Ixian Platform

A Peer-to-Peer Presence, Names and Electronic Cash System

Ixian Team
ixian@ixian.io
www.ixian.io

Version: Whitepaper Draft v0.9.9
Last Updated: 7th January 2026

1. Introduction **2**
2. User Discovery **3**
2.1 IXI Presences 3
2.2 IXI Names 4
3. The Ixian Network **5**
3.1 Peers and their Presences 6
3.1.1 Tier 1 Presences: DLT nodes and Relay nodes 7
3.1.2 Tier 2 Presences: user nodes (clients) 7
3.2 Transactions 9
4. Block Production **10**
4.1 Candidate Blocks 10
4.2 Proof of Collaborative Work (PoCW) 11
4.3 Fee policies 12
5. Steps to Run a DLT Node **13**
6. Steps to Run a Relay Node (Ixian S2) **13**
7. Scaling **14**
7.1 Managing historic blocks and SPV proofs 14
7.2 SuperBlocks 15
8. Security and Privacy **15**
8.1 Privacy of user IP addresses 15
8.2 PoCW Security 16
8.3 IXI Names Privacy 16
9. Conclusion **16**
References **17**

Abstract. Ixian is a decentralized platform for real-time device discovery, communication, and value transfer without relying on centralized servers or registries. It introduces self-authenticating presence: signed, time-bound messages that bind cryptographic addresses to current reachability information. To scale beyond what is feasible on a global ledger, Ixian uses an overlay relay network that clusters clients into sectors and stores user presence locally with redundancy, while the core DLT anchors transactions and registered names. Ixian consensus uses Proof of Collaborative Work (PoCW), combining recent proof-of-work eligibility with collaborative signing to reduce reliance on hash power alone and to improve participation in block finalization. The Ixian technology is implemented; this whitepaper is a working draft that consolidates the protocol design, parameters, and operational model.

1. Introduction

One of the foundational uses of the internet is enabling users and services to discover and communicate with each other. At the core of this communication is an IP address, which is essentially a random numeric identifier, assigned by the Internet Service Provider. However, a major challenge is that the same user typically switches between many ISPs and IP addresses on a daily basis, forcing the user to rely on a small number of gatekeepers that provide authentication, user or service discovery and data relay services.

The problems that users and developers face are inherent to any centralized communication system:

- Secure communication without trusted third parties is not practically achievable.
- Service providers can ban individual users, regions, or entire countries, restricting free communication.
- The security model is not transparent because claims made by service providers cannot be verified.
- Single points of failure lead to service outages and accessibility issues during traffic surges.
- Scaling platforms requires large up-front and ongoing infrastructure investments.

While this system functions adequately for large centralized platforms with significant resources, it greatly restricts smaller players from entering the market to provide innovation and healthy competition.

The Ixian Platform addresses these problems by implementing decentralized user discovery and direct communication between devices and users through a scalable peer-to-peer platform designed for these purposes. The platform consists of a DLT Network inspired by the original Bitcoin design [1], extended with an improved consensus algorithm (Proof-of-Collaborative Work - PoCW), capabilities for secure client discovery, and a data-relay overlay network. The system remains reliable as long as honest nodes collectively contribute the majority of recent valid PoW and signatures.

Our goal is to make digital communication easy, private, and dependable. By implementing Ixian in a fully decentralized manner, it enables true freedom of communication - free from centralized control, and gatekeeping.

2. User Discovery

At present, if a user wants to communicate with another, they must trust their application service provider, SSL Certificate provider (used by the service) and the top-level domain registrar, to give them access to the correct communication channel for the target user or service. The communication data is delivered through the service provider’s servers, which process and relay data. This requires a high level of trust in both the application service provider and top-level domain maintainer that the data is not manipulated, not intercepted, and will reach the correct recipient. Ixian takes a fundamentally different approach that is built on a new type of addressing system called IXI Presences.

2.1 IXI Presences

In our approach to client discovery, we identify each user or device by a hash of their public key of an asymmetric cryptographic algorithm like RSA, ECDSA or FN-DSA [2], which we call an IXI Address. To communicate with another user, one must know their IXI Address, which ultimately resolves to the IP address of the user or to an intermediary Relay node. The IXI Address is the primary identifier for each user and is fixed and unique. It enables users to sign their IP address with the corresponding private key and allows other users to easily verify it without the need for a trusted third party. This forms the foundation of self-authentication across the network.

IXI Presence is a combination of the user's current IP address (or relay reference), IXI Address, Device ID, and timestamp, all signed with the private key of their IXI Address. This signature allows any participant to verify authenticity without relying on third parties. Presences are time-bounded and expire after a few minutes, so they must be refreshed periodically and updated whenever the user’s reachable endpoint changes (e.g., IP change or a different Relay node is selected).

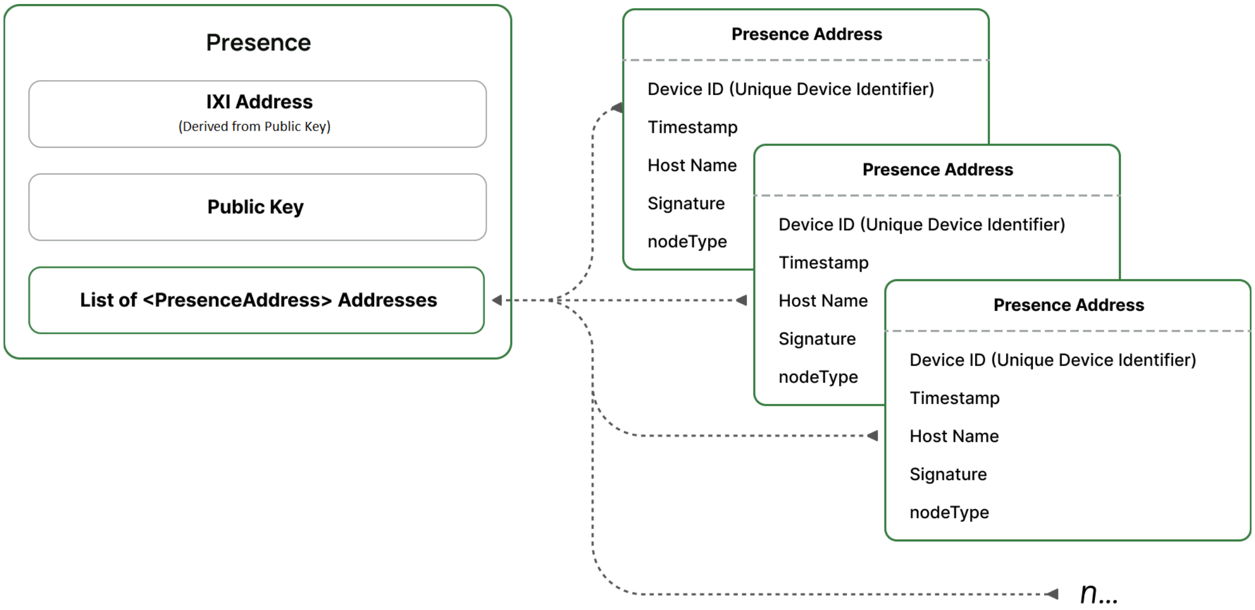
Unlike traditional systems that depend on centralized registries or certificate authorities (PKI) to validate identities, Ixian’s model is entirely self-authenticating. Each Presence is cryptographically verifiable and independent of external trust anchors, removing the need for account-based or stake-based identity systems.

The main challenge is that a list of Presences needs to be stored somewhere during the period when a user or device is online. A common solution is to introduce a trusted server that maintains Presence information and other metadata, and responds to user Presence queries. Each time a user’s IP address changes or they go online or offline, the trusted server must be updated accordingly.

When a user sends data to another user - whether it's a simple message or larger content such as video or files, this data is typically processed and, in most cases, stored by the service provider's server before being relayed to the recipient once they're online (or via push notifications).

While effective, such architectures centralize control and responsibility within a single service provider. As a result, the reliability, security, and privacy of all user communications are inherently dependent on the operational practices and trustworthiness of that provider.

We need a way for the sender to resolve the recipient's IXI Address without relying on a centralized server and without the possibility of tampering with the Presence or other important data. To accomplish this, Presences must be publicly announced, and we need a system which will temporarily store these Presences for the time that the IXI Address is online and respond to users' requests.



IXI Presence data object example

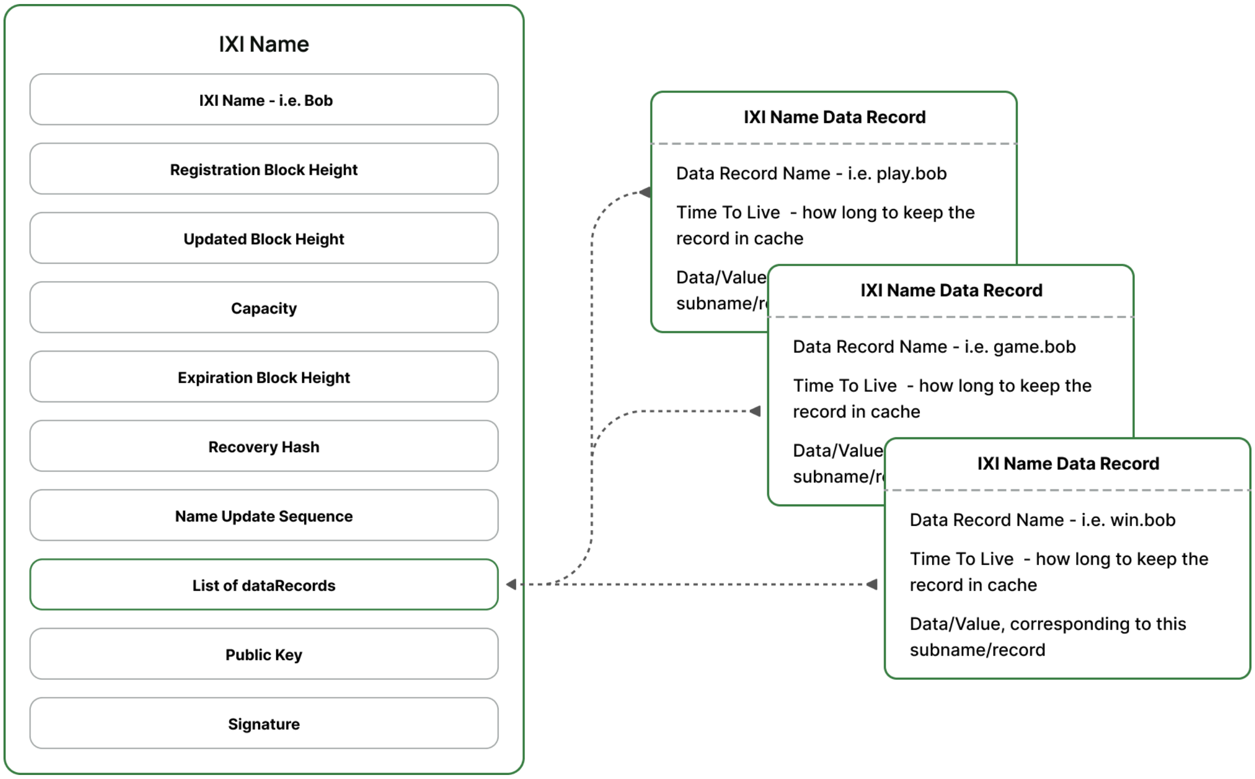
2.2 IXI Names

In order to improve the user experience and to avoid requiring to remember or transmit long cryptographic addresses, we propose a method that attaches the user's IXI Address or any kind of data to a custom user-defined name. These names must be stored by the system for longer periods of time and be available to the senders, even if the recipient is offline longer.

Traditionally Domain Name System (DNS) and accounts were used as a solution. The Domain Name System stores all domain names, and with the help of different companies where the name is registered, keeps track of who owns and can modify the name data. Every time a new name is registered, it is checked with a central server to ensure that it's not already registered.

When a name is modified, like new data is added or modified, the user must be authenticated by the domain name service provider, usually by using an account based system. Account access and operations are local to the service being used. The problem with this solution is that the reliability and integrity of all accounts, domain names, and their data rely solely on the service provider and in most cases other organisations higher in the tech stack chain.

We need a way for the users to register new accounts and names, and allow only the legitimate owners of the name to modify data associated with it. The system must be incentive-based, giving participants an economic reason to store name data, maintain integrity, and prevent spam. It must also make it costly or impractical for a single entity to register all possible names and disrupt availability for others. For our purposes only the latest valid update is the one that is relevant to the users. The only way to achieve this is to keep a record of all registered names associated with owner/management keys and other data records. To accomplish this without a trusted party, we need a system for the participants to agree on a single state of the names. The user who requests name data must be able to verify that this is the latest valid data at the time of the request.


IXI Name
IXI Name - i.e. Bob
Registration Block Height
Updated Block Height
Capacity
Expiration Block Height
Recovery Hash
Name Update Sequence
List of dataRecords
Public Key
Signature
IXI Name Data Record
Data Record Name - i.e. play.bob
Time To Live - how long to keep the record in cache
Data/Value
subname/r
IXI Name Data Record
Data Record Name - i.e. game.bob
Time To Live - how long to keep the record in cache
Data/Value
subname/r
IXI Name Data Record
Data Record Name - i.e. win.bob
Time To Live - how long to keep the record in cache
Data/Value, corresponding to this subname/record

IXI Name data object example

3. The Ixian Network

The foundation of the Ixian Platform is a decentralized peer-to-peer network that maintains a blockchain, similar to the proposal of the Bitcoin white paper. Blockchain-based systems provide a general solution to the issue of having all participants agree on a single state/history without a central authority.

Ixian DLT differs from other blockchains in its handling of:

1. Real-Time Presence list, used for client discovery and is stored in-memory.
2. Registered Names, used to provide human-friendly names tied to a cryptographic address. These are recorded in transactions and indexed by each node.
3. Three distinguished types of peers: users, Relay nodes (Ixian S2), and DLT nodes. These can be grouped into two tiers characterised by the structure of their IXI Presences.
4. No reliance on centralized indexers for any components.

The Ixian DLT acts as the public ledger anchoring all registered Names, transactions, and Tier 1 Presences, organizing them into sectors.

In this section, we will discuss how peers connect and communicate with one another to form the Ixian Platform.

3.1 Peers and their Presences

The Ixian Platform consists of three types of peers, organized into two tiers and established through signed Presences.

Tier 1 peers include DLT nodes and Relay nodes. Their Presences are created via signed announcements and attached Proof-of-Work (PoW). These Presences are globally propagated across the network and shared among DLT nodes, although perfect synchronization is not required.

Tier 2 peers consist of user nodes and overlay clients. Their Presences are also signed but do not require PoW. Unlike Tier 1, these Presences are not globally propagated and instead remain localized.

DLT nodes maintain a redundant subset of Relay node Presences sufficient for network discovery. Clients and Relay nodes may query multiple DLT nodes to assemble a complete and up-to-date view of the relevant network sector. Sectors are determined dynamically in real time, based on the number of active S2 Relay nodes.

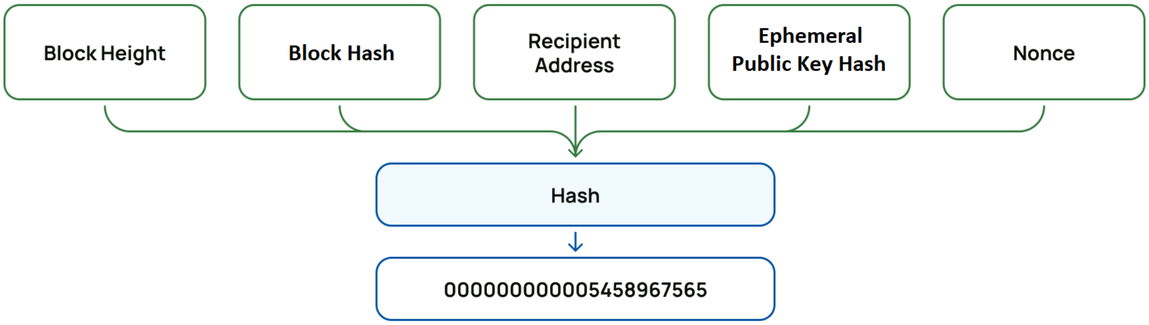
User Presences are confined to their assigned Relay clusters, where typically 7 to 10 Relay nodes redundantly track each Presence under the Starling model. This localized tracking enables efficient discovery while avoiding global state replication.

Together, this architecture provides high fault tolerance and scalability without relying on centralized indexing or full network-wide synchronization. DLT nodes form the backbone of consensus and block production, while Relay nodes cluster billions of clients and maintain localized Presence lists. User clients connect to nearby Relays, enabling communication and service access without exposing their IP addresses to other users or services.

3.1.1 Tier 1 Presences: DLT nodes and Relay nodes

DLT and Relay node Presences are established via individual PoW. They are authorised by other nodes to connect to the network if a minimum PoW is achieved. PoW is used as a costly signal to indicate willingness to participate in the network, thus preventing sybil attacks [3][4] by malicious nodes.

To produce the first PoW solution, a node syncs to the network and finds the latest valid block header information. It then produces a solution to the following hash puzzle with a specified minimum number of leading zeroes.



PoW Solution Example diagram

For DLT nodes, this PoW serves two purposes: It is used to both announce the node to the network and to allow the node to attach it to the latest candidate block, thereby contributing to consensus. Attaching it to the block simply means signing the PoW Solution data with the private key corresponding to the ephemeral public key, while including the block hash of the block which you're attaching to.

A Relay node enters the network by generating PoW, creating its Presence, and connecting to a few DLT nodes. Once created, a Relay node's Presence is propagated across DLT nodes via peer-to-peer gossip. Relay clustering follows the Starling Presence scaling model - a deterministic function based on each node's IXI Address, allowing DLT nodes to group Relay nodes into sectors for efficient discovery. Since DLT nodes are aware of all Relay node Presences, they can easily determine which node belongs to which sector.

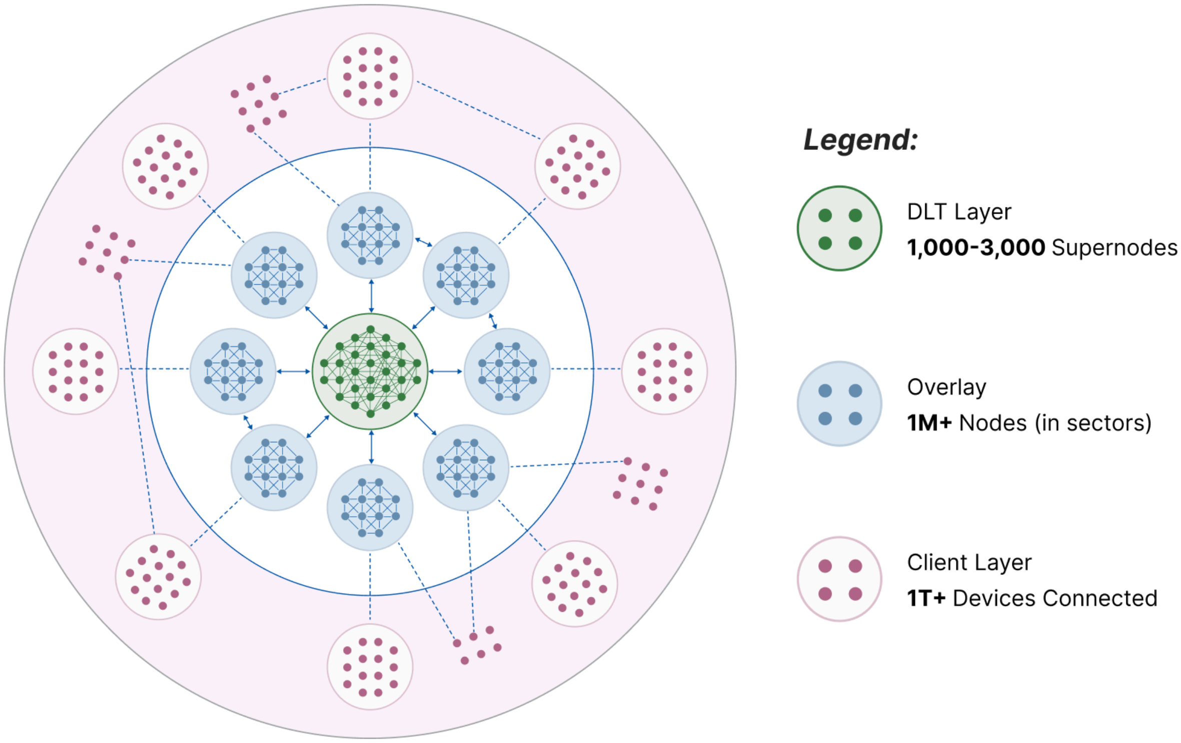
All nodes perform the same PoW algorithm; Relay and DLT nodes differ only in difficulty threshold and resulting privileges.

3.1.2 Tier 2 Presences: user nodes (clients)

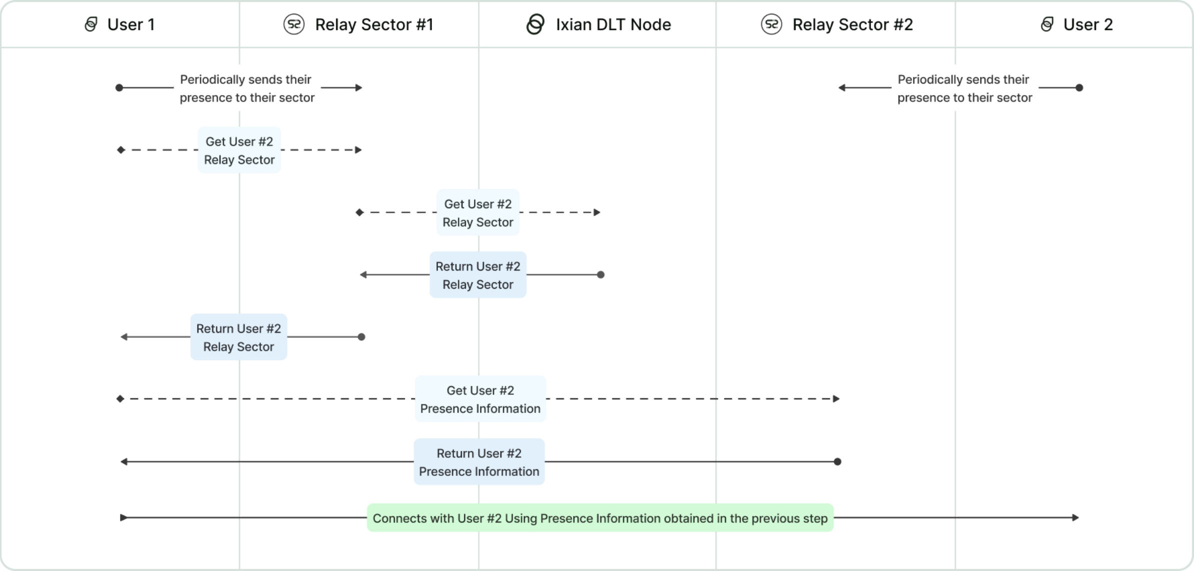
A new client connects to one or more public Relay nodes (or other designated onboarding endpoints) to determine which cluster/sector of Relay nodes they belong to, and then chooses a subset of those Relay nodes to maintain active connections to. The Presence packet of each user contains information about how to reach the user (typically an IXI or IP Address of the Relay node they are connected to).

This means that only Tier 1 Presences (DLT nodes and Relay nodes) can interact with DLT nodes, but user clients can't interact with DLT nodes directly. This segregates the network and helps prevent Sybil and DDoS attacks. Note that the DLT Network knows the Presences of the Relay nodes but not of the user clients.

The emergent network topology is organised into 3 layers. Layer 1 consists of DLT nodes, layer 2 of Relay nodes, and layer 3 of users. It can be visualised in the diagram below.



Ixian Network Topology


User 1
Relay Sector #1
Ixian DLT Node
Relay Sector #2
User 2
Periodically sends their presence to their sector
Periodically sends their presence to their sector
Get User #2 Relay Sector
Get User #2 Relay Sector
Return User #2 Relay Sector
Return User #2 Relay Sector
Get User #2 Presence Information
Return User #2 Presence Information
Connects with User #2 Using Presence Information obtained in the previous step

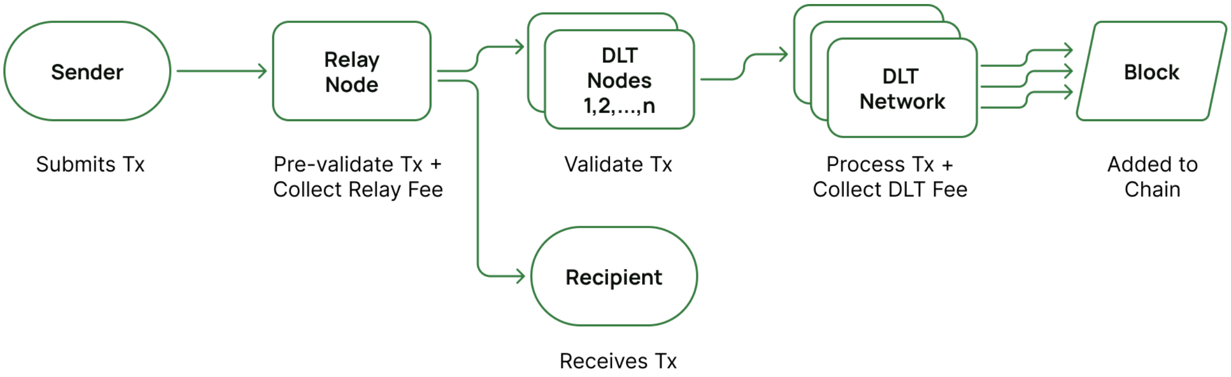
Client discovery diagram

3.2 Transactions

The purpose of a Presence is to demonstrate that a node or a user is online. On the other hand, the purpose of a transaction is to make payments, write data to the chain, and to register Names. Therefore they must be immutable and permanent.

In the Ixian DLT, transactions are simply standard blockchain transactions. Each transaction is locked to a public key, and a valid signature is required to spend the transaction. Transactions are designed to be sent directly between users peer-to-peer, and either the sender or the recipient may submit the transaction to the network. This reduces the load on the DLT Network.

A transaction enters the network by a user submitting it to a Relay node. The Relay node performs a pre-validation check before forwarding it to the DLT nodes it is connected to. The DLT nodes individually validate the transaction before propagating it to other DLT nodes. A valid transaction is then included in the next available block which indicates that it has been accepted and confirmed by the network. Both the Relay node and the DLT Network take a small fee for this process. In order to assure quick transaction propagation and to add redundancy, it is recommended that users submit transactions to multiple Relay nodes.



Transaction submitted to network

Transactions used for IXI Name registration contain a special fee, which is collected in a Name Fee Pool, a portion of which gets released to the DLT nodes with every new block for the duration name registration.

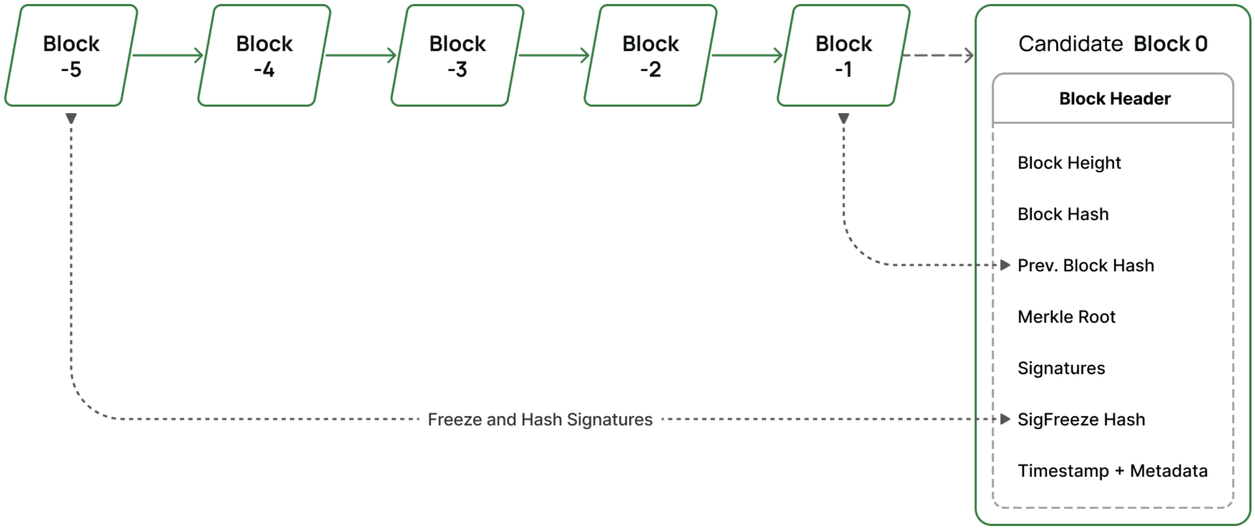
Presence announcements carry no fee, as they are integral to network operation and consensus.

4. Block Production

Block production is the process of gathering new transactions, attaching them to a candidate block, and collecting signatures from other nodes in a process called Proof-of-Collaborative-Work (PoCW). This process is undertaken by multiple nodes simultaneously and a winning block has the most PoCW.

4.1 Candidate Blocks

Constructing a new candidate block is similar to Bitcoin [1]. Transactions are gathered together into a Merkle tree [5], and a block header is constructed that contains the Merkle root, a hash of the previous block header, and other meta-data.



Block Header Example

What is different to Bitcoin, and similar to Proof-of-Stake systems [6], is that a subset of block producing nodes is chosen in advance. Nodes are deterministically selected from recent active signers based on the hash of previous blocks and signer set entropy (see PoCW below).

Each block contains a Signing Transaction, distributing the block reward which consists of a subsidy + transaction fees + a share of IXI Name fees (a portion of IXI Names fee is released to DLT nodes with every block for the duration of the name registration period). Rewards are weighted by the difficulty value of each node's PoW solution, where stronger proofs earn proportionally larger rewards. These rewards mature after a protocol-defined period (960 blocks), meaning the payout for a block occurs in a future block's Signing Transaction.

4.2 Proof of Collaborative Work (PoCW)

The PoCW mechanism governs block validation and reward distribution in the Ixian DLT. It combines elements of PoW and collective block signing, ensuring that no single node can dominate consensus through hash power alone.

To participate in block signing, a DLT node must have produced a valid PoW solution to one of the last 30 recent blocks. This time-bounded requirement prevents nodes from re-using old or exceptionally difficult PoW solutions indefinitely, ensuring that only currently active participants contribute to consensus and that signing power decays naturally over time.

A subset of nodes is deterministically selected (derived from the previous block's hash and signers who signed a block six blocks ago) to produce the next candidate blocks. To ensure redundancy and continuity, three nodes are selected at any given time. These operate in parallel to ensure continuity even if one fails to broadcast. Once a node constructs a candidate block (containing transactions, Merkle roots, and metadata), it signs it and then broadcasts it for validation and collaborative signing by other eligible nodes.

Other eligible nodes validate the candidate block. If valid, they sign it with the keypair linked to their recent PoW and broadcast their signature. Once the minimum required signatures are gathered with enough total PoW, the block is considered accepted. The protocol targets a 75% supermajority of eligible active nodes but adapts dynamically to network participation to keep the chain live. The absolute minimum number of required signatures is two. To maintain scalability, each block accepts a maximum of 1000 signatures, prioritizing the PoW solutions with highest difficulty.

Minimum PoCW difficulty is automatically adjusted every 2 weeks, and is recalculated from an average total PoCW of last blocks since the previous difficulty calculation.

Nodes may continue to add signatures to a block until it is buried under five subsequent blocks, after which the block's signatures become frozen. Freezing ensures signature finality, keeps DLT nodes synchronized with respect to the exact set of signatures attached to the block, and prevents tampering with those signatures after confirmation.

If a block cannot gather the required signatures within a defined period (e.g., two hours), the protocol automatically raises the PoCW difficulty threshold for signing eligibility, enabling block finalization with fewer participants while maintaining security guarantees.

Key Characteristics:

- Collaborative signing distributes control over block validity.
- Weighted PoW rewards maintain incentive fairness without staking [6][7].
- Targets consistent 30s block time.
- Dynamic thresholds prevent network stalls and greatly reduce risk of forks.
- 1000 signer cap preserves DLT scalability.
- The combination of PoW + signatures provides strong resistance against 51% hash attacks.

4.3 Fee policies

The structure of the transaction fee depends on whether the transaction originates from a Tier 1 or a Tier 2 node. For user transactions, we introduce a split fee model: one part for network propagation by a specific Relay node, and a second part for adding the transaction to the blockchain. This is true for all transaction types such as standard payments and Name registration transactions.

For the Relay node fee, each transaction has a specific output assigning coins to a chosen Relay node that is selected in advance by the user. (In practice, this will be chosen by the wallet or app, giving preference to a Relay node that directly connects sender and recipient).

For the DLT Network fee, this is calculated to be the difference between the input and output balance, exactly like Bitcoin. It is determined on a cost-per-byte-basis. DLT nodes accept transactions using a simple first-seen rule. This means that until a new block is produced, if a node receives a valid transaction spending a particular UTXO, then they will not accept another transaction spending the same UTXO (even if there is a higher fee), unless that transaction has been included in a block by a different node. This first-seen rule is important for enabling instant payments and node scaling.

Every IXI Name registration fee is paid to a special Name Fee Pool, which unlocks a portion of the fee on every block, for the duration of name registration.

5. Steps to Run a DLT Node

To ensure a healthy and trustworthy network, every DLT node must continuously perform the following actions. Nodes that fail to meet these obligations may be rejected or blacklisted by peers, forfeiting their PoW eligibility.

1. New Presences and transactions are broadcast to all nodes.
2. Each node re-broadcasts valid Presences and transactions to other nodes.
3. Each node syncs with other nodes to always build upon the latest valid block tip with the most PoCW.
4. Each node periodically creates PoW signatures for recent blocks. This is also announced in a Presence packet.
5. Each node listens for incoming connections and processes all valid transactions from other nodes and gathers them into a new candidate block.
6. Nodes that are eligible for proposing a new block are chosen by a deterministic random function. They are required to add their PoW signature and broadcast this to the network. If this isn't performed within a certain time period, then all nodes in the network become eligible for constructing a new block.
7. Nodes conditionally accept a block if the block data and all of its transactions are valid and not already spent. They express their acceptance by adding a PoW signature obtained in Step 4 and broadcasting it to other nodes.
8. When enough signatures and PoW are attached to a block it is accepted by all nodes. The number of signatures is limited to keep the system scalable.

6. Steps to Run a Relay Node (Ixian S2)

Relay nodes form the communication backbone between billions of user clients and the DLT Network.
Each Relay node must follow the rules below to remain visible and trusted within its sector. Nodes that violate these may be excluded by peers or ignored by DLT nodes.

1. Each node will maintain a connection to at least 6 other Relay nodes within the same sector.
2. Each node will maintain a connection to a few DLT nodes.

3. Each node periodically creates PoW signatures for recent blocks. This is also announced in a Presence packet and earns the node eligibility to be included in the Sector/Relay node's directory which DLT Network maintains.
4. Each node listens for incoming connections and processes all valid transactions, Presences and data relay requests from users.
5. Each node syncs with the DLT Network to always use the longest valid block header tip with collectively the most PoCW.
6. New valid user Presences are broadcasted to four neighbor Relay nodes within the same sector as per Starling model.
7. Each node responds to users requests for sectors, Presences (within the node's sector), block headers and transaction proofs.
8. Each node relays data to other nodes to the best of their ability.

7. Scaling

Scaling a DLT at the network level means sharding the network itself. Many real-world blockchain deployments have shown that this introduces different issues and bottlenecks when transactions are sent from one shard to the other, and has not proven to be an effective approach.

Ixian DLT nodes form the integrity core of the network. They process every transaction to guarantee a unified global ledger, accepting higher infrastructure cost in exchange for uncompromised consistency. The network does not shard and so every DLT node must process every transaction. This means that Ixian DLT nodes are typically run by specialist operators that make it the focus of their business. At the same time, the node protocol is designed to be efficient enough that there can be hundreds to thousands of node operators at any one time.

Relay nodes are introduced as an intermediary layer to connect billions of users and balance the load on DLT nodes. Relay nodes are not directly connected to all other Relay nodes, and are instead grouped together by common Presence lists under the Starling sector model. These sectors do not need to hold a common view of the global network state, and there is no a priori communication requirement between Relay node sectors. Users and applications establish their own paths of communication for each session that do not require any interaction with DLT nodes. They also manage their own application-specific states.

7.1 Managing historic blocks and SPV proofs

Standard DLT nodes will not store the entire history of the blockchain. Instead, they will keep a full copy of the most recent blocks (e.g. for the last 12 months) along with an up-to-date UTXO set, i.e., they are 'pruned'. These nodes are focused on processing new transactions, producing blocks, and establishing consensus rather than serving historic data. Instead, it is intended that full history nodes will emerge that keep a full copy of the blockchain and serve this to users.

Simplified Payment Verification (SPV) [1] is an efficient method to check that a transaction has been included in a block. It only requires end-users to store block headers and a short SPV proof per transaction, without downloading and verifying full blocks and other transactions. When a node constructs a block, it necessarily produces a Merkle tree of transactions [5] which allows SPV proofs to be easily constructed. These are calculated and stored locally for a period of time by DLT nodes and relevant Relay nodes. If an SPV proof is required by an end-user for a recent transaction, they can request this from a node once the transaction has appeared in a block. If an SPV proof is required for a historic transaction, a full history node can provide it, possibly for a fee.

Block Signatures can be pruned after they have been buried under a sufficient amount of blocks without affecting finality, as their PoCW contribution is already secured.

7.2 SuperBlocks

The Ixian DLT produces 2,880 blocks per day. If we estimate that each block is fully signed (1000 signers), it comes with a size of over 500KB per block in an unpruned state. Comparing this with Bitcoin which produces 144 blocks per day with block headers of 80 bytes, we see that the Ixian chain of block headers grows much faster and may pose storage and bandwidth challenges to end-user clients.

To address this, we introduce a special type of block called a SuperBlock. These are produced periodically (every 1000 blocks) and do not contain any transactions other than the Signing Transaction. They are connected to each other (every SuperBlock contains the hash of the previous SuperBlock) and they contain a list of all block hashes since the previous superblock along with the Merkle roots of the UTXO set and the IXI Names directory set. SuperBlocks allow lightweight clients to verify chain integrity [8] using only a minimal subset of block headers, dramatically reducing bandwidth and storage requirements, avoiding reliance on third-party RPC providers [8].

8. Security and Privacy

8.1 Privacy of user IP addresses

As described in Section 3.1.2, Presences are propagated only within each relay sector. A typical Presence contains the IXI and IP Address of the selected Relay node, not the user's own IP.

If a user wants to keep their IP address completely private, they may set up their own Relay node. This is similar to how VPNs achieve privacy, and this type of communication system can be easily replicated on the Ixian network. The system requirements for a Relay node are very light compared to a DLT node, and can easily be run on a user's system - even a Raspberry Pi.

8.2 PoCW Security

The primary identifier across Ixian is the IXI Address, represented live by a signed Presence with the corresponding private key. PoW done by DLT nodes can be attached to Presences and to blocks. To securely attach it to a block, a node must sign the block with their PoW solution data and a private key corresponding to the public key included in the PoW solution data. Nodes may rotate operational key-pairs, but authority always derives from Presence + recent valid PoW, not from long-lived accounts or stake [6][7].

An attacker cannot control the network merely by concentrating hash power.
Every new block requires collaborative signatures from a supermajority (at least 75%) of eligible active nodes, each with valid, recent PoW. Additionally, at least 50% of the signers must overlap with the set of signers from six blocks in the past. To succeed, an attacker would need to control both the current hash rate and a matching proportion of the active signing set across time - an exponentially harder requirement than a traditional 51% attack.

The protocol also discourages block withholding [9]: if no block is produced within 90 seconds, a new subset of producers is selected. If another 90 seconds pass without production, any eligible node may propose a block. This deterministic fallback ensures continuous progress and prevents gridlock even during partial outages.

8.3 IXI Names Privacy

In order to improve privacy of all IXI Names, they are hashed along with subnames/keys of their data records. Data records can be encrypted using a schema where an encryption key is derived from a combination of unhashed name and unhashed key. Meaning that only users who know the unhashed name and key, can access the record data under that key. Custom encryption schemes can also be used on top of this for records.

Threat model note:
Hashing reduces casual scraping but cannot fully prevent offline guessing for common/low-entropy names (dictionary attacks). Privacy should be treated as best-effort based on name entropy.

9. Conclusion

Ixian combines a ledger integrity layer with a scalable overlay network to support decentralized discovery and communication without central servers. The core design choice is to keep user presence out of the global ledger while preserving authenticity through signed presences and sector-based replication. PoCW is designed to reward active participation in validation and signing, not only raw hash power, while maintaining practical block times and finality properties.

This whitepaper’s remaining work is documentation-focused: tightening formal definitions, consolidating protocol parameters, and expanding the explicit security and privacy threat model.

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
[2] R. Perlner, “FIPS 206: FN-DSA (Falcon),” presentation, NIST 6th Post-Quantum Cryptography Standardization Conference, Sep. 25, 2025.
[3] A. Back, “Hashcash - A Denial of Service Counter-Measure,” Aug. 2002.
[4] J. R. Douceur, “The Sybil Attack,” in Peer-to-Peer Systems (IPTPS 2002), LNCS 2429, Springer, 2002.
[5] R. C. Merkle, “Protocols for Public Key Cryptosystems,” in Proc. IEEE Symposium on Security and Privacy, 1980.
[6] J. Brown-Cohen, A. Narayanan, C.-A. Psomas, and S. M. Weinberg, “Formal Barriers to Longest-Chain Proof-of-Stake Protocols,” in Proc. ACM EC 2019, 2019. DOI: 10.1145/3328526.3329567.
[7] S. Motepalli and H.-A. Jacobsen, “Decentralization in PoS Blockchain Consensus: Quantification and Advancement,” arXiv:2504.14351, 2025.
[8] S. Agrawal, J. Neu, E. N. Tas, and D. Zindros, “Proofs of Proof-of-Stake with Sublinear Complexity,” arXiv:2209.08673, 2022.
[9] I. Eyal and E. G. Sirer, “Majority Is Not Enough: Bitcoin Mining Is Vulnerable,” in Financial Cryptography and Data Security (FC 2014), Springer, 2014.