White Paper

JULY 31, 2017

Aion: Enabling the decentralized Internet

Matthew Spoke
matt@aion.network

Nuco Engineering Team
aion@nuco.io

Release v1.0.0, July 31, 2017

Abstract

Mainstream adoption of blockchain systems has been limited due to unsolved questions of scalability, privacy, and interoperability. In this paper, we will outline a proposed design for the Aion Network; a 3rd generation multi-tier blockchain system designed to address these challenges. Core to our hypothesis is the idea that many blockchains will be created to solve unique business challenges, within unique industries. As such, the Aion Network is designed to support custom blockchain architectures, while providing a trustless mechanism for cross-chain interoperability. At the root of this system is the world's first dedicated public enterprise blockchain, Aion-1; a state of the art blockchain that introduces a new paradigm of security, and fair, representative crypto-economic incentives.

Roadmap

This paper is intended as a technical introduction, and will be followed by additional research and design considerations. As such, over the coming months, the Aion team will publish a series of research papers that more thoroughly explain the concepts of the proposed consensus algorithm, the virtual machine and scripting language, the bridge and inter-chain transaction functionality, and the economic system underpinning the network. In addition, following this introductory paper, Aion will release its fundraising strategy and timeline for your consideration. We look forward to your feedback as we share our ideas with you.

Contents

1 INTRODUCTION

Mainstream adoption of blockchains has been limited because of scalability, privacy, and interoperability challenges. Aion is the first multi-tier blockchain network designed to address these challenges.

Core to our hypothesis is the idea that many blockchains will be created to solve unique business challenges within unique industries. As such, the Aion network is designed to support custom blockchain architectures while providing a trustless mechanism for cross-chain interoperability. At the root of the Aion network is the first dedicated, public, enterprise blockchain: Aion-1.

Aion-1 is a state-of-the-art, third-generation blockchain that introduces a new paradigm of security and fair, representative, cryptoeconomic incentives.

This paper:

- Introduces and explains the Aion network—the next generation of blockchain technology and first multi-tier blockchain network—and its necessary infrastructures and protocols.
- Details the vision and technical concepts of Aion-1, a purpose-built, public, third-generation blockchain and a component within the Aion network.
- Provides a roadmap for future implementations of Aion-1 and the Aion network.

These concepts are a work in progress and this paper is intended to establish intent and be exploratory in nature, not declarative. Join the Aion network mailing list to get alerts about more detailed white papers related to specific aspects of Aion as they become available.

2 HISTORY

The landscape of digital currencies and related blockchain technologies has changed significantly since Bitcoin was first introduced in 2008.

2.1 First-generation blockchain

Bitcoin [1] led the way in the creation of numerous alternative currency platforms as the first generation of blockchain technology. These first-generation blockchains provided a solution to conventional transaction limitations by implementing cryptographically-secure, peer-to-peer, digital transactions that are verified by a decentralized global network and recorded into an immutable public ledger. Resulting in a platform that leverages the advantages of being digital, while preserving the economics of scarcity.

2.2 Second-generation blockchain

With the second generation of blockchain, Ethereum introduced the ability to build application-specific logic upon a blockchain network [2]. This enabled new capabilities beyond transactions to incorporate state, business logic, and multi-party contracts to be stored and executed on a blockchain and written to an immutable ledger. These concepts have been incorporated into other distributed ledger technologies and have led to the distinction of building a blockchain and building upon a blockchain.

The emersion of blockchain-based applications is positive for the industry. Applications with novel use cases further demonstrate, and validate, the technology's ability to evolve beyond just a means of transferring value. However, these separate networks are becoming disparate as they are isolated and able only to transfer data off chain or transfer value through centralized exchanges. In a sense, tiny kingdoms with borders between economies and industries are being cemented. As the number of networks grow, the more disconnected and sparse the industry becomes.

Just as in the early days of the internet, disparate blockchains networks have yet to truly realize the benefits of being connected. While specialized blockchain networks will and should be developed, being able to communicate on chain to other networks offers significant benefits, particularly if privacy and scalability can be maintained. A mechanism for joining disparate networks will unlock enormous value for every participating network.

2.3 Aion: the third-generation blockchain

In the future, blockchains will federate data and value in a hub and spoke model similar to the internet. The future of mainstream blockchain adoption will be achieved by the development of a networked, federated blockchain to integrate these separate spokes. That integrated blockchain network is Aion.

Aion is a third-generation blockchain network that will enable any private or public sector organization to:

- **Federate**: Send data and value between any Aion-compliant blockchain and Ethereum.
- **Scale**: Provide fast transaction processing and increased data capacity to all Aion blockchains.
- **Spoke**: Allow the creation of customized public or private blockchains that maintain interoperability with other blockchains, but allow publishers to choose governance, consensus mechanisms, issuance, and participation.

At the root of the Aion network is a purpose-built, public, third-generation blockchain called Aion-1. Designed to connect other blockchains and manage its own robust applications, Aion-1 also provides the economic system that incentivizes interoperability in the ecosystem.

AION tokens are the fuel used to create new blockchains, monetize inter-chain bridges, and secure the overall network.

3 AION MULTI-TIER BLOCKCHAIN NETWORK

The Aion multi-tier blockchain network is like a computer network, providing a protocol and standard for dissimilar systems to communicate. However, in addition to information, the Aion network will pass logic and value among participating blockchains to create a contiguous value chain where every transaction occurs on-chain, with logic and value passing among chains as freely as liquid assets.

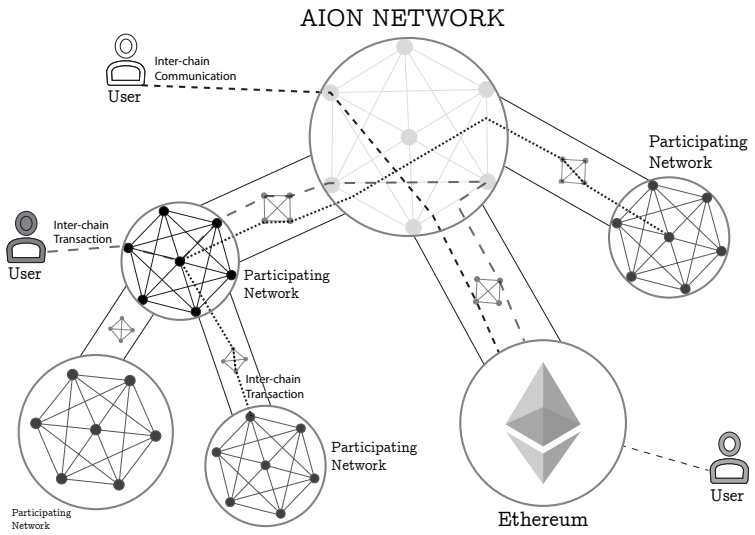

Figure 1: Example of a simple Multi-Tier Blockchain Network network, consisting of all the major actors

These infrastructures, protocols, and concepts will work together to guarantee transmission from an origin to its destination through interchain communication. The value of these technologies is that they enable one blockchain to transact with another blockchain, as well as one blockchain to transact with every connected blockchain.

3.1 Connecting Networks

Connecting networks are networks that facilitate interchain communication and interchain transactions between multiple private or public blockchain networks. Connecting networks are defined by requirements that specify their role within the context of the Aion

network. Connecting networks and interchain transactions provide a universal interface that enables blockchain developers and users to route messages from one network to another. Specifically, a connecting network should provide the following core functionalities:

- Route messages between different blockchain networks through a common bridging protocol that involves translation and propagation of the message, which must be considered final.
- Provide decentralized accountability.
- Provide a bridging protocol.

Aion network protocols specify standards for the external components. While the actual functionality and internal components of each connecting network might vary by vendor and intended purpose, these core functionalities should be implemented.

Point-to-point connections such as inter-blockchain relays or purpose-specific networks such as BTC Relay exist as central hubs. Such protocols, while simple and efficient, often result in complicated state channels that can give rise to contentious situations and are often at the mercy of one or a select group of individuals that run the relaying networks.

A connecting network instead uses bridges and a trust-free blockchain network to validate and ensure the correctness of flowing transactions. By introducing a third party that routes messages from point A to point B, the networks themselves do not have to manage difficult or unclear situations.

3.2 Interchain transaction

An interchain transaction is a trust-free message between blockchain networks, a critical infrastructure component powering interchain communication. Interchain transactions allow any connected blockchain networks to exchange information, like computers on the internet.

Interchain transactions are initially created on a source blockchain and then processed and forwarded by bridges and connecting networks before finally reaching the target blockchain. As stated previously, the creator of an interchain transaction must pay a transaction fee for the communication cost using AION tokens, thereby incentivizing all the participants in each junction of the route.

Interchain transactions are designed to be somewhat analogous to packets by specifying the hops they should perform from the source to target network, which potentially means passing through numerous connecting networks.

3.2.1 Format

Ideally, the interchain transaction format would include three parts:

- **Payload data** that is specific to the creator and is typically regular transaction data, but potentially could be extended to arbitrary data, at the discretion of the creator and the source network.
- **Metadata** about the interchain transaction that contains routing information and fees.
- **Merkle proof** that is only used when the sender wants to bypass the bridge.

The bridge and connecting network validators shall not interpret the data, but do check the integrity of the transaction as a whole. Privacy-sensitive information applications could choose to encrypt the data if necessary.

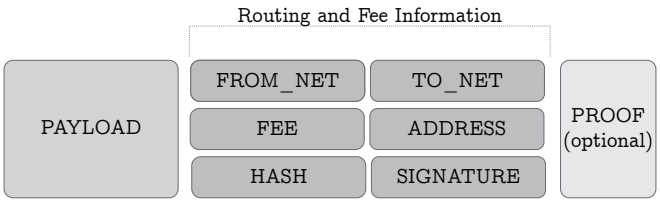

Figure 2: Visual depiction of an inter-chain transaction

3.2.2 Routing

The routing of interchain transactions is a multi-phase process. In each phase, the validators verify the transaction and reach consensus on whether the transaction should be forwarded or rejected. If a transaction gets rejected at any point, any state change as a result of the interchain transaction will be reverted, at least in the connecting network.

The routing path can be divided into two subpaths: the forward path and backward path. In the forward path, an interchain transaction flows from the source chain all the way to the target chain. In the backward path, a confirmation of the interchain transaction is passed back.

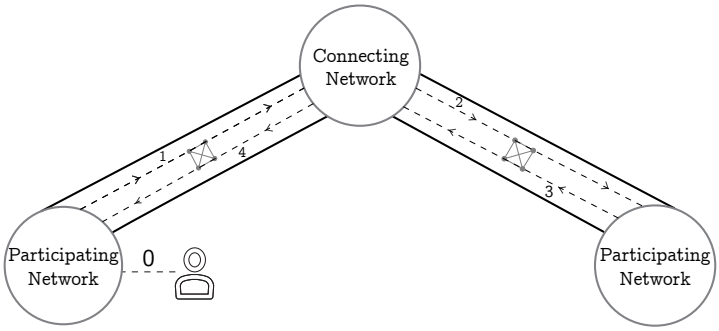

Figure 3: Depiction of an ICT lifecycle, beginning with emission from chain A, and ending with confirmation

If a bridge refuses to broadcast an interchain transaction for any reason, the sender may choose to pass the interchain transaction, including proof, directly to the connecting network. The connecting network will validate the interchain transaction based on its knowledge of a merkle hash chain of the participating network and broadcast it if valid.

The design of the interchain transaction is still under consideration and a detailed paper on the workings of interchain transactions will be published as the project progresses.

3.2.3 State

Interchain transaction state is introduced to represent the different stages/status of a transaction from the perspective of the connecting network.

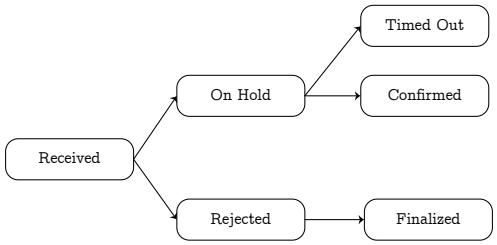

Figure 4: Flow chart of the possible states that can occur within the lifetime of an ICT

- When an interchain transaction is observed in the participating network by the bridge validators for the first time, the state changes to received.
- If over two thirds of the bridge validators vote yes for the interchain transaction, the connecting network will change the state of the interchain transaction to on hold, which will trigger an event where a corresponding connecting network token will be locked until the transaction is processed.
- If less than two thirds of the bridge validators vote yes for the interchain transaction, the state changes to rejected.

- The on-hold transaction will be forwarded by bridge validators that connect the connecting network and the next blockchain on the route.
- Once a confirmation is received from the target blockchain, the state changes to confirmed.
- If no confirmation is received, the state changes to timed out.
- For confirmed interchain transactions, the state changes to finalized and all locked fees are distributed to the connecting network and bridge validators.

3.3 Bridges

A bridge is a communication protocol that facilitates communication between the participating network and the connecting network. A bridge is composed of its own distinct network of validators that assures translation of protocols and accountability between networks.

Bridges are directional; the source blockchain is the chain where transactions are emitted and the target blockchain is the chain where the transactions are forwarded.

A bridge has two main responsibilities:

- Signing and broadcasting an interchain transaction only if they have been sealed in the source blockchain and an interchain transaction forwarding fee has been paid.
- Informing the connecting network of the merkle hash updates of the participating network.

Bridge validators will use a lightweight BFT-based algorithm to reach consensus. Transactions get approved only after receiving over two thirds of the total votes (weighted).

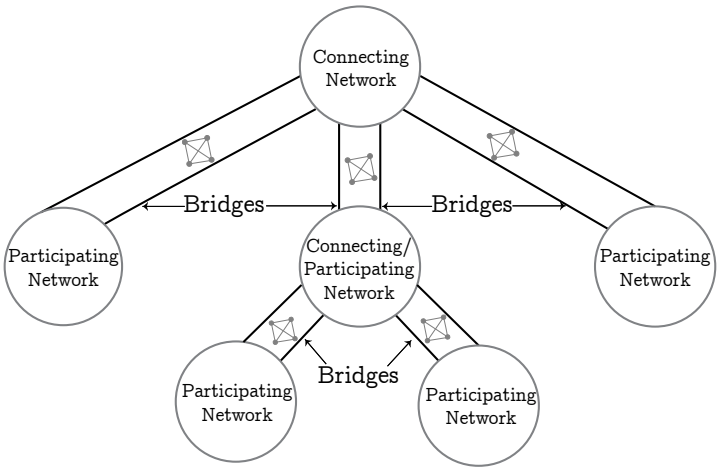

Figure 5: High level overview of bridge to connecting network relationship

3.3.1 Registration

Connecting networks are responsible for registering their directly-connected bridges. For each bridge, a dedicated table of validators will be maintained on the blockchain, sorted by stake. Anyone can join a publicly-available bridge by pledging stake towards it. Specifically, there is a contract or protocol whose purpose is to maintain a global bridge registration, which is updated dynamically as nodes join or leave bridging networks.

For a bridge to be considered valid, a minimum total stake is required. Only the topvalidators will be allowed to participate in bridge consensus.

3.3.2 Competition

Multiple bridges may be generated when multiple groups of validators register for the same blockchain network using different identifiers. From the perspective of the connecting network, these bridges are distinct, although they propagate and receive messages towards the same network.

It is then the user's responsibility to determine which bridge to use by specifying the target network identification. Here, the intent is to drive an open market by incentivizing different bridging networks to compete in terms of stability, reputation, and pricing, with the goal of an optimal fee value driven by market demands.

3.3.3 Bridge consensus

Bridge validators reach consensus by following a lightweight, BFT-based protocol where transactions are processed by one round instead of multiple rounds. Each validator evaluates a transaction based on their view of the previous blockchain. An interchain transaction is deemed valid if two thirds or more of validators voted yes, at which point the next blockchain considers the transaction valid.

Beginning at the start state, a bridge validator is required to wait until an interchain transaction is received and then verifies the validity of signature and transaction fees. Based on the validity of the transaction, it is either dropped (not signed) by the validator, or signed and propagated to the connecting or target network.

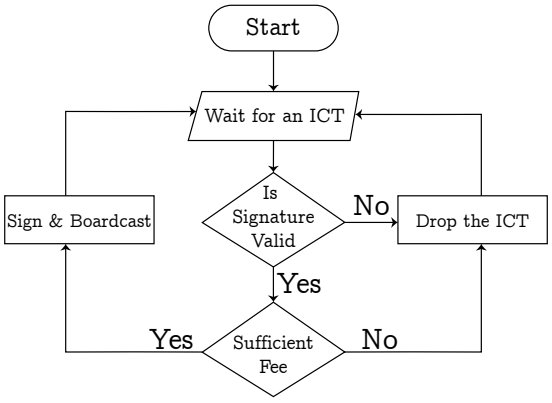

Figure 6: Algorithmic flow chart of bridge validator behaviour

3.3.4 Fee distribution

Bridge validators are rewarded from interchain transaction fees and potentially a portion of block rewards. The fee distribution aim is a fair distribution policy.

Internally, all fees that go to a bridge are distributed to the bridge validators. This could be done either proportionally to the stake each validator has put on the bridge or equally regardless of staking.

Externally, bridges share interchain transaction fees with other bridges on the routing path and with the connecting network validators. There are two possible distribution models for external fees:

- The sender of an interchain transaction specifies how the fees are distributed between bridges and the connecting network. The advantage of this approach is that users have the option to optimize the fees based on the bridge load and minimum rates. The drawback is that users need a basic understanding of the routing path and fee requirements of each bridge before sending the transaction.
- The sender only specifies the total fee and the bridge and connecting network shares this fee based on agreements or hard-coded protocol. This approach has the advantage of being simpler for the user. The drawback of this approach is that changing the ratio between bridge and connecting network is slow, if not hard.

3.4 Participating Networks

One of the core concepts in the Aion network design is that it is dedicated to the federation of compatible blockchains or blockchain-related networks. These can be purpose-specific blockchains, private networks, or consortium blockchains representing collections of entities. Regardless of the context, the interconnectedness and the ability to interoperate in an efficient, secure, and transparent manner increases the value of each network individually and also provides stability to the blockchain ecosystem in general.

A participating network is any network that has successfully implemented requirements to integrate with the connecting network. Participating networks should be blockchains, but are not necessarily limited to such. Some useful participants could be oracles, cryptlets [3], or database clusters in need of verifiable information. The only limitation is the flexibility of the participating network to integrate with the connecting network. Once integrated with the Aion network, participating networks gain access to the communication protocol (interchain transaction) specified earlier, enabling numerous possible use cases.

Participating networks have the full flexibility to customize different modules of their blockchain infrastructure including the consensus algorithm, hashing algorithm, virtual machine (VM), and scripting languages.

3.4.1 AION Compliant Blockchains

Aion-compliant blockchains refer to the participating blockchains that comply with the Aion protocol and on which bridges can be established easily to forward interchain transactions through Aion-1.

To be Aion-compliant, a blockchain must meet certain requirements including:

- Be decentralized in some fashion and support procedures commonly found in blockchains such as atomic broadcast and transactions. The exact implementation is left to the discretion of the bridging protocol and the network itself.
- Be able to recognize interchain transactions as distinct from regular transactions.
- Be aware of the consensus protocol used by the bridge and store a transaction deemed valid.
- Implement locktime or a similar feature that allows tokens to be held by the network for a period of time.

Blockchain vendors will be able to adapt their offerings to be Aion-compliant. The Nuco blockchain infrastructure will be among the first Aion-compliant networks.

More specific details on the requirements will be published as the project progresses.

3.4.2 Existing network compatibility

Unlike Aion-compliant blockchains, existing blockchains are not designed to be interoperable. To enable interchain transaction routing between the Aion network and existing blockchains, additional assumptions and/or compromises are required. In this section, we discuss the possibility of connecting the Ethereum blockchain to the Aion network.

3.4.2.1 AION to Ethereum

As part of the bridge protocol, a lightweight BFT-based consensus algorithm is used by the bridge validators. In Aion connecting networks, these BFT votes are natively aggregated and processed by the blockchain validators. The Ethereum blockchain does not have this built-in functionality, so it needs an interchain transaction contract.

In this model, the interchain transaction contract will synchronize the public keys of bridge validators periodically, depending on the Aion network specification. When an interchain transaction is requested, the bridge validators sign for it with their private key and send the signature to the interchain transaction contract. The interchain transaction contract will collect all the votes (signatures) and provide a provable record of the event that contains the interchain transaction data and voting information. If at least two thirds of votes have been received, the bridge validators will use the record as evidence when confirming the interchain transaction.

Because the computation cost of multi-signature verification is high in the Ethereum blockchain (3,000 gas for a single ECDSA), a higher bridge fee is expected. To reduce this cost, a blockchain with full BFT functionality may be used in the bridge and only the outcome of voting will be stored on the Ethereum blockchain.

3.4.2.2 Ethereum to AION

Sending interchain transactions from the Ethereum blockchain to the Aion network is easier because of Ethereum's programmable transaction size. Transactions intended for other blockchains will need to incorporate routing information in the data field.

There are two possible scenarios that can occur from an Ethereum interchain transaction (interchain transaction from the Ethereum blockchain to the Aion network), depending on the receiving address. If the transaction is sent to an externally-owned account, the data field can be used without modification. If the transaction is sent to a contract account, a workaround will be needed as the data is also interpreted by the Ethereum VM. One approach for this would be appending the interchain transaction magic tag and routing information to original call data, as long as the contract logic does not rely on the `CALLDATASIZE` op code.

To ensure transaction finality, the bridge may require additional block confirmations; typically major exchanges use 120 (half an hour) for transaction confidence.

4 AION-1 BLOCKCHAIN

The Aion-1 blockchain is the genesis implementation of the connecting network. It is designed to be a fair, distributed, open blockchain architecture that is capable of fulfilling the requirements specified in the multi-tier blockchain network architecture. As an open blockchain, Aion-1 was designed with the following goals:

- Connecting blockchains and external services (e.g., oracles and databases) together through the contiguous network and providing accountable communication maintained through a decentralized network.
- Providing the necessary infrastructure to develop high-performance, decentralized, inter-blockchain applications.
- Creating a maintainable network through a robust and sustainable economic model.

Users will be able to deploy adjacent participating networks suitable for their own use cases and communicate with other networks through an accountable routing architecture. Users ranging from large enterprises hosting consortium networks to community-oriented open networks are all welcome to participate. In the future, decentralized applications could sit on top of the connecting network with logic driven by integrating data from a multitude of blockchain networks.

In addition, the Aion-1 blockchain is equipped with a full-functioning economic system intended to drive the continued maintenance and integrity of the network.

4.1 High-level overview

In this proposal, Aion-1 refers to the genesis, or first, implementation of a connecting network. The Aion-1 implementation also serves as a fully-functioning blockchain architecture, comparable to state-of-the-art implementations found in today's market. We envision Aion-1 to be a standardized template that provides building blocks for future network implementations. Key components of the Aion-1 blockchain include:

- **Consensus** will be used to implement the proposed architecture of connecting two or more blockchains. Two variations of the BFT protocol will be designed to reach consensus on the bridge and the connecting network:
  - **Bridge consensus** is a lightweight variation to reach consensus quickly on the bridge.
  - **Connecting Network Consensus** is a consensus protocol focused on providing stability at scale.
- **Aion virtual machine (AVM)** is a custom-built, lightweight, performant, and stable VM that leverages key characteristics of the Java Virtual Machine (JVM), providing concurrency and robustness within a blockchain-specific context. The AVM is responsible for running applications on top of Aion-1. The AVM will include its own scripting language (further described below)

4.2 Consensus

We first explore the consensus algorithm featured in Aion-1 to solve the requirements introduced by the connecting network concept. The consensus algorithm selected needs to maintain consensus of the blockchain for both on-chain transactions, and interchain transactions. To fulfill these requirements in an efficient and immutable manner, Aion-1 uses a consensus algorithm based on a Byzantine Fault Tolerant

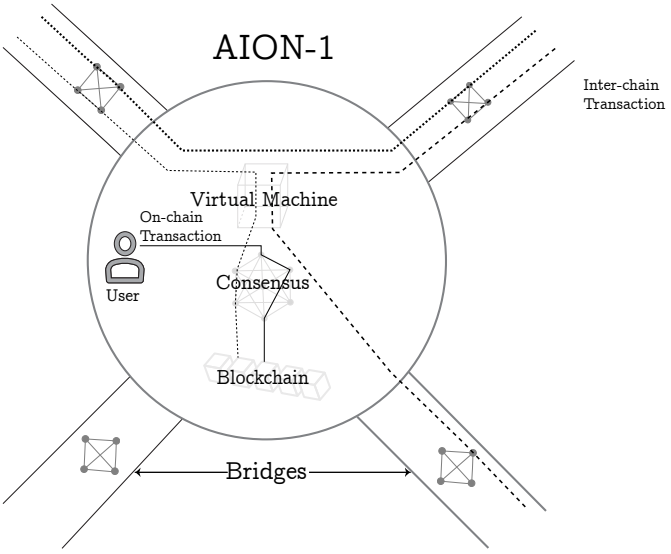

Figure 7: Birds eye view of the AION-1 architecture, depicted are the critical components of the networks: the connecting network structure, consisting of the consensus protocol and applications built on top of a virtual machine.

(BFT) algorithm combined with a hybrid protocol that aims for a fair representation of both parties in backing—partly through a token system and partly through a novel verification algorithm based on concepts used in modern neural networks called proof-of-intelligence.

To meet operational scale and enable wide participation in the network validation process, Aion-1 will employ a representative validation model similar to the delegated model explored by the BitShares team [4] and Lisk [5]. This validation model will allow Aion network participants to back validators who actively participate in the consensus process, enabling a drastic increase of participation beyond what conventional BFT algorithms would technically allow for. The specifics of the BFT-based protocol have not been finalized, but guarantee the standard properties of liveness and safety. These assumptions are complemented by the representative selection approach in that the network should be incentivized to select optimal and correct validators. Some implementations that we are investigating are HoneyBadger [6], Tangaroa [7], and Stellar [8], with special interest in the proposal behaviour within HoneyBadger and the election protocols within Stellar and Tangaroa protocols.

The conceptual design behind a representative network validation scheme is similar to that of a representative democracy in which candidates register themselves and are elected based on the votes they receive from their constituents. However, in this system, validators must be supported by backers and each backer receives a share of the reward. The rationale behind such a design is a belief in the self-governance of the network where collective action of the network directly impacts the security of the network through proper voting.

To summarize, the consensus protocol being proposed is that every node in the network can submit themselves as a candidate and pledge backing towards a candidate. At the start of every term, the highest-backed set of candidates are selected to be the validators for this term. These validators contribute to the block generation process through a BFT-based protocol and are granted a distribution of the block rewards for doing so. This is continued until the term ends and the next term begins, which restarts the process.

4.2.1 Definitions

To provide context for the various aspects of representative consensus, refer to the following set of definitions that are used consistently throughout the rest of the paper:

- **Nomination** is the process by which a node can register to become a validator to participate in representative consensus on Aion-1. Nomination must be completed before any other users in the network are able to pledge backing towards them.
- **Ranking** is used to determine the nominated validators with the highest backing. This ranked list becomes the active set, which means that the valiator node can contribute a vote towards the consensus process.
- **Active set** is the tiered list of active validators. The number of validators in the active set.

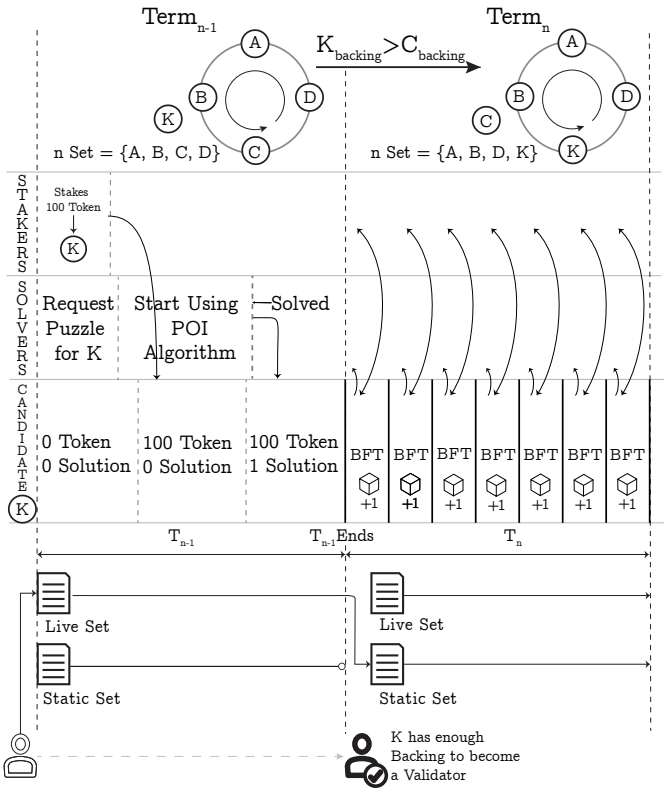

Figure 8: Active/Static Set & Staking Lifecycle, depicts the process of a candidate validator joining consensus

- **Backup sets** compose the candidate validators that are active, but are not on the active set. The backup set are the next highest backed validators. In the event of malicious behaviour or inactivity, the network looks towards this set for replacement validators.
- **Backers** refer to nodes that support validators. There will be more backers than validators on the network and their participation directly impacts the ranking of validators on the active set. Additionally, backers are proportionally rewarded based on the rewards of their validators. Backers consists of two distinct groups: stakers and solvers.
- **Stakers** are users who pledge tokens towards validators as their backing and are a subset of backers.
- **Stakes** are locked amounts of tokens held by the network until a predefined time when they are released back to the staker.
- **Solvers** are users who employ proof of intelligence to solve a cryptographic puzzle given by the network. The proof of intelligence is then converted to backing. Solvers are a subset of backers.
- **Terms** are a defined duration of time that a static set is used by the network for purposes of BFT-based consensus. In each term, a static set of validators validates new blocks. At the end of every term, the active set is frozen to generate a new static set based on changes in stake.

4.2.2 Validator Nomination Process

Any node can self-nominate and register to become a validator, but they require sufficient backing to actively validate on Aion-1. The network maintains and refreshes between terms a network-wide repository of candidate validators, the nomination contract.

Validators become active through a continuous backing process by the network. The members of the active set are always the highest-backed candidates. To facilitate this continuous backing process, there are two copies of the nomination contract at any point in time. The live set is updated as network users back or withdraw their backing from the candidates and the static set exists only for the duration of the term. The consensus protocol derives its active set from the static set. At the end of every term, the static set is overwritten with the live set for the duration of the next term.

The validators will be able to set how their backers are compensated. So, the validator proposes the terms of its backing and, if those terms are agreeable, backers will commit resources to that validator. This creates a balance of influence as the ranking (and subsequent

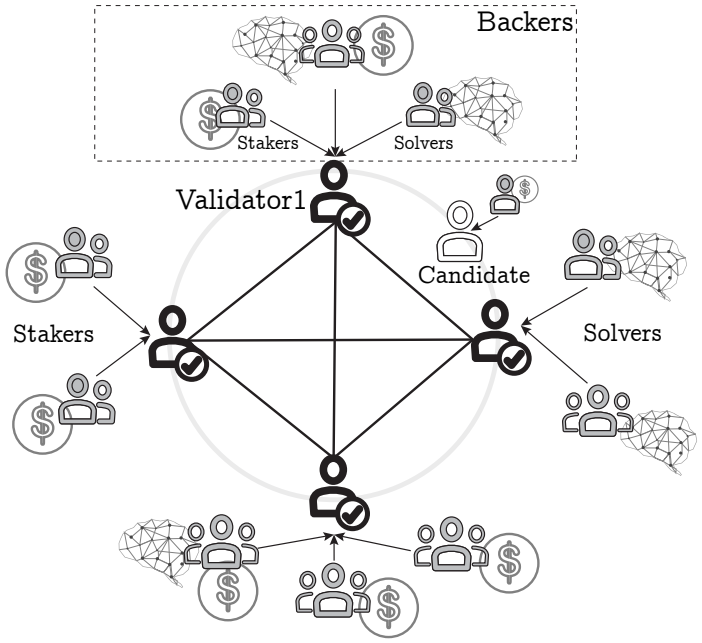

Figure 9: Structural Representation of Representative Voting - the diagram depicts the active set, with each validator backed by a number of users either through staking or solving. Also depicted is a candidate validator - a validator that does not have enough backing to join the active set.

reward) for an active validator is based on the amount of backing it has compared to other validators.

There is also a reputation mechanism for validator nodes. Candidate validators can offer their reputation as a way to attract initial backing. The exact features of reputation are being considered, but they will be quantitative, measurable, and well-defined. The intention is to encourage all participants to consider becoming candidate validators, in addition to using the Aion-1 network for its inherent utility.

4.2.3 Validator-backer rewards distribution

All users are required to present a certain amount of stake towards the network to be considered a candidate validator. However, the rewards of a validator are not necessarily proportional to their stake. Instead, the ratio of rewards going to validator and backers are proposed by the validator at the time of proposal to the nomination contract. The idea is for backers and candidate validators to arrive at some market value that both parties agree on.

4.2.4 Tiered active set

Within the active set, the validators will be organized in a tiered structure. The tiered structure is ranked based on backing, in descending order, from the highest-backed validator to the lowest. Each tier further incentivizes virtuous behaviour by providing a higher reward as compensation. The idea is to encourage decentralization by introducing a cost-benefit equilibrium point for centralized backing, thereby incentivizing backers to diversify.

The belief is this design will incentivize optimization and virtuous actions through interaction between participants. Validators will compete to receive higher backing, and backers will benefit from backing validators, but only up until they are better compensated through diversifying their backing, including backing inactive validators. One potential reward distribution scheme is as follows.

Table 1: Table depicting the reward amount (%) and voting power (%), per individual validator within that tier. Assuming $|n_{set}| = 100$

Tier	Validators	Rewards/Validator	Voting Power/Validator
1	10	2.5%	1%
2	20	1.25%	1%
3	30	0.83%	1%
4	40	0.625%	1%

Referring to the reward scheme depicted in the Table 1, rewards are divided equally among the tiers (25%), then distributed to all validators in those tiers. Validators in higher tiers have higher proportional rewards because of restricted tier sizes. A predefined reward introduces an economic model for backers to evaluate the opportunity cost of backing validators, thereby incentivizing backers to distribute their stakes amongst multiple validators (decentralization), or even nominate themselves as validators. The exact incentive model and structure will undergo rigorous simulation testing.

4.2.5 Backing

Backing refers to either staking tokens or proof-of-intelligence towards a particular validator. The network is designed to be a hybrid network that emphasizes a duality of parties to properly distribute power and monetary value evenly across the network. The belief is that a purely staking-based network (proof of stake) creates a centralization of monetary value within a select group of individuals. Therefore, an opportunity for another class of users, a class who do not possess the monetary value to participate in staking to contribute to the network, are emphasized.

The backing algorithm is broken into two distinct categories:

- Backing by staking
- Backing by solving

These two factors combine to generate backing, a conceptual intermediate value used to determine the rank of a validator, as well as the proportion of rewards given to a backer. In the following sections, each algorithm is discussed and the correlation between these three variables is investigated.

4.2.5.1 By Staking

Backing through tokens is done through staking the tokens towards a particular validator. During the term T_n, a user is able to stake towards a certain validator K within T_{n+1}. The implication of this is that the tokens are escrowed by the network until the end of T_{n+1}, at which point the tokens are returned to the user (provided no malicious actions have occurred). Before that, a user can send another message indicating that they would like the tokens to remain staked towards the same validator K.

Renewing the stake refers to the backer keeping the stakes with the validator. The concept of coinage could be a useful mechanism here where the stakes have a half life. This would encourage liquidity and maintain competition between validators. In return for staking, the backer receives a portion of the validators reward. The reward is proportional to the amount staked, as well as the current tier of the staked validator.

4.2.5.2 By Solving

Another form of backing is done through solving a cryptographic puzzle, the exact details of which are explained in the proof of intelligence section. A unique puzzle is generated per request and the puzzle must be solved through the proof-of-intelligence algorithm to generate a proof of intelligence. The proof is then submitted to the network as proof of an amount of backing for a particular validator. Solvers are also rewarded proportionally to the amount backed.

4.2.5.3 As a function of stakes and solutions

To incentivize a hybrid network, a certain distribution of stakes and proof-of-intelligence is needed. This ratio is currently arbitrarily designed to be 60/40 for stakes and proof-of-computes. The total amount of stake and solutions is accumulated per term and the backing ratio for the network is adjusted until it is consistent with the expected ratio. Therefore, having a greater proportion than the expected ratio results in lower backing per stakes/proof of intelligence, and having a lesser proportion than expected results in a greater amount of backing.

4.2.6 Incentives

The proposed system is designed to discourage bad actors or actions. However, some events may occur. In such events, the validator will be demoted tiers or removed from the active set, preventing participation in consensus and any rewards for themselves and their backers.

To discourage malicious backers, repercussions are dealt with by their method of backing. The consequences imposed by the network are designed to eliminate opportunities for reward, rather than punishment by removal or redistribution of stakes among other validators. In summary, this mechanism removes the zero-sum gain where one's loss is another's gain. Instead, it aligns motivations, and encourages positive collective actions. The belief is that, with this system in place, the individual understands the consequences of their actions and is incentivized to act in a virtuous manner and align with other virtuous actors. Bad actors will be identified by the network, and through the validator's decrease in reputation and backing, they receive feedback immediately and make corrective actions or are removed from the active set.

Table 2: An effect of the proposed punishment system, the top representative is duplicitous and is removed from consensus. All other representatives shift up, and a candidate becomes a representative

Active Set Members	Previous Tier	New Tier
$Member_1$	1	removed
$Member_2$	2	1
$Member_3$	3	2
$Member_4$	4	3
end active set		
$Member_5$	candidate	4

- **Duplicitous Actions** are punished by locking all stakes submitted by the validator for a period of time and immediately removing the validator from consensus. The backers of the validator are punished depending on the method in which they supported the validator. Stakers are punished by locking their stakes for an extended duration. Solvers are punished by the removal of the validator from consensus, thereby rendering their proof-of-intelligence solutions invalidated.
- **Inactivity** by a validator over a period of time is punished by demoting validators in the tier system. If the inactive validator is at the lowest tier level, they are immediately dropped from consensus.

Replacement validators (the next highest ranked) are available and would immediately be shifted into the consensus process until the end of the term.

4.2.7 Reputation

Representative consensus leaves the responsibility of selecting the optimal validator nodes to the network. This process is difficult to achieve unless there is a mechanism for the network to observe past behaviour of candidates and active validators. One option would be to rely on external statistics to select the optimal candidate. However, there would be incentive to manipulate this data. Therefore, a reputation system needs to be built within the network where past actions and statistics of a node are a part of the network protocol, providing trust-less data to allow users to select their appropriate candidates. Some features that could be included in node reputation include:

- **Uptime** is the amount of time a node has been active on the network and is important when determining the age (and therefore reliability) of a node.
- **Total backing** is the total or aggregate amount of backing received up until that point in time and is summed up per end of term (as stakes are locked in for that duration of time) and can indicate past performance for the node.
- **Centrality** is used within the same context as it would be within a social network, would indicate which nodes are the most well connected within the network, and could be a good indicator of reliability and performance.
- **Transaction origin** refers to the first instance of a transaction appearing on the network (that was accepted into the blockchain) and can easily be calculated with a requirement that nodes sign transactions they emit.
- **Network trust** is a global network value for a particular peer that indicates the satisfactory behaviour of the peer from the perspective of the network. This was originally designed for P2P file-sharing systems [9], and has an algorithm that can be adapted to consider parameters relevant to our use case.

These statistics are freely available to the user and an effort will be made for these statistics to be easily accessible through the internet. This creates a knowledgeable user base that is necessary to form the basis of a democratic network.

Lastly, the reputation system is a mechanism that illustrates a node's investment into the network. This investment is subject to consequences of a node's virtuous or malicious actions and is effective in assessing backing risk.

4.2.8 Proof of Intelligence

Proof of intelligence is an economic measure to deter denial of service attacks by requiring participants, solvers in Aion-1, to perform artificial intelligence (AI) computation. The intent is to motivate the creation of AI-specific or specialized hardware that could be used for machine learning and neural network training in the future.

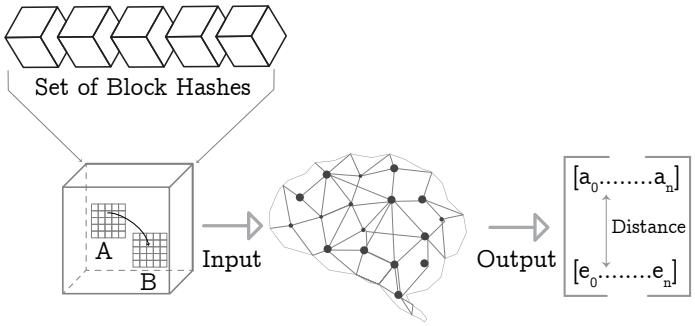

Figure 10: Overview of the Proof-of-Intelligence

4.2.8.1 Mechanism

The proof of intelligence works by requiring participants to train a predefined neural network so that it will output similar results to the proposed ground truth (e.g., the hash of current block given the hashes of previous N blocks as input). The parameters of the trained neural network will serve as a proof that computation took place and is easy to verify by inputting the parameter and confirming the results.

4.2.8.2 Validation

The validation of a proof is fast compared to the training process and is done in the following steps:

1. Load the neural network as defined by the provided parameter vector.
2. Feed the neural network with the hashes of previous N blocks.
3. Run and collect the outputs.

The validation process is being researched and developed.

4.2.8.3 Pooling

The pooling of proof of intelligence is achieved by separating the parameter space into subspaces. Similar to the pooling concept in proof of work where each miner works on a range of the nonce space, proof-of-intelligence solvers work independently on a parameter subspace. They share the proof and block reward.

4.3 Aion virtual machine (AVM)

The AVM architecture is designed to be a blockchain-specific solution with an emphasis on performance, determinism, and robustness. The AVM is a customized lightweight JVM implementation tailored towards executing chain logic (application logic) within distributed networks and hardened against scenarios that arise in such an environment.

The AVM provides the infrastructure for one of the primary functionalities of the connecting network, allowing the abstraction between the blockchain and application-specific logic and paving the way to powerful interchain applications. The rationales of this design choice and other considerations are expanded on in the upcoming sections.

4.3.1 Implementation

At the core of the AVM architecture is a lightweight, machine-friendly, blockchain-specific bytecode interpreter, after careful consideration of practical and technical objectives. Implementation requires:

- **Performance** that is close to native by using a set of machine-friendly instructions.
- **Stability** of the AVM, achieved by using an isolated VM sandbox environment and carefully measuring the computation and resource usage. New VM features will go through a formalized feature request and specification procedure, meaning that new features are well-documented and tested before moving to the production environment.
- **Determinism** for the AVM, guaranteed through a full-featured blockchain development kit as a replacement to any conventional SDK. The proposed blockchain runtime environment would be built from the ground up with determinism as the major goal. This is introduced in conjunction with the Aion VM, which only supports functionality built on top of the Aion blockchain runtime environment within the native and bytecode context.
- **Compatibility** will aim to be backwards, meaning that chain logic will always be valid and executable as the VM infrastructure evolves.
- **Tooling** from existing bytecode analysis can also be adapted to AVM bytecode. Leveraging this interoperability allows tooling that would be suitable for mission-critical code such as chain logic.

The AVM leverages of significant existing research and development efforts. Additionally, using machine-friendly bytecode makes chain-logic execution very efficient.

Customized means that the lightweight VM is configured for purposes of consumption metering (explained in the subsequent section) and isolation from the host machine (network, file I/O, unfiltered system data). The isolated environment would ensure that no meaningful information about the host machine and no unfiltered (non-oracle) communication takes place within chain logic. This is critical to ensuring the safety of the host machine and the determinism of chain logic.

Users wishing to implement the program must submit a transaction with the necessary data (defined by some binary interface). Upon receiving a message, the chain actor calls start() to initiate a boot-up sequence and accepts the data through accept(data). The logic then processes the data, modifies its state, returns a response to the network, and calls stop() to initiate the shutdown sequence.

4.3.2 Limited Consumption

One of the key issues in a VM running on top of a publicly-accessible environment is the potential for misbehaviour through malicious logic execution. In the presence of a turing-complete language, an expendable budget must be set so that the executing logic cannot run indefinitely or behave in a fashion that may damage the host machine or disrupt the consensus mechanism through faulty timing behaviours. Specifically, we define the budget mechanism as the limited consumption, which is a mechanism where an allotted value is specified by usage, space, and bandwidth consumption of the executing logic.

Effectively, the logic execution would take place in an isolated or sandboxed environment. Within our context, usage refers to the CPU usage allotted for this particular chain logic. Space refers to the memory allocation initiated by the executing logic. This prevents the execution of code that uses a large quantity of memory. Bandwidth refers to the input and output consumption of the VM. With these mechanisms in place, users executing logic would effectively rent the VM.

The protocols specify that using this mechanism requires the user to specify exactly the amount of resources given to the VM. From here, one of two things may occur (both of which produce a response):

- Successful logic execution and subsequent response
- An exception in logic execution, either through exceeding the proposed resource bounds or through the logic itself

In the event of an exception in logic execution, the AVM will inform the network of the event through an ERROR response.

4.3.3 Chain-oriented concurrency model

Blockchain networks are classically considered very serial in usage; state changes and transactions occur in serial to provide the determinism necessary for consensus. However, this creates a bottleneck in the amount of transactions that can be processed at any time period. The solution to this lies in the idea of transaction parallelism. In particular, transactions must be implemented such that context is given to the information about the state that they require. If this definition is formalized, a transaction scheduler can be implemented that allows for deterministic parallel transaction execution.

From the perspective of the AVM, support for program-level concurrency, the parallel processing of multiple chain-logic programs, is required. Towards this goal, the AVM is envisioned to be scalable, automatically clustering multiple VMs and scheduling contracts to be processed by each in a deterministic manner.

4.4 Scripting Language

The Aion scripting language is used for writing chain logic that runs on Aion-1 and potentially any connecting/participating network. The Aion language is compiled into AVM bytecode and executed by the AVM.

The Aion language provides the following features:

- Defensive programming
- Blockchain runtime environment
- Blockchain context injection
- Security

4.4.1 Specifications

The Aion language complies with a subset of the Java language specifications and is targeted for blockchain logic. To accomplish this, existing bytecode will be reviewed and possibly refactored to fit within the given context.

Additionally, the Aion language specifications include a blockchain runtime/development kit (BRE/BDK). The intent is to provide the developer with highly-optimized development libraries that implement blockchain-specific functionalities. These include but are not limited to sending transactions, emitting events, retrieving blockchain related data, and communication between chain-logic applications. This runtime environment is used to replace conventional development kits found in general compute purpose environments. In general, users of this language should expect the same syntactic structure, but a completely unique development kit.

4.4.2 Defensive programming

Defensive programming will be supported by the Aion language. According to past research [10], mistakes made by chain-logic developers result from unexpected input data, runtime exceptions, and unexpected state change after reentrance. The Aion language will provide mechanisms to decrease the possibility of these common errors. The mechanisms include:

- Aion verifies the input data before passing it to a chain logic and validates the output data after the execution.

- The scripting language introduces precondition, postcondition, and assertion to help programmers clearly organize their thoughts into a defensive pattern.
- Try/catch exceptions are fully supported by chain logic, emphasizing the handling of application state post exception rather than state rollbacks
- Bounds of array access are checked at runtime.

The addition of tooling provides the means of guiding the developer towards these mindsets, perhaps adding warnings and best practices in areas where unprotected code is found. Other features will also be considered in the future.

4.4.3 Blockchain runtime environment

The blockchain runtime environment facilitates chain-logic execution by providing a deterministic library. This library is carefully tailored to meet the determinism requirements of chain logic. Time access will be limited, using block time instead of current system time. Object allocation will be implemented in a deterministic way so that memory address-based functions will continue to work (e.g., the default `hashCode()` function). In addition, common utilities and algorithms will be carefully examined and included in the blockchain runtime environment.

4.4.4 Blockchain context injection

Dependency injection is a technique whereby one object supplies the dependencies of another object. It allows a client, such as chain logic in blockchain context, the flexibility to be configurable and hides the details of how the dependencies are provided.

Within the Aion scripting language, blockchain context and runtime information are considered dependencies. Any chain logic that requires this information can declare the requirements through the use of annotations. As Aion develops, more resources will be added to the injectable objects.

4.4.5 Security

The security of the Aion language is derived from the defensive nature of the language and the AVM where time, space, and resource usage are strictly gauged and constrained. Additionally, security should be emphasized through the tooling provided for the scripting language. For example, the logical correctness of Aion chain code can be provided by existing bytecode analysis, verification, and model checking tools. Other examples include Java Pathfinder [12], FindBugs [13], and PMD [14].

5 ROADMAP

The objectives laid out in this white paper are both ambitious and experimental. To approach this issue in a pragmatic manner, Aion-1 will be rolled out in an iterative manner, starting from existing technologies and gradually moving towards the intended objectives.

Aion-1 will be executed in three separate phases, with each phase focusing on different aspects of the technology and assembling the building blocks while progressing towards the finalization of the network (phase 3). The exact deadlines for each phase and plans are subject to change as options for each phase are investigated more thoroughly.

5.0.1 Phase 1

The focus of the first phase in the Aion release schedule is the interchain communication and bridging infrastructure. With this in mind, phase 1 functionality will include:

- Modified, high-performance EVM
- Functioning bridging and interchain communication
- Modified proof-of-work consensus algorithm

5.0.2 Phase 2

The second phase of the Aion release plan is targeted towards the migration from our modified EVM architecture towards the proposed AVM architecture. Development priorities of this phase include:

- Aion Virtual Machine
- The Aion scripting language
- EVM legacy codebase continued support

5.0.3 Phase 3

The third phase finalizes the envisioned network infrastructure, providing infrastructure for fast, efficient interchain communication and interchain applications. In addition to the interchain functionalities from Phase 1, and the VM implementation from Phase 2, this phase will introduce our representative consensus including the representative consensus algorithm.

6 CONCLUSION

The proposed solutions articulated in this paper are the result of several years of implementation and experimentation in the blockchain industry. The team behind Aion has been particularly close to several large-scale enterprise projects, where the challenges outlined previously are significantly pronounced. The Aion network is designed to overcome these challenges, and to propose a solution that will enable blockchain applications to achieve their full intended potential. In our research and development of Aion to date, we were fortunate to come across incredible findings and experiments from leading thinkers and researchers working on complementary concepts. We've done our best to capture this and give credit in the references below.

As we continue on our journey to make Aion a reality, and to connect an ever-growing fragmented ecosystem of blockchains, we look forward to engaging with you, and getting you involved.

7 CONTACT

This introductory technical paper presents the concepts of the Aion third-generation blockchain network. The team behind this paper is dedicated to realizing the dream of interconnectivity between blockchain networks, which will play a crucial role in future enterprise, government, and public digital infrastructures. Over the coming months, we will be conducting in-depth research on each of the components introduced in this paper, as well as crafting the first live iteration of this project.

Join the Aion network mailing list to get alerts about more-detailed white papers related to specific aspects of Aion-1 as they become available. You can also keep up to date with our progress through:

- Twitter
- GitHub
- LinkedIn

References

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[2] V. Buterin, “Ethereum whitepaper,” 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper.

[3] M. Gray, “Introducing project ‘bletchley’,” 2016. [Online]. Available: https://github.com/Azure/azure-blockchain-projects/blob/master/bletchley/bletchley-whitepaper.md.

[4] Bitshares, “Delegated proof of stake,” 2015. [Online]. Available: http://docs.bitshares.org/bitshares/dpos.html.

[5] O. Beddows and M. Kordek, “Lisk whitepaper,” 2016. [Online]. Available: https://github.com/slasheks/lisk-whitepaper/blob/development/LiskWhitepaper.md.

[6] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, “The honey badger of bft protocols,” 2016.

[7] C. Copeland and H. Zhong, “Tangaroa: A byzantine fault tolerant raft,” 2014.

[8] D. Mazières, “The stellar consensus protocol: A federated model for internet-level consensus,” 2015.

[9] S. Kamvar, M. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in p2p networks,” 2003.

[10] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts smarter,” 2016.

[11] N. Atzei, M. Batoletti, and C. Tiziana, “A survey of attacks on ethereum smart contracts,” 2016.

[12] NASA, “What is jpf?” 2009. [Online]. Available: https://babelfish.arc.nasa.gov/trac/jpf/wiki/intro/what_is_jpf.

[13] U. of Maryland, “FindBugs™ - find bugs in java programs,” 2015. [Online]. Available: http://findbugs.sourceforge.net/.

[14] PMD, “Welcome to pmd,” 2017. [Online]. Available: https://pmd.github.io/pmd-5.8.1/.