
EZ
Chain

EZChain Platform

Abstract. Since the mainnet launch of Ethereum in July 2015, the first significant wave (2017-2018) of crypto-currencies was crowd-fundraising via ICOs, aimed to develop smartcontract platforms (aka Layer 1 blockchains) and decentralized applications. The second wave (2020) has been led by decentralized finance applications (AMM-DEX, crypto lending & banking). The third wave (2021) observed the booming of NFTs and blockchain gaming as the first massive adoption of blockchain and crypto-currencies, then metaverse. Blockchain and crypto space has achieved remarkable growth in recent years, but many challenges and limitations still exist, for instance, scalability, cross-chain interoperability.

Many people believe that blockchain is not just a disruptive technology but a foundational technology. Blockchain is building Internet 2.0 – the Internet of value, where kinds of assets can be stored and transferred on digital platforms without any border or friction. A new era of truly digital and sharing economy has just started with blockchain, extending to every industry, for example, gaming, digital art, esport, social networking, trade finance, ecommerce, real estate, etc.
This document describes EZChain, a multi-chain and hybrid platform. Among other Layer-1 platforms, EZChain provides horizontal and vertical scalability, cross-chain interoperability and flexibility to build multiple classes of decentralized applications or specific private subnets

Disclosure: *The information presented in this paper is preliminary and subject to change at any time.*

Contents

1. A Brief Review on Blockchain Platforms

Blockchain has emerged as a hot technology trend in recent years with a brilliant perspective of applications in various industries. In particular, crypto-currency now writes its name on global financial markets. Many traditional financial institutions have invested in bitcoin, ether and top coins, e.g. Grayscale, Micro Strategy since early 2020. Undeveloped and developing nations in Asia, Africa and South America are looking for breakthrough technology to enhance and develop their poor payment & banking systems. We are going to investigate the latest and most sophisticated blockchain frameworks and architecture designs in the space to determine the suitable architecture and technologies of EZChain.

1.1 Blockchain Generations

Along the progress of Blockchain generations associated with functionality and usability, we can classify Bitcoin, Dogecoin, Stellar, IOTA and the likes as the first generation. Blockchain 1.0 (or Layer-0) provides a distributed ledger protocol for a single native asset and settlement layer only. It resolves the double-spend problem by a distributed ledger technology and a decentralized computing network accompanied with a consensus. Basically, Blockchain 1.0 technology incorporates nodes, peer to peer decentralization, wallet softwares, and mining rigs to keep the blockchain in function. However, the first generation is very limited in scalability and application.

Building on top of the concepts of Blockchain 1.0, Blockchain 2.0 is centered around the rise of Ethereum mainnet in 2015. Equipping smartcontract, Ethereum was built as a medium for other decentralized applications or dApps to be established on. This expanded the playing floor as developers could now deploy smart contracts to the Ethereum blockchain in an open sourced, permissionless method. This led to the creation of decentralized finance (“DeFi”), decentralized autonomous organizations (“DAOs,” initial coin offerings (“ICOs”), and non-fungible tokens (“NFTs”). Ethereum pioneered the concept of smartcontract and virtual machine (EVM) running on top of Blockchain settlement (i.e. Layer-0), offering Turing completeness on the decentralized computing environment, initiating the age of smartcontract platform, aka Layer-1, where various kinds of tokens and crypto-assets can be issued and exchanged. Cardano, EOS, Binance Smart Chain, Solana, Terra and many others are of Blockchain 2.0 generation, and so for Layer-2 platforms like Polygon, Optimism, Celer Network. Although Layer-2 blockchains solve scalability for Ethereum and the likes, by the nature of sovereignty, Blockchain 2.0 still has a problem of cross-chain interoperability as chains are isolated from each other.

Blockchain 3.0 can be marked by the concept of cross-chain interoperable systems, for instance, sharded chains (Ethereum 2.0, Harmony, Near Protocol), multi-chain systems (Cosmos, Polkadot, Avalanche). A huge effort and progress of Blockchain development in recent years is allowing blockchains to transact with others. For example, the Cosmos ecosystem has permitted users to build interoperable applications that allow for connection among other chains through the inter-blockchain communication ("IBC").
Decentralized hosting service, more generally, end-to-end decentralized platform concept, introduced by the Internet Computer (ICP), can be considered as the next generation of Blockchain (or Blockchain 4.0), which provides a completely scalable, interoperable solution for distributed ledger & settlement layer, virtual machine and smartcontract computing, decentralized name service, decentralized storage and decentralized server to fully host any Web3 applications.

However, what is in store for the future? Although there are no clear-cut definitions of Blockchain 3.0 and Blockchain 4.0, the main talking points appear to be the creation of solutions for services and industries outside of economics. As technology continues to develop and innovate, the potential for blockchain technology branching into other sectors appears limitless. While the development may be further than we think, blockchain technology can be integrated into supply chains, cybersecurity, voting, healthcare, and more. All of these industries have the potential to benefit from a scalable, centric ledger which would enhance traceability, increase efficiency while reducing disruptions, and improve security and transaction speed. Readers can refer to [1] and [2].

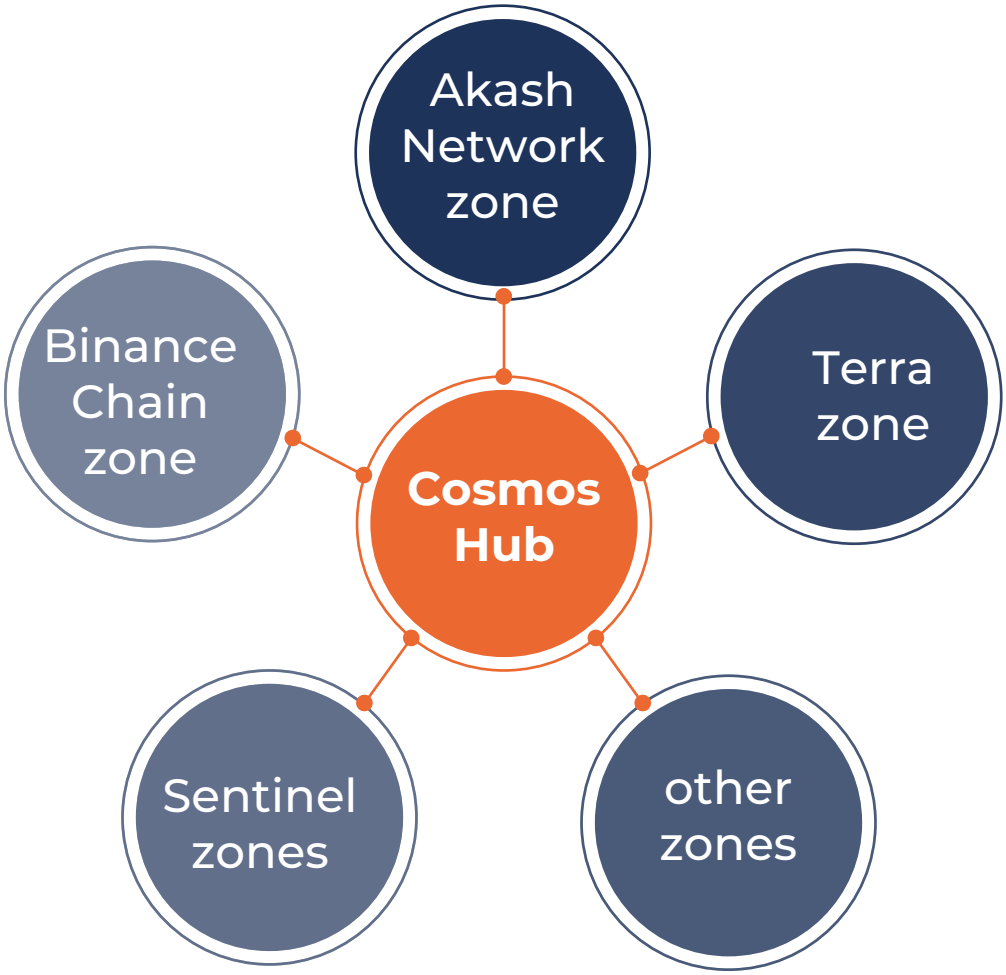
1.2 Blockchain: Big Problems and Solutions

The Blockchain Trilemma (termed by Vitalik Buterin) states a trade-off among decentralization, scalability and security. That challenges a lot of developers to introduce a complete solution. Many projects have been introduced in recent years to solve the extremely low throughput of Bitcoin and Ethereum. Most of them focus on scalability while sacrificing fully decentralization. Tron and EOS are notably projects of semi-decentralized models. Algorand claims it's the first pure proof of stake platform solving the Trilemma, but the security and sustainability of its verifiable random function and consensus protocol needs long-term public exposure to prove the assertion.

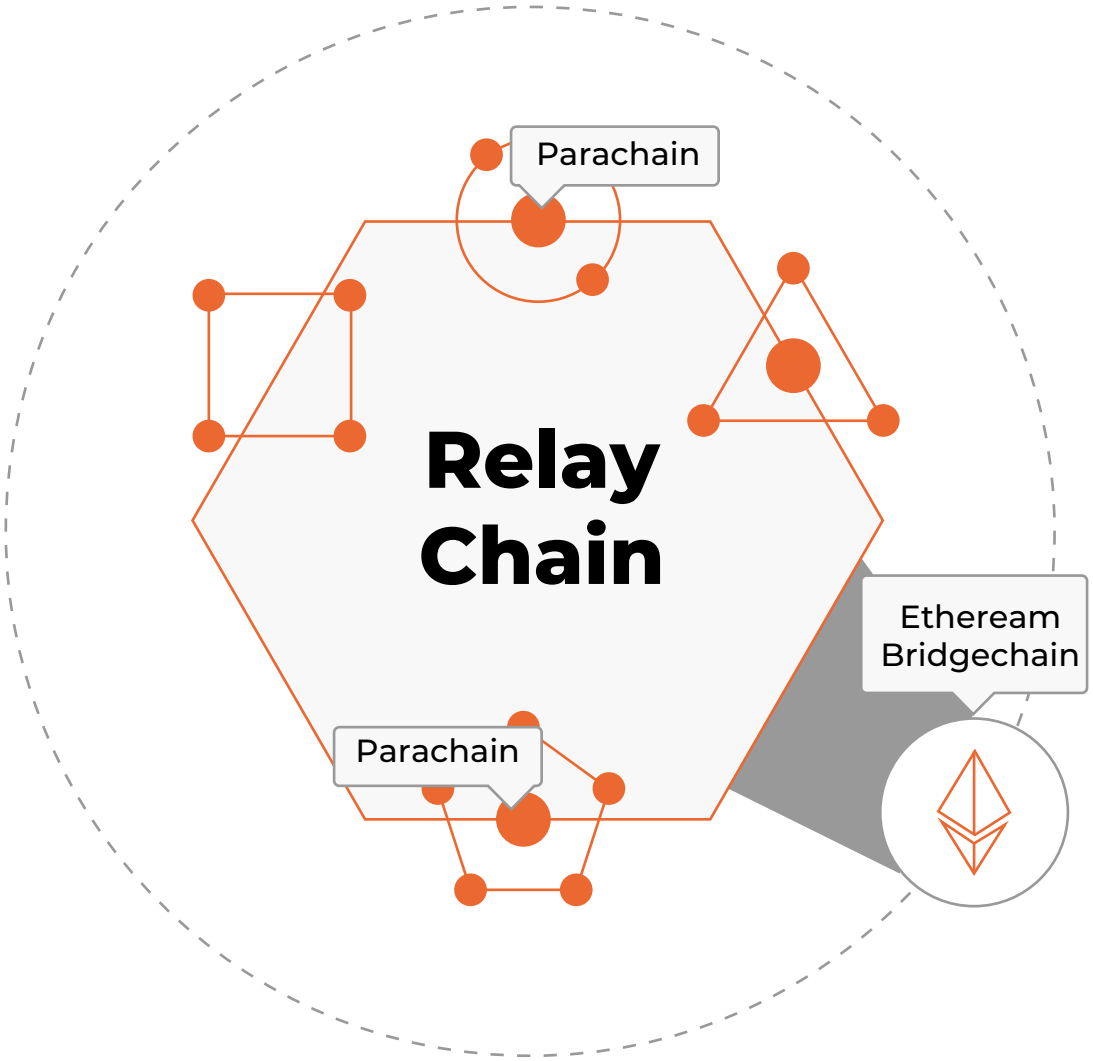
Cross-chain interoperability, on the other side, addresses one of the biggest problems of public blockchains and distributed ledger systems. Each blockchain is sovereign and isolated from the others, i.e.

value exchanging and cross-communication among different chains are impossible. This is the motivation for the birth of Cosmos and Polkadot. The projects propose to build the Internet of Blockchain, i.e. protocols allowing isolated chains to exchange assets and data with others.

Cosmos pioneered the concept of interoperable multi-chain systems, specified by zones & a blockchain hub, and the Inter-Blockchain Communication (IBC) protocol. Each zone is an independent blockchain connecting to Cosmos Hub via IBC. Cosmos doesn't allow chains to communicate directly with each other but via the common hub. This flow may result in overhead for the blockchain if there are many chains interacting simultaneously. Tendermint core (a deployable pBFT consensus), implemented by Cosmos and Binance Chain, does not process fast if the number of nodes is large. Moreover, Cosmos requires a fast finality to connect the hub, hence Bitcoin and Ethereum are out of its playground.



Polkadot approached the inter-chain problem via parachains & relay-chain, and bridge protocol. Parachains are constituent but sovereign blockchains gathering and processing transactions, while the relay chain (the heart of Polkadot) coordinates consensus and transaction delivery between chains. Bridges link parachains with external networks like Ethereum and Bitcoin. Polkadot surpasses Cosmos in the aspects of parallel processing and flexibility to build sovereign blockchains. Polkadot, in fact, provides the Substrate framework to build application-specific or custom blockchains, while parachains can benefit from interoperability and security of the entire Polkadot multi-chain system.



Heterogeneous cross-chain interoperability is a significant progress of blockchain technology. However, both Cosmos and Polkadot are invasive models which need active implementation and participation from external chains. If the majority of miners on Bitcoin and Ethereum networks don't install Cosmos IBC or the Polkadot bridge protocol, they cannot communicate back-forward with Cosmos or Polkadot.

Rebuilding a new sovereign blockchain from existing codebase is popular in the blockchain space. It is a huge benefit of the open-source world helping to boost technology development much faster than ever. Many examples and successful cases can be mentioned. Binance Smart Chain, Polygon (Matic), Tron are rebuilt from Ethereum. WAX is customized from EOS. Among others, Avalanche is chosen and customized to build EZChain. Avalanche architecture has merit between Ethereum 2.0 (Sharding network) and Polkadot (multi-chains network). On the other hand, Avalanche has flexible subnets to build customizable sidechains, and it is the most EVM-compatible platform among interoperable multi-chain designs (Polkadot, Cosmos, Avalanche). We will go to the core protocols of EZChain as follows.

2. Review on Consensus Mechanisms

Together with bitcoin, Proof of Work (PoW) has been introduced by Satoshi Nakamoto since 2008. After that, Ethereum and some other blockchains use the PoW protocol (more explicitly, Nakamoto Consensus) as well. People believe that such consensus helps the blockchains to be trustless and secure. Unfortunately, small networks are easily vulnerable by 51% attack. Huge ones like Bitcoin and Ethereum still be possibly threatened as miners are centralized into several giant mining pools. In addition, the PoW algorithm requires powerful computers to do intensive mathematical computation, hence is energy-inefficient. Another big issue of PoW is restricted scalability. Average numbers of transactions per second (TPS) on Bitcoin and Ethereum are 7 and 15, respectively. This is too slow for mass adoption. When those networks are busy, several individual transactions may take hours to days to be completed, and the transaction fee possibly goes up much higher than usual.

Ones proposes Proof of Stake (PoS) as an alternative to PoW. The idea is that instead of possessing expensive and powerful hardware for mining tasks, ones are required to hold or stake (at least) a certain number of coins. Then the network randomly chooses someone to be a block producer. This solution is obviously energy-saving but the long-range attack and the nothing at stake problems arise on a PoS system. Furthermore, it may not scale greatly if all token holders are called for verifying transactions and approving blocks. To solve those issues, several projects, for instance, EOS, Bitshares, use PoS on their blockchain consensus with a modification so-called Delegated Proof of Stake (DPoS). Some authors criticize decentralization of such blockchains (as pseudo decentralized models). October 2018, EOS faced a corrupt governance – "*mutual voting scandal*".

Vitalik Buterin, the father of Ethereum, raises a Blockchain Trilemma that says about a trade-off among “security, decentralization and scalability”. While security and decentralization are successfully achieved on several existing PoW blockchains (Bitcoin, Ethereum), scalability is still the most difficult problem. Blockchain and distributed ledger technologies are in a very beginning stage but have a fantastic potential of application. Many projects are paying huge effort to build better public blockchains. Ethereum is in transition from a pure PoW system to a PoS one to speed up TPS and to save energy.

In Peer-to-Peer networks like Bitcoin, all nodes have the equivalent role for transaction verification and block production. It is a direct democracy. Pure PoS

systems (Cardano, Algorand) are analogous models. However, high throughput (scalability) is still a challenge for such blockchains. The reason may be the difficulty in solving security problems mentioned in the previous section. Delegated Proof of Stake (DPoS) is operated as a representative democracy wherein stake holders vote for a small number of witnesses to secure and process transactions on the network. Here, without loss of general idea, we present DPoS on EOS.

A token holder must stake coins in order to vote up delegates. More coins are staked, more votes are counted. Staked coins are locked in smartcontracts during voting rounds. Top-21 voted-delegates will be the block producers (BPs) who validate transactions, create new blocks and maintain the network, i.e. be responsible for the whole network operation. A block producer (BP) gets rewards for his work. Other delegates in Top-72 receive little rewards as well to serve as standby BPs. A BP may be voted off if missing his turn or be punished due to any compromise to the network. The punishment (possibly freezing or confiscating) is executed on the number of staked coins in his smartcontract. Thus, DPoS mechanism helps solve fundamental problems (the nothing at stake, the long-rang attack and weak subjectivity) of a naive PoS system. The rationales of DPoS essentially differ from PoW.

[1] Ethereum Classic (ETC), Verge (XVG), Bitcoin SV and many chains suffered 51% attack.

[2] Unlike private/consortium chains, public blockchains allow anyone to join and leave the network without any permission from any body.

- Token holders have a partial control on the network by their votes, and a chance to earn dividends. In contrast, PoW systems reward miners only. Miners may not own any coin, so they try their best to maximize their profits. This creates a conflict between coin owners and mining bodies.
- DPoS significantly reduces the cost of network operation and maintenance while maximizing performance of blockchains, particularly scalability. Several DPoS-based blockchains have successfully scaled up to thousands of TPS, for example, EOS, WAVES, Steem, Tron, BSC. Naturally, PoS is energy-saving but still faces issues from centralized staking. Especially, DPoS model allows a small number of rich guys to control the network so it is not a fully decentralized model.

Beside significant advantages, DPoS has several issues.

- Less incentive for standby BPs and voters. In general, DPoS system rewards the vast amount for BPs, a little for standby ones and nothing for voters. Votes from token holders are important to maintain a partial decentralization of the model. With little reward, standby BPs still pay the equivalent infrastructure cost as BPs to wait for a lucky opportunity.
- More centralized approach. The opportunity for a standby one to become a BP is small. In fact, the list of BPs (Top-21 delegates) of EOS network has been almost unchanged for a long time (see "*mutual voting scandal on EOS*"). Despite voting, there is lack of diversity on the BP nodes as the whole network operation is in control of few richest bodies. This contradicts the decentralization philosophy of public blockchains.
- Common voters have a lack of knowledge to assess BPs' performance. Proof of Stake bases on the idea that token (money) owners have the right and responsibility. More money they own, the more responsibility they should pay. However, this isn't always true in reality. There are many possible cases in which BPs commit bad actions. Since all delegates are public, BPs may collude to compromise the security of the network. Beside richness, what are supplement criteria that help voters choose right delegates?
- Application developers have no control on a blockchain, although their business is running on its top. An application may bring thousands of transactions valued million dollars, but the developer doesn't have any control on the network without a significant staked amount. This conflict between top token holders and business owners is analogous to the one between miners and coin holders in PoW systems.

- BPs possibly compromise the network operating multi-million valued applications. In that case, the developers have no safeguard unless they are in the group of BPs. The fourth disadvantage can be generally considered as the conflict between value makers (workers) and money (value) holders. In a PoW system, coin holders have no control. In a DPoS system, workers don't have any right of operation (transaction verification, validation, etc), while they create the most important value for the network. Improvement approach: our idea is to integrate working and staking into a consensus mechanism, giving proportional right of control and operation to application owners and token holders.

Cardano and Algorand offer pure Proof of Stake (PoS) chains which can offer security and decentralization together with scalability. Ouroboros [3] is the first provably secure proof-of-stake protocol, and the first blockchain protocol to be based on peer-reviewed research. Ouroboros combines unique technology and mathematically-verified mechanisms which, in turn, combine behavioral psychology and economic philosophy to ensure the security and sustainability of the blockchains that depend upon it. The result is a protocol with proven security guarantees able to facilitate the propagation of global, permissionless networks with minimal energy requirements of which Cardano is the first. At the heart of Ouroboros is the concept of infinity. Global networks must be able to grow sustainably and ethically: to provide greater opportunities to the world while also preserving it. This becomes possible with Ouroboros. Ouroboros facilitates the creation and fruition of distributed, permissionless networks capable of sustainably supporting new markets.

Based on the invention of Verifiable Random Function (VRF), the Algorand blockchain [4] uses a decentralized Byzantine Agreement protocol that leverages pure proof of stake (Pure POS). This means that it can tolerate malicious users, achieving consensus without a central authority, as long as a supermajority of the stake is in non-malicious hands. This protocol is very fast and requires minimal computational power per node, giving it the ability to finalize transactions efficiently.

Unfortunately, throughput of Cardano is still low and scalability (horizontal & vertical sides) of Algorand needs much practical evidence to prove.

3. EZChain Specializations

EZChain envisions an interactive, secure and scalable ecosystem of seamless decentralized finance applications based on a new generation of blockchain. Learning from existing blockchains, it is clear that no solution is perfect, and no single chain can fit everything. EZChain team deeply studies various Blockchain platforms and see that, among others, Avalanche provides a hybrid architecture of a multi-chain system which can build a high-performance, scalable, customizable, and secure blockchain platform.

The overall goal of EZChain is to provide a unifying platform for the creation, transfer, and trade of digital assets, targeting three broad use cases:

- Building application-specific blockchains, spanning permissioned (private) and permissionless (public) deployments.
- Building and launching highly scalable and decentralized applications (Dapps).
- Building arbitrarily complex digital assets with custom rules, covenants, and riders (smart assets).

By construction, EZChain possesses the following properties:

- **Scalability.** EZChain is designed to be massively scalable, robust, and efficient. The core consensus engine is able to support a global network of potentially hundreds of millions of internet-connected, low and high-powered devices that operate seamlessly, with low latencies and very high transactions per second.
- **Security.** EZChain is designed to be robust and achieve high security. Classical consensus protocols are designed to withstand up to f attackers, and fail completely when faced with an attacker of size f + 1 or larger, and Nakamoto consensus provides no security when 51% of the miners are Byzantine. In contrast, EZChain provides a very strong guarantee of safety when the attacker is below a certain threshold, which can be parametrized by the system designer, and it provides graceful degradation when the attacker exceeds this threshold. It can uphold safety (but not liveness) guarantees even when the attacker exceeds 51%. It is the first permissionless system to provide such strong security guarantees.
- **Decentralization.** EZChain is designed to provide unprecedented decentralization. This implies a commitment to multiple client implementations and no centralized control of any kind. The ecosystem is designed to avoid divisions between classes of users with different interests. Crucially, there is no distinction between miners, developers, and users.

- **Governance and Democracy.** EZChain is a highly inclusive platform, which enables anyone to connect to its network and participate in validation and first-hand in governance. Any token holder can have a vote in selecting key financial parameters and in choosing how the system evolves.
- **Interoperability and Flexibility.** EZChain is designed to be a universal and flexible infrastructure for a multitude of blockchains/assets, where the base $EZC is used for security and as a unit of account for exchange. The system is intended to support, in a value-neutral fashion, many blockchains to be built on top. The platform is designed from the ground up to make it easy to port existing blockchains onto it, to import balances, to support multiple scripting languages and virtual machines, and to meaningfully support multiple deployment scenarios.

3.1 Consensus mechanism

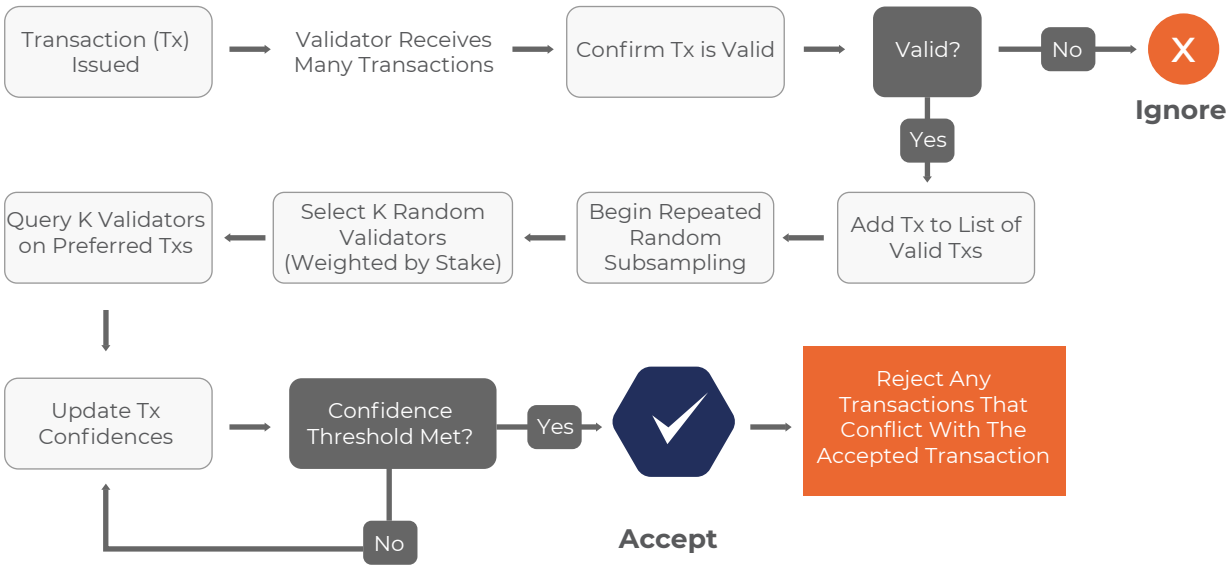
We begin with the core component which powers the EZChain platform: the consensus engine. EZChain uses the family of Snow* consensus protocols [5], combining the best properties of classical consensus protocols with the best of Nakamoto consensus. Based on a lightweight network sampling mechanism, they achieve low latency and high throughput without needing to agree on the precise membership of the system. They scale well from thousands to millions of participants with direct participation in the consensus protocol. Further, the protocols do not make use of PoW mining, and therefore avoid its exorbitant energy expenditure and subsequent leak of value in the ecosystem, yielding lightweight, green, and quiescent protocols.

3.1.1 Mechanism and Properties

The Snow* protocols operate by repeated sampling of the network. Each node polls a small, constant-sized, randomly chosen set of neighbors, and switches its proposal if a supermajority supports a different value. Samples are repeated until convergence is reached, which happens rapidly in normal operations.

We elucidate the mechanism of operation via a concrete example. First, a transaction is created by a user and sent to a validating node, which is a node participating in the consensus procedure. It is then propagated out to other nodes in the network via gossiping. What happens if that user also issues a conflicting transaction, that is, a double spending? To choose amongst the conflicting transactions and prevent the double- spend, every node randomly selects a small subset of nodes and queries which of the conflicting transactions the queried nodes think is the valid one.

If the querying node receives a supermajority response in favor of one transaction, then the node changes its own response to that transaction. Every node in the network repeats this procedure until the entire network comes to consensus on one of the conflicting transactions.



Surprisingly, while the core mechanism of operation is quite simple, these protocols lead to highly desirable system dynamics that make them suitable for large-scale deployment.

- *Permissionless,* Open to Churn, and Robust. The latest slew of blockchain projects employ classical consensus protocols and therefore require full membership knowledge. Knowing the entire set of participants is sufficiently simple in closed, permissioned systems, but becomes increasingly hard in open, decentralized networks. This limitation imposes high security risks to existing incumbents employing such protocols. In contrast, Snow* protocols maintain high safety guarantees even when there are well- quantified discrepancies between the network views of any two nodes. Validators of Snow* protocols enjoy the ability to validate without continuous full membership knowledge. They are, therefore, robust and highly suitable for public blockchains.
- *Scalable and Decentralized* A core feature of the Snow family is its ability to scale without incurring fundamental tradeoffs. Snow protocols can scale to tens of thousands or millions of nodes, without delegation to subsets of validators. These protocols enjoy the best-in-class system decentralization, allowing every node to fully validate. First-hand continuous participation has deep implications for the security of the system. In almost every proof-of-stake protocol that attempts to scale to a large participant set,

the typical mode of operation is to enable scaling by delegating validation to a subcommittee. Naturally, this implies that the security of the system is now precisely as high as the corruption cost of the subcommittee. Subcommittees are furthermore subject to cartel formation. In Snow-type protocols, such delegation is not necessary, allowing every node operator to have a first- hand say in the system, at all times. Another design, typically referred to as state sharding, attempts to provide scalability by parallelizing transaction serialization to independent networks of validators. Unfortunately, the security of the system in such a design becomes only as high as the easiest corruptible independent shard. Therefore, neither subcommittee election nor sharding are suitable scaling strategies for crypto platforms.

- *Adaptive*. Unlike other voting-based systems, Snow* protocols achieve higher performance when the adversary is small, and yet highly resilient under large attacks.
- *Asynchronously Safe*. Snow* protocols, unlike longest-chain protocols, do not require synchronicity to operate safely, and therefore prevent double-spends even in the face of network partitions. In Bitcoin, for example, if synchronicity assumption is violated, it is possible to operate independent forks of the Bitcoin network for prolonged periods of time, which would invalidate any transactions once the forks heal.
- *Low Latency*. Most blockchains today are unable to support business applications, such as trading or daily retail payments. It is simply unworkable to wait minutes, or even hours, for confirmation of transactions. Therefore, one of the most important, and yet highly overlooked, properties of consensus protocols is the time to finality. Snow* protocols reach finality typically in ≤ 1 second, which is significantly lower than both longest-chain protocols and sharded blockchains, both of which typically span finality to a matter of minutes.
- *High Throughput*. Snow* protocols, which can build a linear chain or a DAG, reach thousands of transac- tions per second (5000+ tps), while retaining full decentralization. New blockchain solutions that claim high TPS typically trade off decentralization and security and opt for more centralized and insecure consensus mechanisms. Some projects report numbers from highly controlled settings, thus misreporting true performance results. The reported numbers for $EZC are taken directly from a real, fully implemented EZChain network running on 2000 nodes on AWS, geographically distributed across the globe on low-end machines.

Higher performance results (10,000+) can be achieved through assuming higher bandwidth provisioning for each node and dedicated hardware for signature verification. Finally, we note that the aforementioned metrics are at the base-layer. Layer-2 scaling solutions immediately augment these results considerably.

3.1.2 Comparative Charts of Consensus mechanisms

Table 1 describes the differences between the three known families of consensus protocols through a set of 8 critical axes

	Nakamoto	Classical	Snow*
Robust (Suitable for Open Settings)	+	-	+
Highly Decentralized (Allows Many Validators)	+	-	+
Low Latency and Quick Finality (Fast Transaction Confirmation)	-	+	+
High Throughput (Allows Many Clients)	-	+	+
Lightweight (Low System Requirements)	-	+	+
Quiescent (Not Active When No Decisions Performed)	-	+	+
Safety Parameterizable (Beyond 51% Adversarial Presence)	-	-	+
Highly Scalable	-	-	+

Table 1. Comparative chart between the three known families of consensus protocols. EZChain, Snowman, and Frosty all belong to the Snow* family.

3.2 Platform Architecture

In this section, we provide an architectural overview of EZChain – hybrid, multi-chain platform and discuss various implementations in details. The EZChain platform cleanly separates three concerns: chains (and assets built on top), execution environments, and deployment.

3.2.1 Virtual Machines

Each blockchain is an instance of a Virtual Machine (VM.) A VM is a blueprint for a blockchain, much like a class is a blueprint for an object in an

object in an object-oriented programming language. The interface, state and behavior of a blockchain is defined by the VM that the blockchain runs. The following properties of a blockchain, and other, are defined by a VM:

- The contents of a block
- The state transition that occurs when a block is accepted
- The APIs exposed by the blockchain and their endpoints
- The data that is persisted to disk

We say that a blockchain "uses" or "runs" a given VM. When creating a blockchain, one specifies the VM it runs, as well as the genesis state of the blockchain. A new blockchain can be created using a pre-existing VM, or a developer can code a new one. There can be arbitrarily many blockchains that run the same VM. Each blockchain, even those running the same VM, is logically independent from others and maintains its own state.

3.2.3 Subnetworks

A subnetwork, or subnet, is a dynamic set of validators working together to achieve consensus on the state of a set of blockchains. Each blockchain is validated by one subnet, and a subnet can validate arbitrarily many blockchains. A validator may be a member of arbitrarily many subnets. A subnet decides who may enter it, and may require that its constituent validators have certain properties. The EZChain platform supports the creation and operation of arbitrarily many subnets. In order to create a new subnet or to join a subnet, one must pay a fee denominated in $EZC.

The subnet model offers a number of advantages:

- If a validator doesn't care about the blockchains in a given subnet, it will simply not join that subnet. This reduces network traffic, as well as the computational resources required of validators. This is in contrast to other blockchain projects, in which every validator must validate every transaction, even those they don't care about.
- Since subnets decide who may enter them, one can create private subnets. That is, each blockchain in the subnet is validated only by a set of trusted validators.
- One can create a subnet where each validator has certain properties. For example, one could create a subnet where each validator is located in a certain jurisdiction, or where each validator is bound by some real-world contract. This may be beneficial for compliance reasons.

There is one special subnet called the Default Subnet. It is validated by all validators. (That is, in order to validate any subnet, one must also validate the Default Subnet.) The Default Subnet validates a set of pre-defined blockchains, including the blockchain where $EZC lives and is traded.
At launch, EZChain hybrid platform is composed of 3 chains (Governance Chain, DAG-system and EVM-chain) described as follows.

Governance Chain
is a default subnet, running Snowman consensus protocol [5], responsible for validator registration and managing sub-samples operating regular subnets. On the other words, Governance Chain coordinates all validators, creates subnets and manages the entire platform.

EVM-Chain
is a special subnet, built-in to be EVM-compatible blockchain, running the Snowman consensus protocol. Developers can write Solidity smartcontract, issue token types of ERC20/ERC721/ERC1155 and build dapps easily as on Ethereum.

DAG-System
is a default, special subnet which is a non-linear chain (like IOTA), uses Un-spent Transaction Output (UTXO). The native token of EZChain ($EZC) is generated and lives on the DAG-system. Other assets can be issued, transferred and exchanged on the DAG-system as well. EZChain utilizes Directed Acyclic Graph (DAG) structure, UTXO model to build-in an asynchronous, secure, scalable asset exchange ledger.

The DAG-system stores all available balances in a datastore called Unspent Transaction Outputs (UTXOs). A UTXO Set is the unique list of outputs produced by transactions, addresses that can spend those outputs, and other variables such as lockout times (a timestamp after which the output can be spent) and thresholds (how many signers are required to spend the output). Readers can refer to many UTXO models like Bitcoin, Cardano, Dogecoin.

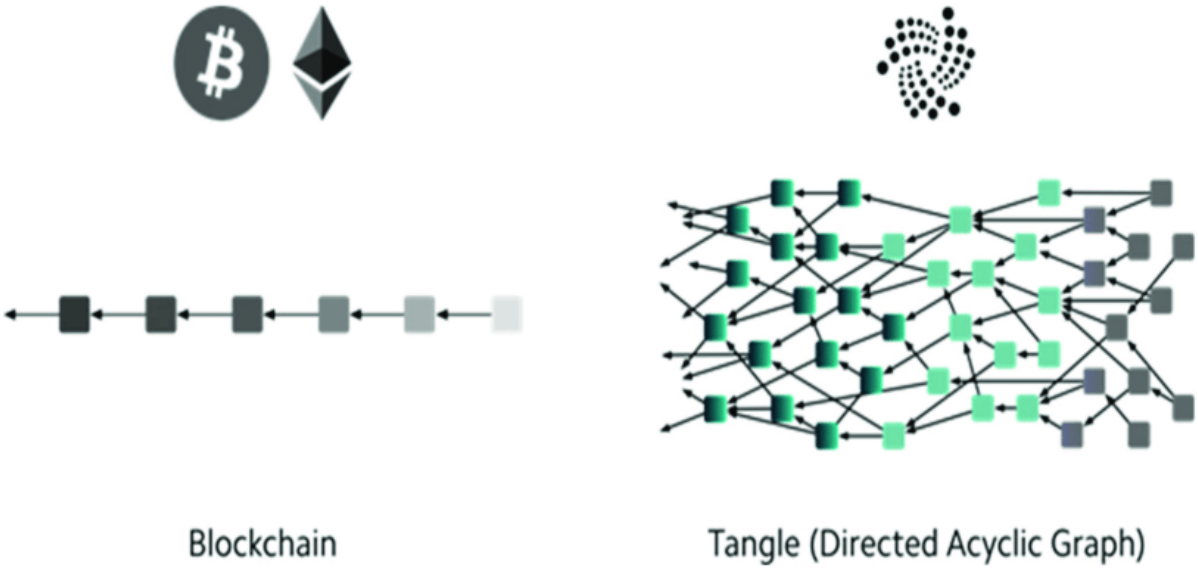
Unlike a blockchain, which consists of blocks, directed acyclic graphs have vertices and edges. Thus, crypto transactions are recorded as vertices. These transactions are then recorded on top of one another. Similarly to a blockchain, however, transactions are also submitted to the DAG via nodes. Simply put, whereas a blockchain system looks like a chain, DAG's system looks more like a graph. The DAG model is currently seen in the industry as a possible substitute for blockchains in the future due to its efficiency in data storage and processing of online transactions.

The DAG model is seen as a possible solution to the current decentralization issue in crypto. With this model, miners will not have to compete for new blocks to add to the chain. With nodes developed simultaneously, transactions can likewise be processed faster. Developers are eyeing DAG as a better, more secure solution that can improve a network's usability once it becomes more scalable.

As mentioned, a directed acyclic graph is more efficient at data storage. Its structure is tree-like, with interconnected nodes as its 'branches.' Since each node can have more than one parent root, the model allows for more transactions to be validated simultaneously. This is because users do not have to wait for transactions to complete before processing a new one. Thus, in a directed acyclic graph, each new transaction has to reference previous transactions before getting accepted into the network. This is no different from how blocks on a blockchain also reference previous blocks. The rationale behind this is that a transaction can only be successfully confirmed when it is referenced by another transaction, and so on. In a DAG, each vertex represents a transaction. There are no blocks, so mining is also not required. Transactions are built on top of one another instead of gathering them into blocks. Then, as previously mentioned, proof-of-work tasks are done whenever a node submits a transaction, to validate prior transactions and avoid spam. By principle, new transactions are built on top of older ones in a DAG-based cryptocurrency. The main difference with blockchain is that in a DAG, multiple transactions can be referenced, instead of just one at a time. Some systems have an algorithm that selects 'tips' or transactions to build on based on accumulated weight (or the number of confirmations leading up to the tip). Double-spend protection in DAGs works with nodes confirming older transactions by assessing a path tracing back to the DAG's first transaction. This confirms whether or not the sender has sufficient balance. Should a user build on an invalid path, then that transaction is at risk of being ignored. Conflicts resulting from multiple paths are resolved via a selection algorithm that favors tips that have a heavier accumulated weight.

	DAG	Linear chain
Pros	● Suited for microtransactions and high volumes of transactions ● Eliminates the need for mining equipment ● Fees may be reduced significantly ● Lower energy consumption	● Well-established and widely-used by cryptocurrencies like Bitcoin and Ethereum ● Transparent and unalterable, highly-secure ● Cost-effective for high-value transactions
Cons	● Vulnerable to attacks due to low volume of transactions ● Still in its infancy; has not yet sustained high levels of decentralization	● Demanding storage and network bandwidth requirements ● Large amounts of power consumed ● High transaction fees

IOTA, Nano are notable projects utilized DAG data structure. DAG and UTXO make EZChain a hybrid platform (consisting of a DAG ledger & linear chains, UTXO & account models). Hybrid design results in the complexity of EZChain but makes it more scalable, secure and flexible to various application purposes.



Blockchain

Tangle (Directed Acyclic Graph)

[3] Avalanche, Fantom are hybrid platforms as well

4. Smartcontract on EZChain

At launch EZChain supports standard Solidity-based smart contracts through the Ethereum virtual machine (EVM). We envision that the platform will support a richer and more powerful set of smart contract tools, including Smart contracts with off-chain execution and on-chain verification.

- art contracts with parallel execution. Any smart contracts that do not operate on the same state in any subnet in EZChain will be able to execute in parallel.
- An improved Solidity, called Solidity++. This new language will support versioning, safe mathematics and fixed-point arithmetic, an improved type system, compilation to LLVM, and just-in-time execution.

If a developer requires EVM support but wants to deploy smart contracts in a private subnet, they can spin-up a new subnet directly. This is how EZChain enables functionality-specific sharding through the subnets. Furthermore, if a developer requires interactions with the currently deployed Solidity smart-contracts, they can interact with the Ethereum subnet, which is a spoon of Ethereum. Finally, if a developer requires a different execution environment from the Ethereum virtual machine, they may choose to deploy their smart contract through a subnet that implements a different execution environment, such as DAML or WASM. Subnets can support additional features beyond VM behavior. For example, subnets can enforce performance requirements for bigger validator nodes that hold smart contracts for longer periods of time, or validators that hold contract state privately.

5. EVM++

EVM++ is a set of non-intrusive, fully backward compatible enhancements for the original Ethereum Virtual Machine popularized by Ethereum. The goal is to extends the functionality of EVM and its API to make it more accessible, efficient, and scalable while being fully backward-compatible:

- The entire toolchain is kept intact: contract binary, languages (Solidity, Viper), compilers (solc), framework (Truffle, Hardhat), etc.
- Contracts and dapps developed for EVM can be used with EVM++ without any change and vice versa, although any improved features will only be available on EZChain's EVM.

The planned EVM++ features are as follows.

-

5.1 Batch Transaction

A batch of transactions that is executed atomically in one legacy transaction. The execution is halted with the first call reverted, the entire state is reverted, and the last result data is the revert return. This feature allows many multi-step logics to be atomically executed: send coin/tokens to multiple accounts, approve token and call contract, etc.

A tx is a batch if it has the following input:

- to: 0x555555.....55555
- data with the first 4 bytes is the function signature of callBatch([]Tx)

```
struct Tx {
        address to
        bytes memory data
        uint256 value       // ether value to transfer
}
function callBatch([]Tx calldata txs) external ([]bytes calldata results);
```

All txs in the batch is executed in the sender context with:

- from: bacthTx's sender
- gasLimit: remain/leftover gas

The results of a batchTx is the array of each encapsulated txs' result.The execution is halted with the first call reverted, the entire state is reverted, and the last result data is the revert return.

Use Cases:

- multi-send: transfer 1 EZC to address a, 2 EZC to address b, etc...
- approve and send: approve an ERC20 token, call some 3rd party contract to spend the approved token
- deploy and call: deploy a contract, execute some functions of that contract
- call and get the logs using eth_call logs

5.2 Fee Payer

A transaction sent by one account while being paid for by another account, allowing an account with no EZC to send transactions with the help of friends or 3rd-party services.

The zero-fee transaction is wrapped in another transaction along with the sender signature. The wrapped transaction most likely will be a batch transaction, with the first inner transaction is an ERC20 transfer to the fee payer.

```
function callBatch(
        address to
        bytes memory data
        uint256 value
uint256 r, s, v                 // sender signature
) external ([]bytes calldata);
```

Confirming this transaction will have both account nonces increased and both signatures verified.

5.3 Onetime Bytecode Execution

run an arbitrary one-time bytecode without being deployed. This enables many complex logics and more efficient interactions for dapp.

Usage: User sends a transaction to a special address (0x555...555) with zero value and the following data encoded: exec(data bytes, bytecode bytes). The transaction is executed as if there bytecode bytes are deployed in the sender address, with transaction data.

5.4 Transaction Call Logs

client API to query for call status and all logs emitted from the contract execution. With this feature, clients can always know exactly what would happen in a contract call before they decide to sign it, such as: swapping an ERC20 token with a transfer fee and knowing exactly how much they will receive.

5.5 Native EZC Transfer Logs

transferring EZC emits an ERC20's `Transfer` event log for clients to track and handle accordingly.

EVM: Contract interaction can emit event logs for clients to track the state change, including ERC20/NFT transfer and balance. Unfortunately, the native coin of the network (ETH) emits no event at all.

EVM++: with native coin transfer events, dapp can easily handle native token transfer along with other ERC20/NFT tokens.

Implementation: add a log to each eth transfer call:

- address: 0x0000000000000000000000000000000000000000
- topics:
  - 0xddf252ad1be2c89b69c2b068fc378daa952-ba7f163c4a11628f55a4df523b3ef (ERC20 Transfer)
  - from: sender address
- to: recipient address
- data: 32 bytes value in wei

From the JS client:

```
provider.getLogs({
address:
'0x0000000000000000000000000000000000000000',
  topics: [
    '0xddf252ad1be2c89b69c2b068fc378daa952-
ba7f163c4a11628f55a4df523b3ef',
  ],
  fromBlock: N,
  toBlock: N,
})
```

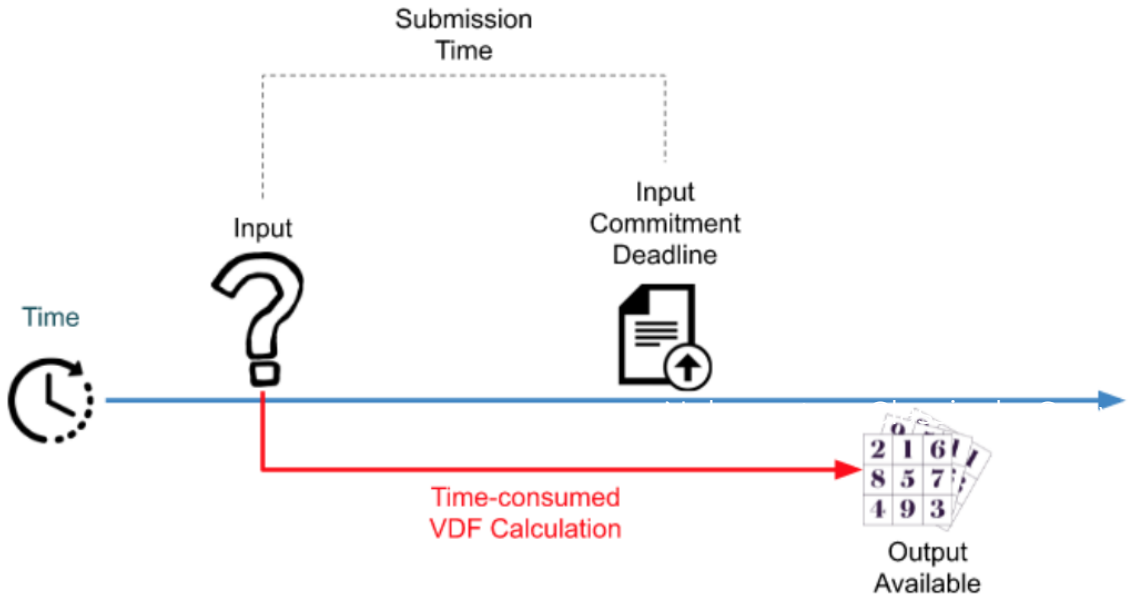
5.6 Verifiable Delay Function (VDF)

generate truly unbiased random numbers for dapps with massive values. Although it will take more time for the network to generate and confirm the result, it is not suitable for real-time applications.

The VDF implementation of POA Network and Harmony using class groups based on approaches described in the following papers:

- Simple Verifiable Delay Functions. Pietrzak, 2018
- Efficient Verifiable Delay Functions. Wesolowski, 2018

Using class groups allows the VDF proof to be publicly verifiable without the trusted setup.



5.7 Delegated Storage

allow contract code to store states in each user's account storage, which enables the protocol to implement the state rent and state pruning. This feature also allows clients to easily enumerate all contracts that have been interacted with an account, such as list all token/NFT owned by a user.

Permission Scopes:

- Contract can only access user storages delegated to its own address.
- Account cannot modify any of its delegated storages, since they're bound to the contract logics.
- Account can delete entire storages delegated to a single contract address in a transaction.
- Account cannot delete a part of the storage delegated to a single contract.

State Rent Economic and Design:

- Each account (EOA or contract) pays for its own storage usage.
- When an account is out of rent, its entire storage can be pruned without affecting any other user of the same contract logic.

6. The Native Token $EZC

The native token of EZChain platform, $EZC, is capped-supply, where the cap is set at 1,000,000,000 tokens, with 500,000,000 EZC tokens pre-minted at the mainnet launch. However, unlike other capped-supply tokens which bake the rate of minting perpetually, $EZC is designed to react to changing economic conditions. In particular, the objective of $EZC's monetary policy is to balance the incentives of users to stake the token versus using it to interact with the variety of services available on the platform. Participants in the platform collectively act as a decentralized reserve bank. The levers available on EZChain are staking rewards, fees, and airdrops, all of which are influenced by governable parameters. Staking rewards are set by on-chain governance, and are ruled by a function designed to never surpass the capped supply. Staking can be induced by increasing fees or increasing staking rewards. On the other hand, we can induce increased engagement with the EZChain platform services by lowering gas fees, and decreasing the staking rewards.

6.1 Token inflation & block reward

EZChain uses dynamical block reward [6] to avoid deflation or hyper-inflation. To that end, the reward rate will be subject to governance, within pre-established boundaries. This will allow token holders to choose the rate at which the EZC coin reaches its capped supply. Transaction fees, denoted by the set F, will be governed eventually. F is effectively a tuple which describes the fees associated with the various instructions and transactions supported in future releases. Finally, staking times and amounts will also be governable.

- Δ : EZC staking amount, defining the minimal stake required to be placed as a bond before participating in the system. The default value on genesis will be 3000 EZC coins.
- δ_{min}: The minimal amount of time required for a node to stake into the system. The default value on genesis will be 2 weeks.

  δ_{max}: The maximal amount of time a node can stake. The default value on genesis will be 52 weeks.
- γ, λ: The two key parameters in governing the minting rate function.
- F: the fee structure, which is a set of governable fees parameters that specify costs to various transactions.

[4]The premint amount (500M) will be locked and vested by smartcontract for 5 years.

EZC has a capped-supply (max supply) of 1,000,000,000 (1B) tokens. The genesis block will premint 500,000,000 EZC tokens locked in a 5-year vesting contract. The rest of the 500M EZC tokens will be minted according to the following Equation (eq.1).

6.2 Minting Function

Rj is total number of tokens at year *j*, with *R*1 = 500*M*, and *Rl* representing the last year that the values of , λ>0 were changed; cj is the yet un-minted supply of coins to reach 500M at year j such that cj≤500*M*; u represents a stakeholder, with *u.samount* representing the total amount of stake that u has, and u.stime the length of staking for u.

$$R_j = R_l + \sum_{\forall u} \rho\left(u.s_{amount}, u.s_{time}\right) \times \frac{c_j}{L} \times \left(\sum_{i=0}^{j} \frac{1}{\left(\gamma + \frac{1}{1+i^{\lambda}}\right)^i} \right) \qquad (eq.1)$$

where,

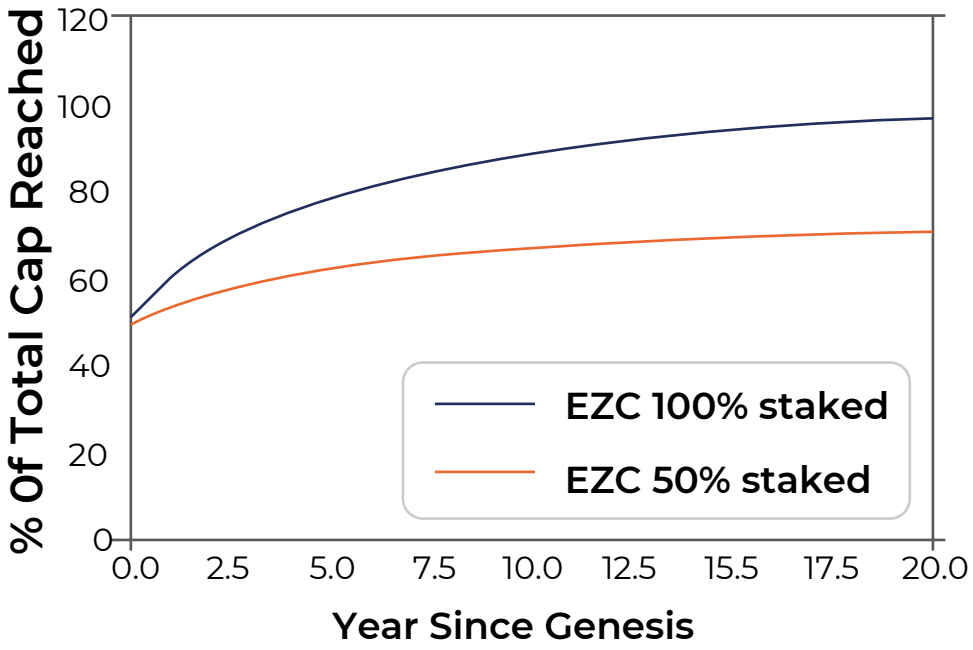
$$L = \left(\sum_{i=0}^{\infty} \frac{1}{\left(\gamma + \frac{1}{1+i^{\lambda}}\right)^i} \right) \qquad (2)$$

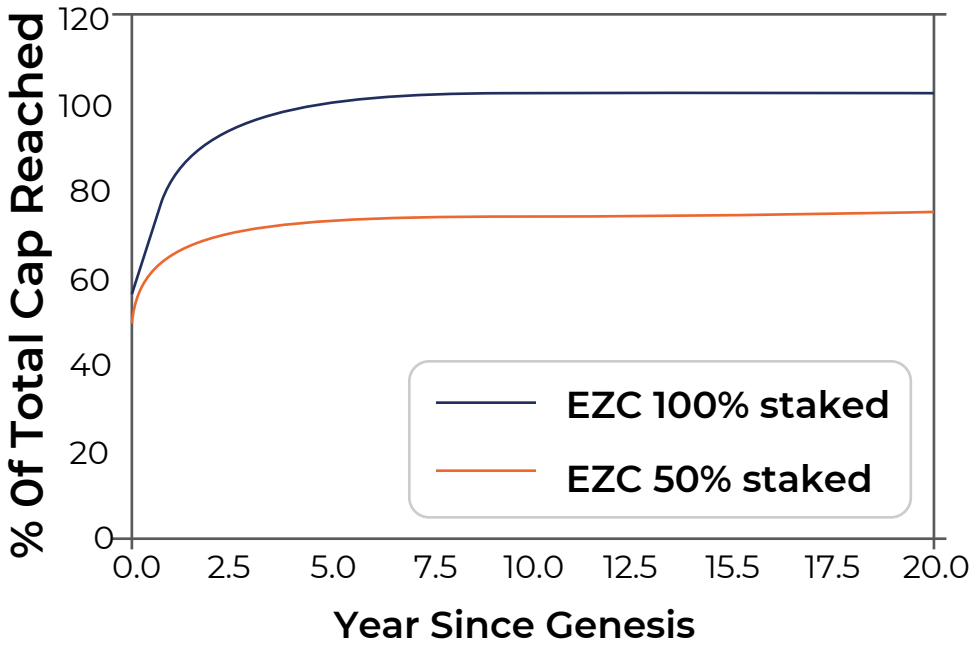
At genesis, c1=500M. The values of , λ are governable, and if changed, the function is recomputed with the new value of c*. We have that $\sum_{*} \rho(*) \leq 1.\rho(*)$ is a linear function that can be computed as follows (u.stime is measured in weeks, and u.samount is measured in EZC tokens):

$$\rho\left(u.s_{amount}, u.s_{time}\right) = \left(0.002 \times u.s_{time} + 0.896\right) \times \frac{u.s_{amount}}{R_j} \qquad (3)$$

If the entire supply of tokens at year j is staked for the maximum amount of staking time (one year, or 52 weeks), then $\sum_{\forall u} \rho\left(u.s_{amount}, u.s_{time}\right) = 1.$ If, instead, every token is staked continuously for the minimal stake duration of two weeks, then $\sum_{\forall u} \rho\left(u.s_{amount}, u.s_{time}\right) = 0.9.$ Therefore, staking for the maximum amount of time incurs an additional 11.11% of tokens minted, incentivizing stake-holders to stake for longer periods. Due to the capped supply, the function above guarantees that regardless of the number of governance changes, we will never exceed a total of 500M tokens. Therefore,

$$lim\ R_j = 500\,000\,000$$





$EZC token emissions, calculated over a 20 year and 100 year horizon, with $\gamma = 1.15, \lambda = 1.1$. The curve for "EZC (100% staked)" represents the case where every token is being staked repeatedly for the maximum staking duration of one year, i.e. $\sum_{\forall u} \rho(u.s_{amount}, u.s_{time}) = 1$. On the other hand, the curve of "EZC (lower)" represents the case where only 50% of the tokens are being staked repeatedly over the minimal staking duration of two weeks, i.e. $\sum_{\forall u} \rho(u.s_{amount}, u.s_{time}) = 0.45$. We note that, for simplicity, these graphs represent the case where , λ are fixed at genesis and never governed afterwards. The goal of changing , λ is to increase total supply of tokens in case the empirically observed total staked supply is too low.

7. Optimization

7.1 Pruning

Many blockchain platforms, especially those implementing Nakamoto consensus such as Bitcoin, suffer from perpetual state growth. This is because – by protocol – they have to store the entire history of transactions. However, in order for a blockchain to grow sustainably, it must be able to prune old history. This is especially important for blockchains that support high performance, such as EZChain.

Pruning is simple in the Snow* family. Unlike in Bitcoin (and similar protocols), where pruning is not possible per the algorithmic requirements, in $EZC nodes do not need to maintain parts of the DAG that are deep and highly committed. These nodes do not need to prove any past history to new bootstrapping nodes, and therefore simply have to store the active state, i.e. the current balances, as well as uncommitted transactions.

7.2 Client Types

EZChain can support three different types of clients: archival, full, and light. Archival nodes store the entire history of the $EZC subnet, the staking subnet, and the smart contract subnet, all the way to genesis, meaning that these nodes serve as bootstrapping nodes for new incoming nodes. Additionally, these nodes may store the full history of other subnets for which they choose to be validators. Archival nodes are typically machines with high storage capabilities that are paid by other nodes when downloading old state. Full nodes, on the other hand, participate in validation, but instead of storing all history, they simply store the active state (e.g. current UTXO set). Finally, for those that simply need to interact securely with the network using the most minimal amount of resources, EZChain supports light clients which can prove that some transaction has been committed without needing to download or synchronize history. Light clients engage in the repeated sampling phase of the protocol to ensure safe commitment and network wide consensus. Therefore, light clients in EZChain provide the same security guarantees as full nodes.

7.3 Sharding

Sharding is the process of partitioning various system resources in order to increase performance and reduce load. There are various types of sharding mechanisms. In network sharding, the set of participants is divided into separate subnetworks to reduce algorithmic load; in state sharding, participants agree on storing and maintaining only specific subparts of the entire global state; lastly, in transaction sharding, participants agree to separate the processing of incoming transactions.

In EZChain, the first form of sharding exists through the sub-network functionality. For example, one may launch a gold subnet and another real-estate subnet. These two subnets can exist entirely in parallel. The subnets interact only when a user wishes to buy real-estate contracts using their gold holdings, at which point EZChain will enable an atomic swap between the two subnets.

8. Conclusion

In this paper, we discussed the architecture of the EZChain platform. Compared to other platforms today, which either run classical-style consensus protocols and therefore are inherently non-scalable, or make usage of Nakamoto-style consensus that is inefficient and imposes high operating costs, the EZChain is lightweight, fast, scalable, secure, and efficient. The native token, which serves for securing the network and paying for various infrastructural costs, is simple and backwards compatible. $EZC has capacity beyond other proposals to achieve higher levels of decentralization, resist attacks, and scale to millions of nodes without any quorum or committee election, and hence without imposing any limits to participation.

Besides the consensus engine, EZChain innovates up the stack, and introduces simple but important ideas in transaction management, governance, and a slew of other components not available in other plat- forms. Each participant in the protocol will have a voice in influencing how the protocol evolves at all times, made possible by a powerful governance mechanism. EZChain supports high customizability, allowing nearly instant plug-and-play with existing blockchains.

Disclaimer

EZC (the unique native token of EZChain) represents the intrinsic value of the chain, and is used for network governance and consensus only. It is not designed as a security, financial instrument or any security-type, not tentative for speculation. The presale funding is used for EZChain development and adoption acquiring, not for profit in any way. In addition, the sale is only applied for sponsors, partners and accredited investors, purely funded via crypto assets and/or stablecoins (e.g. USDT), no cash accepted. Investing in crypto-currencies and tokensale has certain risks and investors should take due diligence carefully themselves. The EZChain Team doesn't intend, guarantee, or have any legal duty for any profit or loss of any investment on the project.

References

[1]	J. Caccappolo. [Online]. Available: https://crosstower.com/resources/education/blockchain-1-0-2-0-3-0/.
[2]	Unibright. [Online]. Available: shorturl.at/hklL5.
[3]	Cardano. [Online]. Available: https://cardano.org/ouroboros/.
[4]	Algorand. [Online]. Available: shorturl.at/hjGUX.
[5]	T. R. e. al, "Scalable and Probabilistic Leaderless BFT Consensus through Metastability," 2020.
[6]	S. B. e. al, "Avalanche Native Token Dynamics," 2020.