ISUSGO ECOSYSTEM WHITEPAPER

2022/01/15

ABSTRACT

ISUSGO ecosystem layer components are designed to be scalable, transparent, evolvable, security-oriented and economically resizable. It fundamentally opposes the Nakamoto consensus approach and argues that the blockchain structure can be integrated even into a simple code structure.

In favor of multiple and public development, the ISUSGO consensus structure can be used by any category of digital artifacts used in everyday life.The blockchain structure of a mainstream cryptocurrency takes on a rigid structure in order to decentralize. However, the ISUSGO structure derives its flexibility from the developers who built it. Utilitarianism is prioritized along with privacy.

As seen in recent history, a critical decision was made in the Ethereum structure, moving from PoW to PoS. This shows what the "Nakamoto" approach can lead to in developing technology. In neural networks, later decisions explain the difficulties in changing the main network structure.

Today's most advanced and internationally recognized projects (Youtube, Tiktok, Instagram, Amazon, etc.), the last holdouts of Web 2.0, have moved a little closer to Web 3.0 philosophy in recent years to encourage creators. However, as creators are still advocated, one cannot be sure of complete decentralization and how the data is used.

According to the ISUSGO consensus, anyone operating on the internet in any way can profit from privacy. Content producers always need sincere consumers. And almost everyone can be a content producer or consumer in their daily life.We should all reap the benefits of following the traces we leave, or others leave, on the Internet. Of course, this should be confidential and inaccessible to others without their permission.

According to the ISUSGO consensus, it is sufficient that the main structure is decentralized. Projects derived from ISUSGO or resized with ISUSGO foundations are consequently decentralized.

ISUSGO PLATFORM
2022/01/15

1 Introduction

This paper provides an architectural overview of the ISUSGO platform. The key focus is on the three key differentiators of the platform: the engine, the architectural model, and the governance mechanism.

1.1 ISUSGO Goals and Principles

ISUSGO is a high-performance, scalable, customizable, and secure blockchain platform. It targets 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).

Forward-looking statements generally relate to future events or our future performance. This includes, but is not limited to, ISUSGO's projected performance; the expected development of its business and projects; execution of its vision and growth strategy; and completion of projects that are currently underway, in development or otherwise under consideration. Forward-looking statements represent our management's beliefs and assumptions only as of the date of this presentation. These statements are not guarantees of future performance and undue reliance should not be placed on them. Such forward-looking statements necessarily involve known and unknown risks, which may cause actual performance and results in future periods to differ materially from any projections expressed or implied herein. ISUSGO undertakes no obligation to update forward-looking statements. Although forward-looking statements are our best prediction at the time they are made, there can be no assurance that they will prove to be accurate, as actual results and future events could differ materially. The reader is cautioned not to place undue reliance on forward-looking statements.

The overarching aim of ISUSGO is to provide a unifying platform for the creation, transfer, and trade of digital assets.By construction, ISUSGO possesses the following properties:

Scalable ISUSGO 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.

Secure ISUSGO 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,ISUSGO provides a very strong guarantee of safety when the attacker is below a certain threshold, which30 can be parametrized by the system designer, and it provides graceful degradation when the attacker exceedsthis 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.

Decentralized ISUSGO 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.

Governable and Democratic **$ISUS** 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.

Interoperable and Flexible ISUSGO is designed to be a universal and flexible infrastructure for a multitude of blockchains/assets, where the base **$ISUS** 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.

Outline The rest of this paper is broken down into four major sections. Section 2 outlines the details of the engine that powers the platform. Section 3 discusses the architectural model behind the platform, including subnetworks, virtual machines, bootstrapping, membership, and staking. Section 4 explains the governance model that enables dynamic changes to key economic parameters. Finally, in Section 5 explores various peripheral topics of interest, including potential optimizations, post-quantum cryptography, and realistic adversaries.

Naming Convention The name of the platform is ISUSGO, and is typically referred to as “the ISUSGO platform”, and is interchangeable/synonymous with “the ISUS network”, or – simply – ISUS. Codebases will be released using three numeric identifiers, labeled “v.[0-9].[0-9].[0-100]”, where the first number identifies major releases, the second number identifies minor releases, and the third number identifies patches. The first public release, codenamed ISUSGO Mainnet, is v. 1.1.56. The native token of the platform is called “$ISUS”. The family of consensus protocols used by the ISUSGO platform is referred to as the ISUS* family. There are three concrete instantiations, called ISUSGO, ISUS Chain, and ISUSXCross.

2 The Engine

Discussion of the ISUSGO platform begins with the core component which powers the platform: the consensus engine.

Background Distributed payments and – more generally – computation, require agreement between a set of machines. Therefore, consensus protocols, which enable a group of nodes to achieve agreement, lie at the heart of blockchains, as well as almost every deployed large-scale industrial distributed system. The topic has received extensive scrutiny for almost five decades, and that effort, to date, has yielded just two families of protocols: classical consensus protocols, which rely on all-to-all communication, andNakamoto consensus, which relies on proof-of-work mining coupled with the longest-chain-rule. While classical consensus protocols can have low latency and high throughput, they do not scale to large numbers of participants, nor are they robust in the presence of membership changes, which has relegated them mostly to permissioned, mostly static deployments. Nakamoto consensus protocols [5, 7, 4], on the other hand, are robust, but suffer from high confirmation latencies, low throughput, and require constant energy expenditure for their security.

The ISUS Chain* family of protocols, introduced by ISUSGO, combine the best properties of classical consensus protocols with the best of Nakamoto consensus. Based on a lightweight network samplin 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.

Mechanism and Properties The ISUS Chain* 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 doublespend? To choose amongst the conflicting transactions and prevent the doublespend, 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, ISUS Chain* protocols maintain high safety guarantees even when there are wellquantified discrepancies between the network views of any two nodes. Validators of ISUS Chain* 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 ISUS Chain* is its ability to scale without incurring fundamental tradeoffs. ISUS Chain 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 ISUS Chain-type protocols, such delegation is not necessary, allowing every node operator to have a firsthand 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, ISUS Chain* protocols achieve higher performance when the adversary is small, and yet highly resilient under large attacks.

*Asynchronously Safe*. ISUS Chain* 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 to 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. ISUS Chain* 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*. ISUS Chain* protocols, which can build a linear chain or a DAG, reach thousands of transac140 tions per second (5050+ 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 **$ISUS** are taken directly from a real, fully implemented ISUSGO network running on 2100 nodes on ISS, geo-distributed across the globe on low-end 146 machines. Higher performance results (12,019+) 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.

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

	Nakamoto	Classical	ISUS Chain*
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.Evolving technology over time causes performance differences between platforms. The ISUS Chain* protocol is already showing high performance.

3 Platform Overview

In this section, we provide an architectural overview of the platform and discuss various implementation details. The ISUSGO platform cleanly separates three concerns: chains (and assets built on top), execution environments, and deployment.

3.1 Architecture

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 ISUSGO 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 **$ISUS**.

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 $ISUS lives and is traded..

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-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 Bootstrapping

The first step in participating in ISUSGO is bootstrapping. The process occurs in three stages: connection to seed anchors, network and state discovery, and becoming a validator.

Seed Anchors Any networked system of peers that operates without a permissioned (i.e. hard-coded) set of identities requires some mechanism for peer discovery. In peer-to-peer file sharing networks, a setof trackers are used. In crypto networks, a typical mechanism is the use of DNS seed nodes (which we refer to as seed anchors), which comprise a set of well-defined seed-IP addresses from which other members of the network can be discovered. The role of DNS seed nodes is to provide useful information about the set of active participants in the system. The same mechanism is employed in Bitcoin Core, wherein the src/chainparams.cpp file of the source code holds a list of hard-coded seed nodes. The difference between BTC and ISUSGO is that BTC requires just one correct DNS seed node, while ISUSGO requires a simple majority of the anchors to be correct. As an example, a new user may choose to bootstrap the network view through a set of well established and reputable exchanges, any one of which individually are not trusted. We note, however, that the set of bootstrap nodes does not need to be hard coded or static, and can be provided by the user, though for ease of use, clients may provide a default setting that includes economically important actors, such as exchanges, with which clients wish to share a world view. There is no barrier to become a seed anchor, therefore a set of seed anchors can not dictate whether a node may or may not enter the network, since nodes can discover the latest network of ISUSGO peers by attaching to any set of seed anchors.

Network and State Discovery Once connected to the seed anchors, a node queries for the latest set of state transitions. We call this set of state transitions the accepted frontier. For a chain, the accepted frontier is the last accepted block. For a DAG, the accepted frontier is the set of vertices that are accepted, yet have no accepted children. After collecting the accepted frontiers from the seed anchors, the state transitions that are accepted by a majority of the seed anchors is defined to be accepted. The correct state is then extracted by synchronizing with the sampled nodes. As long as there is a majority of correct nodes in the seed anchor set, then the accepted state transitions must have been marked as accepted by at least one correct node.

This state discovery process is also used for network discovery. The membership set of the network is defined on the validator chain. Therefore, synchronizing with the validator chain allows the node to discover the current set of validators. The validator chain will be discussed further in the next section.

3.3 Sybil Control and Membership

Consensus protocols provide their security guarantees under the assumption that up to a threshold number of members in the system could be adversarial. A Sybil attack, wherein a node cheaply floods the network with malicious identities, can trivially invalidate these guarantees. Fundamentally, such an attack can only be deterred by trading off presence with proof of a hard-to-forge resource. Past systems have explored the use of Sybil deterrence mechanisms that span proof-of-work (PoW), proof-of-stake (PoS), proof-of-elapsed-time (POET), proof-of-space-and-time (PoST), and proof-of-authority (PoA)

At their core, all of these mechanisms serve an identical function: they require that each participant have some “skin in the game” in the form of some economic commitment, which in turn provides an economic barrier against misbehavior by that participant. All of them involve a form of stake, whether it is in the form of mining rigs and hash power (PoW), disk space (PoST), trusted hardware (POET), or an approved identity (PoA).

This stake forms the basis of an economic cost that participants must bear to acquire a voice. For instance, in Bitcoin, the ability to contribute valid blocks is directly proportional to the hash-power of the proposing participant. Unfortunately, there has also been substantial confusion between consensus protocols versus Sybil control mechanisms. We note that the choice of consensus protocols is, for the most part, orthogonal to the choice of the Sybil control mechanism. This is not to say that Sybil control mechanisms are drop-in-replacements for each other, since a particular choice might have implications about the underlying guarantees of the consensus protocol. However, the ISUS Chain* family can be coupled with many of these known mechanisms, without significant modification.

Ultimately, for security and to ensure that the incentives of participants are aligned for the benefit of the network, **$ISUS** choose PoS to the core Sybil control mechanism. Some forms of stake are inherently centralized: mining rig manufacturing (PoW), for instance, is inherently centralized in the hands of a few people with the appropriate know-how and access to the dozens of patents required for competitive VLSI manufacturing. Furthermore, PoW mining leaks value due to the large yearly miner subsidies. Similarly, disk space is most abundantly owned by large datacenter operators.Further, all sybil control mechanisms that accrue ongoing costs, e.g. electricity costs for hashing, leak value out of the ecosystem, not to mention destroy the environment. This, in turn, reduces the feasibility envelope for the token, wherein an adverse price move over a small time frame can render the system inoperable. Proof-of-work inherently selects for miners who have the connections to procure cheap electricity, which has little to do with the miners' ability to serialize transactions or their contributions to the overall ecosystem. Among these options, we choose proof-of-stake, because it is green, accessible, and open to all. We note, however, that while the **$ISUS** uses PoS, the ISUSGO network enables subnets to be launched with PoW and PoS.

Staking is a natural mechanism for participation in an open network because it enables a direct economic argument: the probability of success of an attack is directly proportional to a well-defined monetary cost function. In other words, the nodes that stake are economically motivated to not engage in behavior that might hurt the value of their stake. Additionally, this stake does not incur any additional upkeep costs (other then the opportunity cost of investing in another asset), and has the property that, unlike mining equipment, is fully consumed if used in a catastrophic attack. For PoW operations, mining equipment can be simply reused or – if the owner decides to – entirely sold back to the market.

A node wishing to enter the network can freely do so by first putting up a stake that is immobilized during the duration of participation in the network. The user determines the amount duration of the stake. Once accepted, a stake cannot be reverted. The main goal is to ensure that nodes substantially share the same mostly stable view of the network. We anticipate setting the minimum staking time on the order of a week.

Unlike other systems that also propose a PoS mechanism, **$ISUS** does not make usage of slashing,and therefore all stake is returned when the staking period expires. This prevents unwanted scenarios such as a client software or hardware failure leading to a loss of coins. This dovetails with our design philosophy of building predictable technology: the staked tokens are not at risk, even in the presence of software or hardware flaws.

3.4 Smart Contracts in $ISUS

At launch ISUSGO 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 smartcontract tools, including:

Smart contracts with off-chain execution and on-chain verification.
Smart contracts with parallel execution. Any smart contracts that do not operate on the same state in any subnet in ISUSGO will be able to execute in parallel.
Smart contracts with parallel execution. Any smart contracts that do not operate on the same state in any subnet in ISUSGO will be able to execute in parallel.

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 ISUSGO enables functionality-specific sharding through the subnets. Furthermore, if a developer requires interactions with the currently deployed Ethereum smart contracts, they can interact with the Athereum 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.

4 Governance and The $ISUS Token

-----> 4.1 The $ISUS Native Token

Monetary Policy The native token, **$ISUS**, is capped-supply, where the cap is set at 10, 000, 000, 000 tokens, with 5, 000, 000, 000 tokens available on mainnet launch. However, unlike other capped-supply tokens which bake the rate of minting perpetually, **$ISUS** is designed to react to changing economic conditions. In particular, the objective of **$ISUS**'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 ISUSGO 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 ISUSGO platform services by lowering fees, and decreasing the staking reward.

Uses

Payments True decentralized peer-to-peer payments are largely an unrealized dream for the industry due to the current lack of performance from incumbents. **$ISUS** is as powerful and easy to use as payments using Visa, allowing thousands of transactions globally every second, in a fully trustless, decentralized manner. Furthermore, for merchants worldwide, **$ISUS** provides a direct value proposition over Visa, namely lower fees.

Staking: Securing the System On the ISUSGO platform, sybil control is achieved via staking. In order to validate, a participant must lock up coins, or stake. Validators, sometimes referred to as stakers, are compensated for their validation services based on staking amount and staking duration, amongst other properties. The chosen compensation function should minimize variance, ensuring that large stakers do not disproportionately receive more compensation. Participants are also not subject to any "luck" factors, as in PoW mining. Such a reward scheme also discourages the formation of mining or staking pools enabling truly decentralized, trustless participation in the network.

Atomic swaps Besides providing the core security of the system, the $ISUS token serves as the universal unit of exchange. From there, the ISUSGO platform will be able to support trustless atomic swaps natively on the platform enabling native, truly decentralized exchanges of any type of asset directly on ISUSGO.

4.2 Governance

Governance is critical to the development and adoption of any platform because – as with all other types of systems – ISUSGO will also face natural evolution and updates. $ISUS provides on-chain governance for critical parameters of the network where participants are able to vote on changes to the network and settle network upgrade decisions democratically. This includes factors such as the minimum staking amount, minting rate, as well as other economic parameters. This enables the platform to effectively perform dynamic parameter optimization through a crowd oracle. However, unlike some other governance platforms out there, ISUSGO does not allow unlimited changes to arbitrary aspects of the system. Instead, only a pre-determined number of parameters can be modified via governance, rendering the system more predictable and increasing safety. Further, all governable parameters are subject to limits within specific time bounds, introducing hysteresis, and ensuring that the system remains predictable over short time ranges.

A workable process for finding globally acceptable values for system parameters is critical for decentralized systems without custodians. ISUSGO can use its consensus mechanism to build a system that allows anyone to propose special transactions that are, in essence, system-wide polls. Any participating node may issue such proposals.

Nominal reward rate is an important parameter that affects any currency, whether digital or fiat. Unfortunately, cryptocurrencies that fix this parameter might face various issues, including deflation or inflation. To that end, the nominal reward rate is subject to governance, within pre-established boundaries. This will allow token holders to choose on whether **$ISUS** is eventually capped, uncapped, or even deflationary.

Transaction fees, denoted by the set F, are also subject to governance. F is effectively a tuple which describes the fees associated with the various instructions and transactions. Finally, staking times and amounts are also governable. The list of these parameters is defined in Figure 1.

– **Δ** : Staking amount, denominated in **$ISUS**. This value defines the minimal stake required to be placed as bond before participating in the system.
– **δmin** : The minimal amount of time required for a node to stake into the system.
– **δmax** : The maximal amount of time a node can stake.
– **ρ** : (**πΔ**, **τ δmin**) → **R** : Reward rate function, also referred to as minting rate, determines the reward a participant can claim as a function of their staking amount given some number of π publicly disclosed nodes under its ownership, over a period of τ consecutive δmin timeframes, such that τ δmin ≤ δmax.
– **F** : the fee structure, which is a set of governable fees parameters that specify costs to various transactions.

In line with the principle of predictability in a financial system, governance in **$ISUS** has hysteresis, meaning that changes to parameters are highly dependent on their recent changes. There are two limits associated with each governable parameter: time and range. Once a parameter is changed using a governance transaction, it becomes very difficult to change it again immediately and by a large amount. These difficulty and value constraints relax as more time passes since the last change.

Overall, this keeps the system from changing drastically over a short period of time, allowing users to safely predict system parameters in the short term, while having strong control and flexibility for the long term.

5 Discussion

-----> 5.1 Optimizations

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 ISUSGO.

Pruning is simple in the ISUS Chain* family. Unlike in Bitcoin (and similar protocols), where pruning is not possible per the algorithmic requirements, in $ISUS 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 active state, i.e. the current balances, as well as uncommitted transactions.

Client Types ISUSGO can support three different types of clients: archival, full, and light. Archival nodes store the entire history of the **$ISUS** 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, ISUSGO 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 ISUSGO provide the same security guarantees as full nodes.

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 as 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 ISUSGO Borealis, the first form of sharding exists through the subnetworks 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 ISUSGO will enable an atomic swap between the two subnets.

5.2 Concerns

Post Quantum Cryptography Post-quantum cryptography has recently gained widespread attention due to the advances in the development of quantum computers and algorithms. The concern with quantum computers is that they can break some of the currently deployed cryptographic protocols, specifically digital signatures. The ISUSGO network model enables any number of VMs, so it supports a quantum resistant virtual machine with a suitable digital signature mechanism. We anticipate several types of digital signature schemes to be deployed, including quantum resistant RLWE based signatures. The consensus mechanism does not assume any kind of heavy crypto for its core operation. Given this design, it is straightforward to extend the system with a new virtual machine that provides quantum secure cryptographic primitives.

Realistic Adversaries The ISUSGO paper provides very strong guarantees in the presence of a powerful and hostile adversary, known as a round-adaptive adversary in the full point-to-point model. In other terms, the adversary has full access to the state of every single correct node at all times, knows the random choices of all correct nodes, as well as can update its own state at any time, before and after the correct node has the chance to update its own state. Effectively, this adversary is all powerful, except for the ability to directly update the state of a correct node or modify the communication between correct nodes. Nonetheless, in reality, such an adversary is purely theoretical since practical implementations of the strongest possible adversary are limited at statistical approximations of the network state. Therefore, in practice, we expect worst-case-scenario attacks to be difficult to deploy.

Inclusion and Equality A common problem in permissionless currencies is that of the “rich getting richer”. This is a valid concern, since a PoS system that is improperly implemented may in fact allow wealth generation to be disproportionately attributed to the already large holders of stake in the system. A simple example is that of leader-based consensus protocols, wherein a subcommittee or a designated leader collects all the rewards during its operation, and where the probability of being chosen to collect rewards is proportional to the stake, accruing strong reward compounding effects. Further, in systems such as Bitcoin, there is a “big get bigger” phenomenon where the big miners enjoy a premium over smaller ones in terms of fewer orphans and less lost work. In contrast, ISUSGO employs an egalitarian distribution of minting: every single participant in the staking protocol is rewarded equitably and proportionally based on stake. By enabling very large numbers of people to participate first-hand in staking, ISUSGO can accommodate millions of people to participate equally in staking. The minimum amount required to participate in the protocol will be up for governance, but it will be initialized to a low value to encourage wide participation. This also implies that delegation is not required to participate with a small allocation.

6 Conclusion

In this paper, we discussed the architecture of the ISUSGO 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 ISUSGO 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. **$ISUS** 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, ISUSGO innovates up the stack, and introduces simple but important ideas in transaction management, governance, and a slew of other components not available in other platforms. 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. ISUSGO supports high customizability, allowing
nearly instant plug-and-play with existing blockchains.

ISUSGO NATIVE TOKEN ($ISUS) DYNAMICS

2022/06/01

Abstract

This paper discusses the key implementation details, in particular the token economics (tokenomics), of the native token of the ISUSGO platform, called $ISUS. The native token secures the network, pays for fees, and provides the basic unit of account between the multiple blockchains deployed on the larger ISUSGO network. For additional details on ISUSGO, which serves as a versatile and universal platform, allowing anyone to launch new blockchains with their own rules, virtual machines, and validator sets, we guide the reader to either the accompanying architectural paper.

1 Introduction

The economic model of any new digital currency/asset is one of the most critical components of the platform that the asset resides on. This is especially true for the native token of a selfsovereign, permission-less platform, like ISUSGO. In this paper, we discuss the economic design of the native token, called **$ISUS**. The discussion is broken down into the governance properties of the token, its supply, minting (rewards) function of stakers, and other pertinent economics details, such as the transactional economy.

1.1 Key $ISUS Properties

The resources spent by a validator for staking are proportional to that validator's total stake.
*ISUS Mining* is a temporary PoW function. After the halves, it will be completely removed and PoS will be switched to PoS consensus.
In ISUSGO internal or external decentralized applications developed for e-Commerce, **$ISUS** amounts will remain in the sent wallet forever. If the receiving wallet address is no longer functional, the **$ISUS** amount is burned.
The rewards accumulated by a validator for validating are proportional to that validator's total stake
The sum of the *ISUS Ads* acquisition amount is distributed in such a way that 70% of the total acquisition amount is distributed to users in ISUSGO sub-applications. (This item may vary according to validator challenges).
**$ISUS** is a supply-limited token with a maximum limit of 10 billion tokens.
The total amount of **$ISUS** in the prize pool is 600,000,000. It was sent to an uncontrollable *ISUS Wallet* for the promotion of sub-projects or external projects and user experiences.
The total amount of **$ISUS** allocated for the development team, founders and Hackathon is 1.5 billion units. The wallets were kept locked and sent to an uncontrollable ISUS Wallet.
ISUSGO The amount of **$ISUS** allocated for the development of the ecosystem, making it scalable, fast and flexible is 700,000,000. Wallets were kept locked and sent to an uncontrollable ISUS Wallet.

The amount of **$ISUS** *reserved* for projects developed for the ecosystem with the ISUS Chain* and offering e-commerce solutions (Cryptocurrency rebates, affiliate marketing, B2B, B2C, C2C, etc.) is 1.2 billion units. wallets are kept locked and sent to an *ISUS Wallet* that cannot be controlled.
#External cryptocurrency exchange listings (Binance, Coinbase, kraken, etc.) will start with the burning of the remaining **$ISUS** from the reward pool.
No **$ISUS** Amount has been allocated for *ISUS AI*, but *ISUS AI* owns the block rewards in the remaining tiers. This ratio is only designed to be *ISUS AI* total supply <=1% max **$ISUS** supply.
Finally, the total amount of **$ISUS** released is 5 billion units.

2 Governance

We initiate our survey of the ISUSGO economic design by first discussing governance, as it plays a critical role within future components. To enable the system to adapt to changing economic conditions, the ISUSGO platform enables key system parameters to be modified dynamically based on user input. A workable process for finding globally acceptable values for system parameters is critical for decentralized systems without custodians. ISUSGO can use its consensus mechanism to build a system that allows anyone to propose special transactions that are, in essence, system-wide polls. Any participating node may issue such proposals.

3 Minting Function

Rj is total number of tokens at year j, with R1 = 5B, and Rl representing the last year that the values of , 2 R were changed; cj is the yet un-minted supply of coins to reach 5B at year j such that cj 5B; u represents a staker, with u.samount representing 85 the total amount of stake that u possesses, and u.stime the length of staking for u.

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

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

At genesis, c1 = 5B. The values of and are governable, and if changed, the function is recomputed with thenew value of c⇤. We have that P ⇤ ⇢(⇤) 1. ⇢(⇤) is a linear function that can be computed as follows (u.stime is measured in weeks, and u.samount is measured 90 in **$ISUS:**

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

If the entire supply of tokens at year j is staked for the maximum amount of staking time (one year, or 52 weeks), then P 8u ⇢(u.samount, u.stime) = 1. If, instead, every token is staked continously for the minimal stake duration of two weeks, then P 8u ⇢(u.samount, u.stime) = 0.9. Therefore, staking for the maximum amount of time incurs an additional 11.11% of tokens minted, incentivizing stakers 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 10B tokens.

Token emissions between **$ISUS** and BTC, calculated over a 20 year and 100 year horizon, with = 1.15 and = 1.1. The curve for "**$ISUS** (100% staked)" represents the case where every token is being staked repeatedly for the maximum staking duration of one year, i.e. P 8u ⇢(u.samount, u.stime) = 1. On the other hand, the curve of "**$ISUS** (lower)" represents the case where only 50% of the tokens are being staked repeatedly over the minimal staking duration of two weeks, i.e. P 8u ⇢(u.samount, u.stime)=0.45. We note that, for simplicity, these graphs represent the case where and are fixed at genesis and never governed afterwards. The goal of changing and is to increase total supply of tokens in case the empirically observed total staked supply is too low.

4 Minting Mechanism

Minting in **$ISUS** is designed to incentivize nodes to behave in a way that positively helps global outcomes. This is accomplished by special minting transactions. A node earns the right to mint by first putting up a stake and then participating actively in the consensus process. Specifically, node rewards are directly linked to their uptime and response latency. Every node maintains local information about the liveness and behavior of each other node with which it interacted. Whenever a node v is sampled by u, the latter maintains a local tuple of (response bit, timestamp). The first entry is a single bit representing whether v responded within the timeout, and the second represents the timestamp of the response. In other words, minting in ISUSGO is done via proof-of-uptime and proof-of-responsiveness. This mechanism has important consequences. In particular, since there is no "leader" accumulating rewards, there is no "rich-get-richer" compounding e↵ects.

5 Transaction Economy

-----> 5.1 Fees Structure

The fee structure in the ISUSGO platform carries several di↵erentiating features that distinguish it from other existing and upcoming platforms.

Staker Fees. Unlike other protocols that pay all fees to the elected leader, such as in Bitcoin, in ISUSGO fees are simply burned. Therefore, payment is global and for the good of the entire ecosystem. Fee burning increases scarcity of tokens in the system. The minting process o↵sets the transaction fee burning, therefore there is no danger of the system grinding to a long term halt due to gradual destruction of coins

Transaction Costs. In ISUSGO, transaction fees di↵er depending on the type of transac tion. Instantiations of new subnetworks carry the heaviest fees. In contrast, other types of transactions, such as simple payments of **$ISUS**, carry little cost. For other subnets, transactions pay fees in that subnet's native token, as well as some amount in the **$ISUS** token. A transaction native to a subnetwork may specify its own transaction fee structure, and it is up to the creator of the subnet to choose a fee structure that incentivizes validation for open, permissionless subnetworks.

.

Sliding Cost Function. Transaction fees carry a sliding-cost function. The fee is not set by the issuer of the transaction, but rather by a globally verifiable fee-function. As the congestion in the network increases, fees increase. At the end of some specific period of time, the function is recalculated to accommodate natural increases in transaction volume in the network.

Transaction Tiers. Unlike in a model such as Ethereum's, where every transaction invocation must pay some gas, ISUSGO adopts a di↵erent model that incorporates two types of transaction processing mechanisms. All keys with positive account balances will be able to immediately interact with the platform, where the fees will be based on an allotment mechanism, functionally similar to a tiered payments model adopted in cloud computing platforms. Every transaction will name a sender address (i.e. the invoker), which will be checked for current invocation allotment. If the address still has free invocations left, the transaction does not have to carry any fees attached by the sender. Past a certain amount of calls, the sender will need to attach some fees based on the resources used to compute the transaction. Additionally, users may opt instead pay for their transactions using computation. To that end, future releases will support free frequency-limited transactions, which do not require fees in coins but require some pre-computation. Whenever a new transaction is generated, the user will compute and attach a valid PoW on the transaction, which can be checked by all other parties.

5.2 Spam Management

Although simple payments do carry fees, the value will be virtually zero. This, however, can lead to spam in the network. In future releases, to prevent congestion, each transaction carries with it a local PoW. The PoW is initially of low diculty and therefore a transaction can be immediately issued with very little overhead. However, if a specific key generates a large amount of transactions in a short amount of time, each subsequent transaction will carry a larger amount of diculty in its PoW puzzle. This mechanism works in conjunction with the fee burning.

5.3 Protocol Design

We start with a non-BFT protocol called Slush and progressively build up to ISUS Chain*, all based on the same common majority-based metastable voting mechanism. These protocols are single-decree consensus protocols of increasing robustness.

```
procedure onQuery(v, col')
    if col = ⊥ then col := col'
    respond(v, col)
procedure slushLoop(u, col0 ∈ {R, B, ⊥})
    col := col0 // initialize with a color
    for r ∈ {1...m} do
        // if ⊥, skip until onQuery sets the color
        if col = ⊥ then continue
        // randomly sample from the known nodes
        K := sample(N\u, k)
        P := [query(v, col)   for v ∈ K]
        for col' ∈ {R, B} do
            if P.count(col') ≥ α then
                col := col'
    accept(col)
```

DISCLAIMER

On behalf of Isusgo ("Company") providing this white paper to the buyer, the buyer accepts it The content of this white paper is confidential to the company and the buyer agrees not to disclose, distribute or allow publication verbally, directly or indirectly, or to otherwise publish its content Isusgo, except prewritten Company consent. for purposes the "recipient" of this consent includes, without limitation, any principal, employee or representative of the buyer. This whitepaper and the proposals made in it this is for Participants only. This is white paper provides a summary of its key features. company. Contains only general recommendations prepared without goals of any participant, financial situation or needs. Participants must read Examine the white paper carefully and information is suitable for them Respect for your goals, financial situation and needs. This white paper does not include: all possible information participant may request. In every situation, interested parties should run their own business Company investigation and analysis and the data contained in this white paper Any company representation or warranty with respect to accuracy or completeness of information included in this white paper. Besides, The company will not have any liability. to the buyer or any person who arises trust in this white paper decide to apply for company shares. company, financial and the non financial information it contains this white paper reasonable knowledge and to the best of his ability. However, buyers should trust themselves researching all financial information and No representations or warranties are or will be made to be made by the Company regarding the accuracy of or the completeness of such information. The company does not make any representations about the underlying value of tokens on offer. Prospective participants must do reasonable value.

These risks and uncertainties include factors. and industry-specific risks As the company generally operates economic conditions. real performance or events may differ materially from what is expressed or implied therein. expressions. All forward-looking statements are attributable Persons acting on behalf of the company or Your company is clearly qualified warning statements and their integrity in this chapter. Except when clearly necessary Legally, the Company does not make any commitments.

Obligation to publicly update or revise forward- looking statements provided As a result of this technical review, new information, future events or otherwise, or Risks affecting this information. The company, its officers or any Person named in this whitepaper their consent or any relevant person The preparation of this white paper, any representation or warranty (express or implied) with respect to accuracy or probability performance of forward- looking statements except to the extent required by law. How are you today forward- looking statements reflect opinions retained only as of the date of this whitepaper.

PARTICIPANT ALERT

High turnout for a token offer risks. Highly speculative and previously participate in any project about information is given, possible participants It is highly recommended to look for the suitable professional advice; Information contained in this white paper Prepared by or on behalf of Company. isusgo did not take any action. independent review of information contained in this White paper.

FEATURED COMMENTS

Information contained in this whitepaper disclaimer about the proposed job opportunity It is not intended to be the only information on which decision will and will not be made a document in lieu of a disclosure document, or any other notice that may be required under law. Detailed information may be required for make a token participation decision; Prospective participants should be aware There is no established market for trading of any tokens that may be offered.

FUTURE STATEMENTS

Apart from historical information, There maybe topics covered in this white paper. forwardlooking statements. Such expressions are only estimates and are subject to inherent risks and uncertainty. Based on forward-looking statements, assumptions and estimates and explain The company's future plans, strategies, and expectations can often be defined By using the words 'anticipate', 'will', "believe", "guess", "plan", "expect", "intend", 'search' or similar expressions. Participants They are warned not to over-confidence. about forward-looking statements. by him nature contains forwardlooking information numerous assumptions, inherent risks and both general and specific uncertainties contributing to the possibility of forecasts, forecasts, projections and other No forward-looking statements will be made.

To have these tokens do not constitute a share of equity or ownership in the company. token The economy is new and exciting. Regulator conditions may require this token The mechanics can be replaced or replaced. ISUSGO tokens do not have any rights, uses, purpose, attributes, functions or features, express or implied, including, without limitation, any use, purpose, features, functions or features ISUSGO platform. Company warrants and does not represent in any way ISUSGO tokens have any rights, uses, purpose, attributes, functions or features. ISUSGO tokens value. The company reserves the right to: Refusing or canceling the ISUSGO coin purchase may request it at any time at its sole discretion.

NOT A SECURITY

It is important to note that any token is given. Not intended on Isusgo's platform be a security and this document prospectus, tender document or a request for an investment in a stock or share offer. Tokens issued on our site platform referred to in this document not give any property or debt Inside Isusgo's ecosystem. Currently tokens Trading or future issues are non-refundable. Isusgo doesnot make any guarantees value, secondary market or commitments the value of such tokens. Buyer and owners will participate in each economy in its own time. the only risk.

References

1. Bitcoin: bitcoin/bitcoin (Oct 2018), https://github.com/bitcoin/bitcoin
2. Douceur, J.R.: The sybil attack. In: International Workshop on Peer-to-Peer Systems. pp. 251–260. Springer (2002)
3. itcoin-ng: A scalable blockchain protocol. In: 13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016, Santa Clara, CA, USA, March 16-18, 2016. pp. 45–59 (2016), https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal 5. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
4. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger (2014)
5. M. Stigmar.: Peer-to-peer Teaching in Higher Education: A Critical Literature Review pp. 124-136 (2016)
6. Z. Li.: A blockchain-based framework of cross-border e-commerce supply chain (2019)
7. Guttmann, Affiliate marketing spend in the U.S.2010-2022, (2019)
8. W.Nichols, Advertising Analytics 2.0. Harvard business review, 91(3), 60–68, (2013)
9. P. S.Almeida and J. J.Gondim, Click Fraud Detection and Prevention System for Ad Networks. Journal of Information Security and Cryptography (Enigma), 5(1), 27–39, (2018)
10. D.Brown and N.Hayes, Influencer marketing. Routledge, (2008)
11. A.Cavoukian, Privacy by Design: The 7 Foundational Principles, (2011)
12. M. Chehrehpak, S. P. Afsharian and J. Roshandel, Effects of implementing information security management systems on the performance of marketing and sales departments. International Journal of Business Information Systems, 15(3), 291–306, (2014)
13. C.Gronroos, From marketing mix to relationship marketing: towards a paradigm shift in marketing. Asia-Australia Marketing Journal, 2(1), 9–29, (1994).
14. P. Coll and J. L. Micó, Influencer Marketing in the Growth Hacking strategy of digital brands. Observatorio (OBS*), 13(2), (2019)
15. S. G. Cort, Industry corner: industrial distribution: how goods will go to market in the electronic marketplace. Business Economics, 53–55, (1999)
16. K.Jones, How Much Will Blockchain Really Affect Digital Marketing?Forbes (2018)
17. M. Nofer, P. Gomber, O. Hinz and D. Schiereck, "Blockchain", Bus. Inf. Syst. Eng., vol. 59, pp. 183-187, Mar. 2017
18.S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, [online] Available: https://bitcoin.org/bitcoin.pdf.
19. S. Haber and W. S. Stornetta, "How to time-stamp a digital document", J. Cryptol., vol. 3, pp. 99-111, Jan. 1991
20. V. Buterin, Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform, 2013, [online] Available: https://github.com/ethereum/wiki/wiki/White-Paper.
21. Y. Amihud Illiquidity and stock returns: cross-section and time-series effects J. Financ. Mark. (2002)
22. J.A. Batten et al. Liquidity, surprise volume and return premia in the oil market Energy Econ. (2019)
23. Common variable immunodeficiency disorders: division into distinct clinical phenotypes Blood (2008)