Local Ownership (if applicable):
  - If TO = *LocalUser* (*i.e.*, the new owner is the user of this node):
    - If $t_{Bn.AT} + tl > t_{now}$:
      - Add a Reject Block following (*TB*) and Broadcast.
      - Return and Halt Process.
    - If $AT + 2t_0 < t_{now}$ or $AT - t_0 > t_{now}$ (t_0 is also the maximum allowed clock difference between nodes.):
      - Add a Reject Block following (*TB*) and Broadcast.
      - Return and Halt Process.
    - If *ParentDelayed* (if a higher security level is configured):
      - Judge the reason for the delay of B_n according to the local Online History.
      - If B_n was not broadcast:
        - Add a Reject Block following (*TB*) and Broadcast.
        - Return and Halt Process.
    - Monitor Network Connectivity:
      - Continuously rebroadcast (*TB*) to ensure all connected nodes are received.
      - Continuously test network connectivity to ensure the node connects with the network.
    - Wait for Confirmation Period:
      - Wait for a duration of $4t_0$ (or longer, if a higher security level is configured).
    - Check for Conflicts:
      - During the waiting period, monitor for:
        - Any conflicting Transfer Blocks with the same (*PH*).
        - Any fork or conflict alerts including a conflicting Transfer Block broadcast by other nodes.
      - If no conflicts or alerts are received by the end of $4t_0$:
        - Acknowledge the transfer as successful.
        - Update the local state to recognize (*LocalUser*) as the new (*CurrentOwner*).

End Process

3. Key Features and Advantages

AOB achieves high decentralization through private rights without public power, fundamentally differentiating it from authoritarian public chains. Its security surpasses Bitcoin by eliminating double spending attacks. The atomic structure enables superior scalability compared to both PoW and PoS systems. Notably, AOB implements a groundbreaking stablecoin mechanism anchored to hash computing power through non-competitive PoW, where value stabilization occurs naturally as mining activity adjusts based on profitability. This creates

ledgerjournal.org
ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

the first truly decentralized stablecoin without relying on centralized reserves or complex algorithms.

4. Implementation and Demonstration

To validate the feasibility of the AOB concept and showcase its practical application, we have developed a prototype system and a demonstration page. These resources provide readers with an opportunity to intuitively understand the working principles of AOB while laying a foundation for further research and development.

This demonstration can be accessed at the following URL: `https://saintthor.github.io/aob/play_en`.

This demonstration page allows users to simulate various operations within the AOB system, such as creating Atomic Ownership Blockchains, transferring ownership, and verifying transactions.

To facilitate further research and development, we have made the source code of the AOB prototype system publicly available. The complete code repository can be found at the following GitHub repository:

`https://github.com/saintthor/decentralization`.

This code repository provides the core functionality used in the AOB demo page, including the creation and transfer of each blockchain, but excludes network protocols. In handling double- spending attacks, the system penalizes attackers. Due to the broadcast time being set to 0, demo nodes can more easily identify the order of fork broadcasts compared to real nodes and reject later-broadcast forks.

5. Conclusion and Future Research Directions

5.1. Summary of Key Findings—AOB represents a significant advancement in the blockchain landscape, offering a highly secure, efficient, and decentralized framework for various applications. From financial instruments and digital currencies to commercial applications and the burgeoning metaverse, AOB's proposed features suggest potential applications across multiple sectors, though further research is needed to quantify its impact. Its ability to facilitate decentralized cryptocurrencies, support Central Bank Digital Currencies (CBDCs), enhance loyalty programs, enable barter trade, and manage virtual assets and NFTs underscores its versatility and robustness.

5.2. Potential Areas for Further Research—While AOB offers numerous advantages, several areas require further research to fully realize its potential. Innovations are needed to address current limitations and enhance the scalability, interoperability, and user experience of AOB-based systems. Potential areas for further research include:

- Evidence Storage: AOB is less convenient than centralized blockchains for evidence storage applications, requiring exploration of how to enhance its ease and efficiency in this area.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

- Fee-based Ownership Transfer: Implementing a fee structure for ownership transfers to prevent an excessive number of blocks on the chain. This approach could help manage chain growth and improve overall system efficiency.
- Zero-Knowledge Proofs: Due to the need for security mechanisms, AOB currently cannot provide fully anonymous accounts throughout the process.
- User Education: The deep-rooted influence of centralized blockchains makes it challenging to shift public perception and understanding towards decentralized systems.

5.3. Long-term Vision for Development and Adoption—The long-term vision for the development and adoption of the Atomic Ownership Blockchains is to create a universally accepted, highly secure, and efficient decentralized framework that can revolutionize various sectors. AOB's unique capabilities position it as a transformative technology that can address existing limitations in traditional blockchain systems and provide significant benefits across multiple domains.

AOB can record the ownership of almost all forms of wealth, providing extremely decentralized transfer mechanisms and cryptographic-level security. This approach aims to enhance the clarity of property rights, potentially reducing disputes and improving overall system efficiency. By establishing an immutable and transparent ledger, AOB enhances trust and accountability in transactions, making it an ideal solution for managing digital assets, financial instruments, and virtual properties.

5.4. Final Thoughts on the Future of Decentralized Systems and Security—The development and adoption of AOB holds the promise of ushering in a new era for decentralized systems and security. By providing a robust and secure foundation for both monetary systems and the tracking of tangible goods, AOB has the potential to bring comprehensive commercial activities onto a platform fortified by cryptographic-level security. This innovation aims to establish clear and verifiable ownership for every unit of wealth, significantly reducing disputes and ambiguities surrounding financial rights. Ultimately, AOB's implementation could pave the way for a more transparent, secure, and equitable digital economy where trust is built into the very fabric of the system, rather than relying on intermediaries.

Author Contributions

The authors received no financial support for the research, authorship, and/or publication of this article.

Conflict of interest

The author discloses that a patent related to the technology discussed in this paper has been granted in China (Patent No. CN201980001603.4A). Additionally, a patent application for the same technology is currently pending in the United States (Application No. 17/265,105). These patents/applications do not alter the author's adherence to Ledger policies on sharing data and materials. All data and materials described in this paper will be made freely available to any

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

researcher wishing to use them for non-commercial purposes, without breaching participant confidentiality.

References

[1] Nakamoto, S. "Bitcoin: A Peer-to-Peer Electronic Cash System." (2008) https://bitcoin.org/bitcoin.pdf.

[2] Lin, Q., Li, C., Zhao, X., Chen, X. "Measuring Decentralization in Bitcoin and Ethereum Using Multiple Metrics and Granularities." *arXiv* (2021) https://arxiv.org/pdf/2101.10699.

[3] Nabilou, H. "Bitcoin Governance as a Decentralized Financial Market Infrastructure." *Stanford Journal of Blockchain Law & Policy* (30 June 2021) https://stanford-jblp.pubpub.org/pub/bitcoin-governance/release/2.

[4] Dimitri, N. "Consensus: Proof of Work, Proof of Stake and Structural Alternatives." In N. Vadgama, J. Xu, P. Tasca (Eds.) *Enabling the Internet of Value* (2022) https://link.springer.com/chapter/10.1007/978-3-030-78184-2_4.

[5] Paul, S. "What Is Bitcoin Mining? How Does Crypto Mining Work?" *G2* (3 July 2024) https://www.g2.com/articles/cryptocurrency-mining.

[6] Knight, R. "What the Bitcoin Halving Means for BTC Mining Centralization." *Cointelegraph* (1 March 2024) https://cointelegraph.com/news/bitcoin-halving-btc-mining-centralization.

[7] Laneve, C., Veschetti, A. "A Formal Analysis of Blockchain Consensus." *Università di Bologna Innovation Lab on Blockchain and New Technologies* (accessed 11 August 2025) https://site.unibo.it/blockchain-and-newtechnologies/en/papers/llncs-main.pdf.

[8] Kaur, G. "Bitcoin Nodes vs. Miners: Key Differences Explained." *Cointelegraph* (8 August 2025) https://cointelegraph.com/learn/bitcoin-nodes-vs-miners.

[9] Zaghloul, E., Li, T., Mutka, M., Ren, J." Bitcoin and Blockchain: Security and Privacy." *IEEE Internet of Things Journal* **7.10** 10288-10313 (2020) https://doi.org/10.1109/JIOT.2020.3004273.

[10] Reiff, N. "How Does a Blockchain Prevent Double-Spending of Bitcoins?" *Investopedia* (2024) https://www.investopedia.com/ask/answers/061915/how-does-block-chain-prevent-doublespending-bitcoins.asp.

[11] Ramos, S., Pianese, F., Leach, T., Oliveras, E. "A Great Disturbance in the Crypto: Understanding Cryptocurrency Returns Under Sttacks." *Blockchain: Research and Applications* **2.3** 100021 (2021) https://doi.org/10.1016/j.bcra.2021.100021.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425

[12] Karame, G., Androulaki, E., Capkun, S. "Double-Spending Fast Payments in Bitcoin." In *CCS '12: Proceedings of the 2012 ACM Conference on Computer and Communications Security* 906-917 https://doi.org/10.1145/2382196.2382292.

[13] Ciaian, P., Kancs, d., Rajcaniova, M. "The Economic Dependency of Bitcoin Security." *Applied Economics* **53.49** 5738-5755 (2021) https://doi.org/10.1080/00036846.2021.1931003.

[14] Pagnotta, E. "Decentralizing Money: Bitcoin Prices and Blockchain Security." *The Review of Financial Studies* **35.2** 866-907 (2022) https://doi.org/10.1093/rfs/hhaa149.

[15] Miller, A., Bentov, I., Kumaresan, R., McCorry, P. "Sprites: Payment Channels That Go Faster Than Lightning." *arXiv* (accessed 11 August 2025) https://arxiv.org/abs/1702.05812.

[16] Decker, C., Wattenhofer, R. "A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels." *ETH Zurich* (accessed 11 August 2015) https://tik-old.ee.ethz.ch/file/716b955c130e6c703fac336ea17b1670/duplex-micropayment-channels.pdf.

[17] Poon, J., & Dryja, T. "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments." *Lightning Network* (accessed 11 August 2025) https://lightning.network/lightning-network-paper.pdf.

[18] No Author. "Bitcoin Lightning Network." *Zion* (accessed 11 August 2025) https://docs.zion.fyi/architecture/bitcoin-lightning-network.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2025.425