



**Partisia
Blockchain**

Yellow Paper

Yellow paper describing the complete infrastructure and the implemented features as well as the inherent security and incentive mechanisms.

Version 0.98 / April 29 2022

Table of contents

1 Introduction	4
2 Economic and technical overview	8
2.1 The overall architecture	8
2.2 Create accounts	10
2.3 The node operators	10
2.3.1 Whitelisting node operators	11
2.4 Selecting node operators	13
2.5 Smart Contracts	13
2.6 Pricing and paying for blockchain usage	14
2.6.1 Pricing and Payment Schemes	15
2.6.2 Smart contract payment mechanisms	17
2.7 Exchange risk	19
2.8 Staking as node operator	20
2.8.1 Staking as a Baker node	21
2.8.2 Staking as a ZK node	22
2.8.3 Staking as Oracle node	22
2.9 Staking and utility for all	23
2.9.1 Delegated community staking	23
2.9.2 Mini nodes	24
2.9.3 Smart contract staking	24
2.9.4 MPC tokens as a freely traded asset	25
2.10 Scoring	25
2.11 Software update	26
3 The Basic Blockchain Services	26
3.1 Consensus and block creation	27
3.1.1 Proof-of-Verification	28
3.1.2 FastTrack consensus	29
3.2 Sharding	32
3.3 The payment scheme for basic blockchain services	34
4 Zero-knowledge computation	37

4.1 Introduction to private smart contracts	38
4.2 MPC orchestration	38
4.3 Pre-processing	40
4.4 Payment schemes	40
4.5 Scoring	40
4.6 Staking	40
5 Oracle services	41
5.1 Cross-chain security models	41
5.2 Bring Your Own Coins (BYOC)	43
5.2.1 Staking	46
5.3 Token Bridge	49
5.3.1 Staking	51
6 Notation and definitions	55

1 Introduction

This document describes the cryptographic and economic mechanisms that constitute the backbone of Partisia Blockchain as a self-organized decentralized infrastructure operated by independent node operators.

Partisia Blockchain is a complete semi-permissioned public blockchain offering generic coordination of public and private information. The blockchain is designed to solve for privacy and interoperability across all platforms.

- Privacy is leveraged through decentralized zero-knowledge computation, most notably secure multiparty computation (MPC). This is all about protecting data in all its possible stages: At rest, in transit and in use, which supports a new security paradigm for collaborative solutions with decentral control and privacy.
- Interoperability is leveraged by the decentralized control, privacy and specially designed oracle services allowing for transfer of values and data.

As a decentralized infrastructure, Partisia Blockchain is operated and governed by the node operators. The role of the Partisia Blockchain Foundation is to mint tokens and to ensure that the funds raised are properly used to develop the software and make it available for the node operators. There are the following four types of nodes:

- **Reader nodes**, a read only observer role available for anyone without whitelisting. Unlike the other types of nodes, Reader nodes are not compensated.
- **Baker nodes**, operating the basic blockchain with consensus and governance.
- **ZK nodes**, selected to run zero-knowledge computations.
- **Oracle nodes**, selected to run the oracle services.

A basic feature is BYOC (Bring Your Own Coin) where external coins are used as means of payment for services on Partisia Blockchain. BYOC brings in external coins or tokens from users, which are represented as BYOC Twins on the Partisia Blockchain. The users can select any external coin or token supported by the network as means of payment and the spent BYOC Twins are distributed to the node operator who converts the BYOC Twins back into the original coins or tokens. On the other hand, the MPC token can be traded freely although its functionality is limited to staking on Partisia Blockchain. The result of this is a decoupled token economy where staked MPC tokens enable a node operator to operate a node and receive fees in external coins and where the staked MPC token enhances the cryptographic mechanisms. In economic terms the node operators are the residual claimants of fees and the MPC tokens are the entry barrier. Also, for services where staking provides additional guarantees, MPC tokens

function as a dynamic entry barrier where more staking generates more fees. This basic token economy is illustrated in Figure 1.

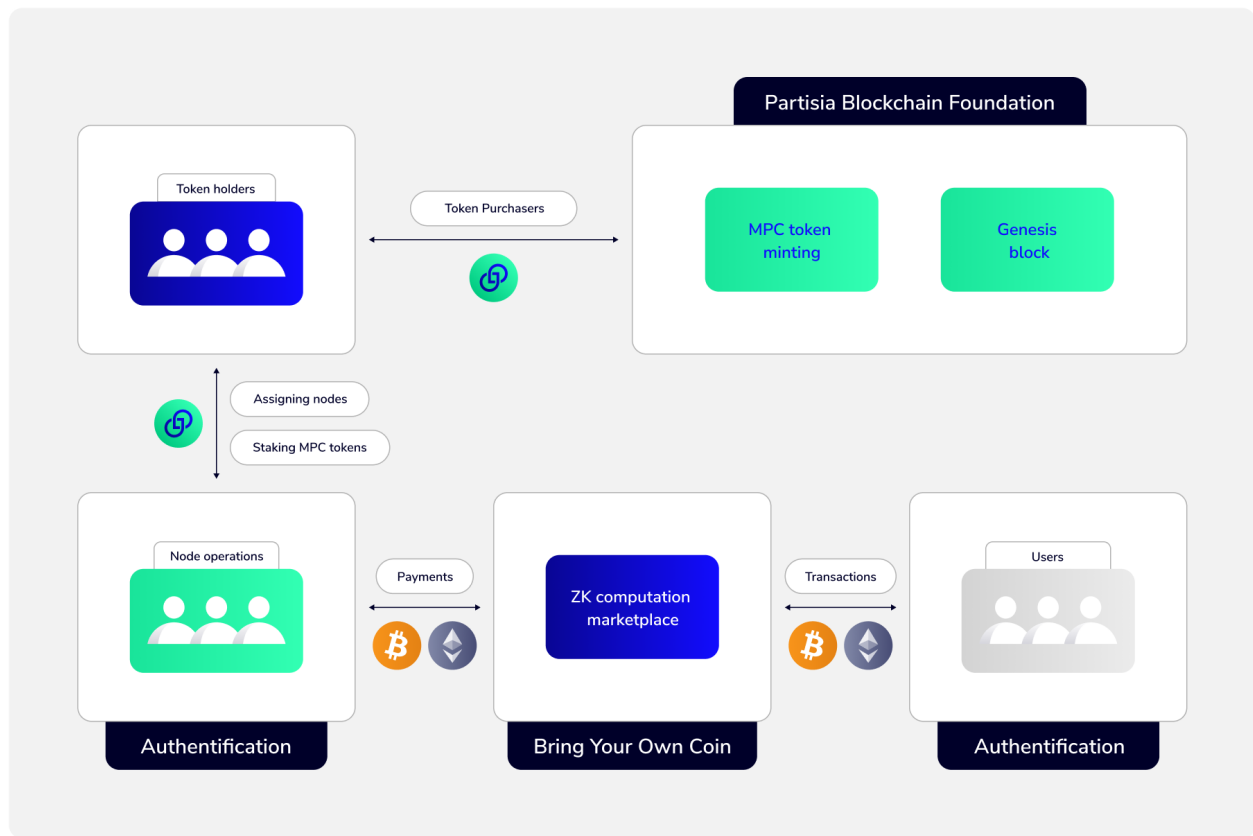


Figure 1: The basic token economy

The basic staking by node operators will gradually be extended with a number of different additional staking opportunities for all token holders. Examples of these opportunities are: 1) “delegated community staking” where token holders delegate stakes to node operators, 2) mini nodes tailored various small tasks and 3) staking smart contracts to keep them on-chain in case they run out of gas. Section 9.2 provides an overview.

Another basic design choice is the two types of services offered:

- *The Basic blockchain services* cover block creation, consensus and governance operated by all whitelisted Baker nodes.
- *The ZK and Oracle services* cover integrated services like BYOC, ZK computation and Token Bridges operated by a selected subset of whitelisted ZK or Oracle nodes.

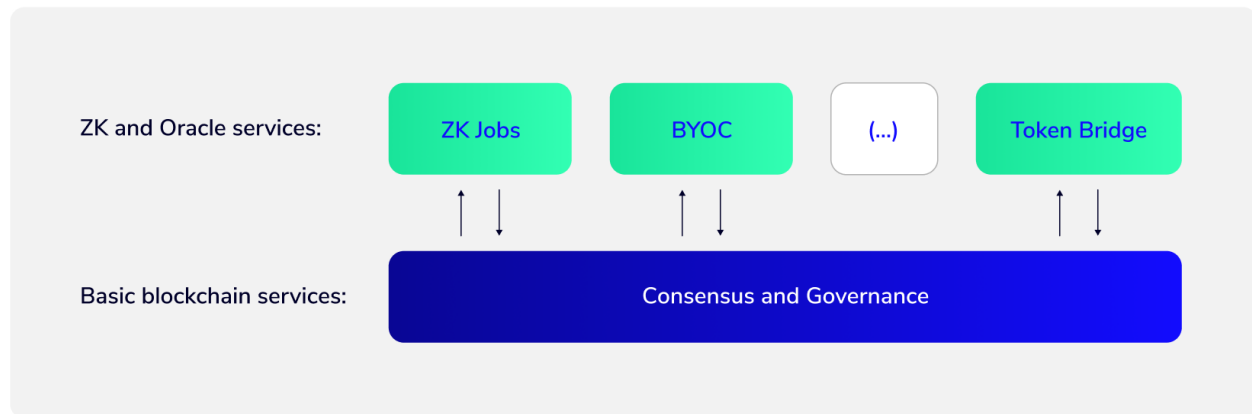


Figure 2: The two type of services

Figure 2 illustrates the separation between the Basic blockchain services and the ZK and Oracle services with respect to the real effort from the involved node operators. To give an example, BYOC is operated by a selected subset of whitelisted Oracle nodes, however, the BYOC service relies on basic blockchain services. This separation allows for separate incentive models for the two types of services as further described in Section 2. From a technical perspective, all services are instructed by public and private smart contracts. Since ZK and Oracle services rely on basic blockchain services, interacting with the smart contracts that define e.g. the BYOC feature, involves interacting with both types of services.

Partisia Blockchain is both a basic blockchain with consensus and governance - a *first layer* - as well as a *second layer*, with interoperable ZK and Oracle services delivered as-a-service across blockchains and other platforms. There are a number technological and economical reasons for this design choice:

From a technological point of view, the basic blockchain service ensures that the Partisia Blockchain can smoothly evolve and offer the same high quality of ZK and Oracle services across platforms. The main drivers are:

- That the basic blockchain service is designed for orchestrating generic ZK computation. In particular, the basic blockchain service provides consensus and integrity that allows for a more trustworthy execution of ZK computation technologies.
- That the cross-chain functionalities rely on the same high security provided by Partisia Blockchain.
- That the basic blockchain service brings in whitelisted computation nodes ready for ZK and Oracle services.

With this in place, existing blockchains or services on existing blockchain can easily integrate with Partisia Blockchain to solve for privacy and interoperability as simple as integrating with an API.

From an economic point of view, the blockchain allows for tailored incentive provision. On the one hand, revenue splitting is designed to align node operators' behaviour in the basic P2P layer and incentivize the desired behaviour across the different services on the Partisia Blockchain. On the other hand, the build-in staking mechanisms introduce economic punishments that provide users with enhanced guarantees against collusive behaviour that attempt to circumvent the thresholds built into the cryptographic protocols. Also, the basic blockchain provides the starting point for a market for trust where the reputation of the individual nodes becomes part of the market mechanisms selecting and pricing ZK and Oracle nodes and services. Finally, the integrated BYOC functionality aligns Partisia Blockchain with other first layer blockchains in a similar fashion as a pure second layer solution. To see this, note that all payments are made in coins or tokens provided by the first layer partner to Partisia Blockchain. This is important for growing the network, which ultimately strengthens both the cryptographic and economic incentives built into the network.

Or in other words, all of the technical and economic difficulties involved in applying advanced decentralized ZK computation technologies has been sorted out up front by the Partisia Blockchain network.

The yellow paper is structured as follows: Section 2 provides a brief technical and economic overview and the Basic blockchain service is presented in Section 3. ZK and Oracle services are presented in Section 4 and 5.

2 Economic and technical overview

A public blockchain is an economic system that applies cryptographic and economic mechanisms to enable and incentivize independent node operators to operate the network and make it available for anyone to use.

In this section we provide the basic technical and economic building blocks applied to Partisia Blockchain. The concrete services offered are detailed in Section 3 and 4.

2.1 The overall architecture

The Partisia Blockchain infrastructure consists of the layers illustrated in figure 3 below and the different layers are designed to achieve the following properties:

- P2P communication: Designed to achieve fast and robust communication.
- Consensus and states: Designed to achieve fast and robust consensus and finalization.
- Public smart contracts: Designed for simple, efficient and secure execution of public computations.
- Private smart contracts: Designed for simple, efficient and secure execution of ZK computations.
- API: Designed to simplify the use of Partisia Blockchain across platforms.
- Built-in interoperability applications: Designed for simple, efficient, robust and secure transfer of values and confidential use of data.

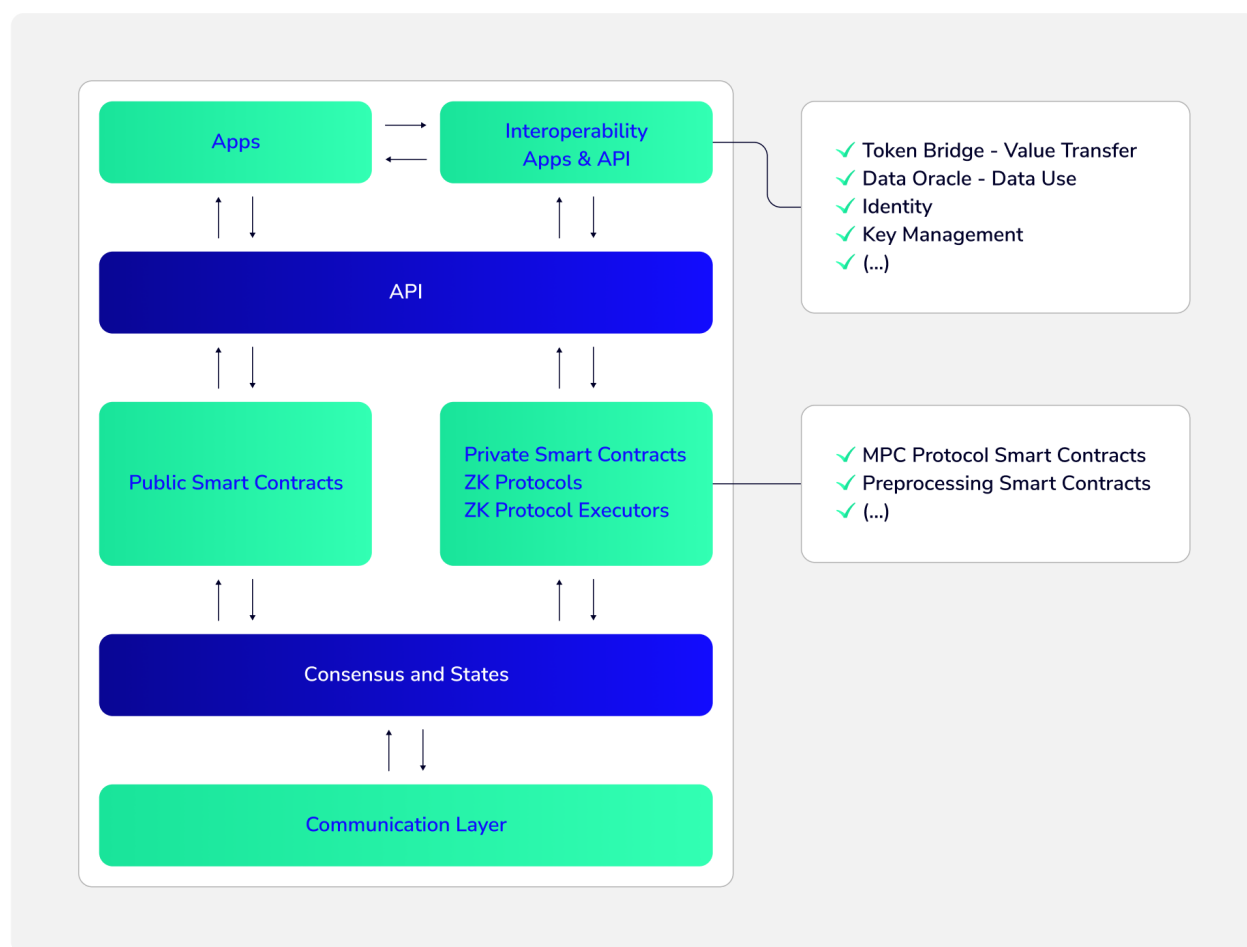


Figure 3: Partisia Blockchain layers

The single most important contribution by Partisia Blockchain is the ability to run any privacy-preserving computations in combination with computations on public information. The architecture addresses this through a deliberate separation of private and public coordination, which enables developers of applications to tailor the use of public and private computation. This design choice is important to address different regulatory as well as business requirements across jurisdictions - including somewhat conflicting regulation on transparency and data protection. As an example of this balance, the public ledger may contain public information about who is trading what and the privacy-preserving computation may enable full auditability into prices and other private information involved in the transaction.

2.2 Create accounts

For node operators and users that directly interact with Partisia Blockchain, the initial interaction with the Partisia Blockchain automatically creates an account.

Services that bring in users that do not directly interact with Partisia Blockchain are responsible for KYC of these users. To support this, decentralized privacy-preserving KYC will be provided as a basic service on the Partisia Blockchain, which ensures that all services on the Partisia Blockchain can be fully compliant from AML to data protection regulations.

2.3 The node operators

Partisia Blockchain consists of a set of protocols operated in a P2P network of computing nodes. Being a decentralized network, the basic decision-making units are the node operators, where the Partisia Blockchain defines the possible set of decisions to be made by the network.

There are the following 4 types of node operators:

- **Reader nodes:**
 - Role: Read-only observer role to be used by any who needs to keep track of the activities on Partisia Blockchain such as external services, application providers etc.
 - Whitelisting: There is no whitelisting and anyone can set up a Reader node.
 - Compensation: There is no compensation to Reader nodes.
 - Staking: There is no staking involved in operating a Reader node.
- **Baker nodes:**
 - Role: Operate the basic blockchain with consensus and governance.
 - Whitelisting: There is a decentralized whitelisting process for Baker nodes with external validators.
 - Compensation: Baker nodes share all fees paid for providing the basic blockchain services.
 - Staking: Minimum 25,000 MPC tokens.
- **ZK nodes:**
 - Role: Operate MPC and ZK computation.

- Whitelisting: There is a decentralized whitelisting process for ZK nodes with external validators.
- Compensation: ZK nodes share all fees paid for providing the MPC and ZK computations minus any basic blockchain services.
- Staking: Minimum 100,000 MPC tokens. However, MPC tokens will be locked to secure values across ZK jobs and ZK nodes with too few unlocked MPC tokens will not be eligible. Hence, the market drives the demand for MPC tokens.
- **Oracle nodes:**
 - Role: Selected to operate the different Oracle services such as BYOC and Token bridges.
 - Whitelisting: There is a decentralized whitelisting process for Oracle nodes with external validators.
 - Compensation: Oracle nodes share all fees paid for providing the Oracle services minus any basic blockchain services.
 - Staking: Minimum 250,000 MPC tokens. However, MPC tokens will be locked to secure values across Oracle jobs and Oracle nodes with too few unlocked MPC tokens will not be eligible. Hence, the market drives the demand for MPC tokens.

2.3.1 Whitelisting node operators

As a public semi-permissioned blockchain, all entities that meet the requirements for operating a node can initiate a decentralized whitelisting process.

The whitelisting process is entirely orchestrated through smart contracts on the Partisia Blockchain and the Partisia Blockchain Foundation has no role in either parts of the process.

The basic mechanism for whitelisting node operators is as follows:

1. Node operator creates an account on Partisia Blockchain.
2. The Node operator fuels the whitelisting smart contract running the whitelisting process.
3. Node operator provides information that is made publicly available:

- a. Name and website for the legal entity controlling the node.
 - b. Jurisdiction for the legal entity controlling the node and the jurisdiction of the location of the actual servers.
 - c. A confirmation stating that the legal entity operates one node only (the node however may cover all services i.e. Baker, ZK and Oracle services).
 - d. The above information is confirmed by one or more external validators who have integrated with the Partisia Blockchain and receive rewards for the validation work.
4. Node operator commits stakes and performs test run
 - a. The node operator stakes the required MPC tokens required for operating a Baker, ZK and/or Oracle node.
 - b. The node operator passes an operational test computation reflecting the respective role of Baker, ZK and/or Oracle nodes.
5. Node operators are accepted by the network if:
 - a. The publicly available information has been signed by one or more external validators who are available through the Partisia Blockchain.
 - b. Sufficient stakes have been committed.
 - c. The test run has been successfully conducted.
 - d. The whitelisting smart contract has been paid.

The node operator candidate pays for the above process with BYOC to the respective smart contracts. The cost covers the external validation, test computations which may involve other whitelisted ZK or Oracle nodes as well as the basic transactions involved in the whitelisting process.

The external validation replaces validation through voting rules among the incumbent node operators and hereby avoiding so-called *market foreclosure* where incumbent node operators block new node operators from entering the network. As the network grows the node operators will become part of a market for trust where the individual node operator's trust score will become an integrated part of the market for services on the network. Also, as proper quality assessment measures have been established, the whitelisting status shall be renewed yearly for the lowest scoring node operators as well as for any randomly selected nodes. Otherwise the whitelisting status of a node can only be revoked by the node operator itself.

2.4 Selecting node operators

Different services offered by Partisia Blockchain may require a selected subset of nodes to perform certain tasks.

The Basic blockchain service is operated by a committee of Baker nodes. As long as the total number of Baker nodes is smaller than 1,000 there is only a single committee consisting of all the nodes. When the total number of whitelisted Baker nodes exceeds 1,000, the committee is a random selection of 1,000 whitelisted Baker nodes.

The ZK and Oracle services are performed by select subsets of ZK or Oracle nodes. For the initial selection model, however, the selection mechanism is kept simple and subsets of nodes are treated equal for all types of ZK or Oracle services based on a uniform distribution.

As the number of nodes grows, the selection mechanism will gradually become part of the economic model and regulatory compliance for Partisia Blockchain:

- Direct selection:
 - Users select node operators directly.
- User-defined jurisdiction:
 - Random selection based on uniform distribution among eligible nodes.
- User-defined staking limits:
 - Random selection based on uniform distribution within eligible nodes.
 - Trust score adjustment.
 - Random selection adjusted based on trust score.
- Market for trust:
 - Market mechanisms select nodes and set prices based on both bids and trust scores.

2.5 Smart Contracts

As any other blockchain, smart contracts are the developers' interface to program the network to perform the desired operations. The generic public and private coordination of information that comes with Partisia Blockchain involves the following three distinct types of smart contracts: Public, Private and System smart contracts.

Public smart contracts

Public smart contracts are similar to regular smart contracts on other blockchains. The contract has three types of interaction:

- *Create* allows the contract to be deployed on the blockchain.
- *Interact* allows the contract to perform the involved instructions and interact with the world.
- *Destroy* allows the contract to be taken down. A system contract automatically takes down contracts after a period of time as the users have little incentives to do so.

A public contract knows the contract state and the execution context (current transaction) and interacts with other contracts through event transactions (a transaction initiated by another transaction).

Private smart contracts

A private smart contract orchestrates ZK computations (most notably MPC) on the blockchain. Section 4 describes how ZK computations are orchestrated and the different private smart contracts are involved.

System smart contract

System smart contracts are concrete implementations of the abstract governance model. System contracts can manipulate accounts and their balances, they can create contracts and change the way accounts and consensus is handled on the chain.

2.6 Pricing and paying for blockchain usage

Fees on Partisia Blockchain are paid for with BYOC Twins, which are external coins or tokens represented on Partisia Blockchain. The only way a user can spend their BYOC Twins is by interacting with a smart contract and sending a signed transaction with a specified spending or cost.

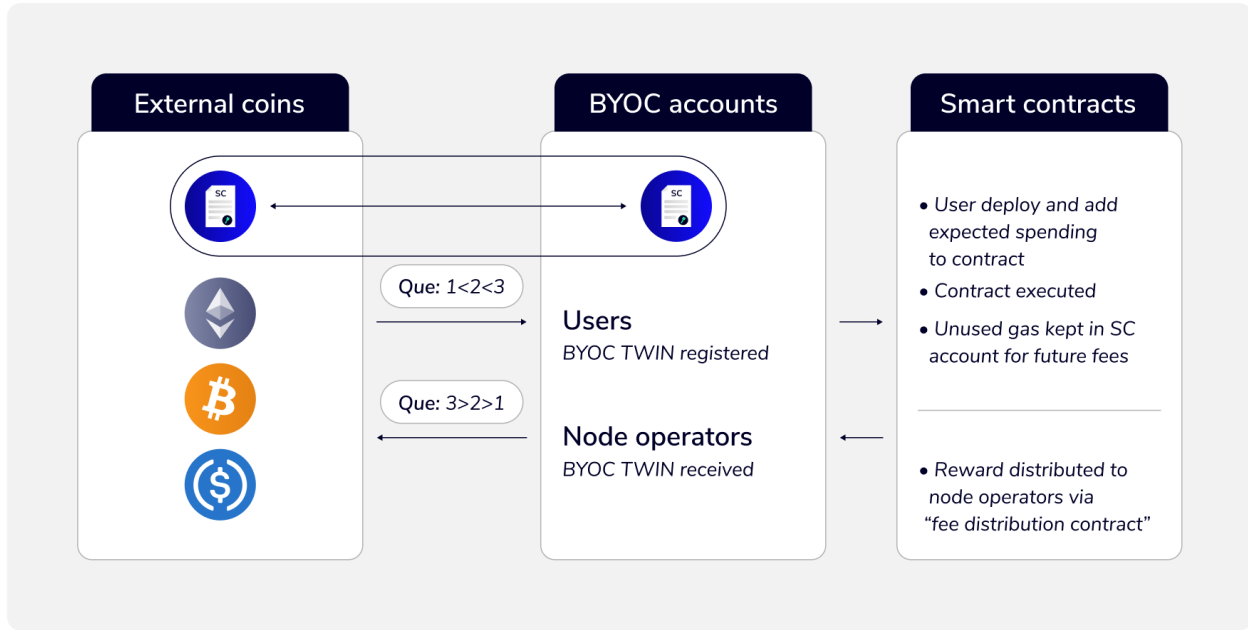


Figure 4: Paying for the use of Partisia Blockchain with BYOC

Figure 4 illustrates the interaction between external coins, users' and node operators' BYOC accounts and the smart contracts on the Partisia Blockchain. The simple process illustrated is as follows: A user can move external coins in and out of the BYOC account and use the resulting BYOC Twins to activate/interact with smart contracts. The smart contracts are executed by the Baker and selected ZK or Oracle nodes that are paid the fees involved in operating the smart contracts. BYOC Twins are released by the involved smart contracts upon completion of the tasks. Potential congestion involved in converting in and out of BYOC is managed by a queue.

As illustrated in Figure 4, "External coins" and "BYOC accounts" are managed by the BYOC Oracle service that safeguards the use of external coins as BYOC Twins. External coins live on external blockchains and the creation and consumption of BYOC Twins is managed by the BYOC Oracle. Section 5.2 describes the BYOC Oracle in more detail.

2.6.1 Pricing and Payment Schemes

We distinguish between what the users pay (pricing schemes) and what the computation nodes receive (payment schemes).

Pricing schemes are the fees paid by users and the initial prices are “posted prices” set by the Partisia Blockchain Foundation.

Based on market projections and estimated costs involved in operating a node, the follow initial tentative prices has been chosen:

- **Basic fees** prices the basic blockchain services and consist of:
 - Network fee: 5 USD cents/kb.
 - CPU fee: 5 USD cents per 1000 instructions.
 - Storage fee: 1 USD cent/kb per year.
- **ZK and Oracle fees** prices the Oracle services:
 - BYOC (tentative):
 - 0,1% of the value transferred with a threshold of 25 USD as minimum transfer (equal to a BYOC fee of 2,5 USD cents).
 - ZK computation¹
 - Basic blockchain services priced with *Basic fees*.
 - MPC multiplication fee (tentative): 5 USD cents per 1000 multiplications.
 - MPC triples fee (tentative): 5 USD cents per 1000 triples.
 - Staking fee (tentative): 1% of the locked staked value per month.
The locked stakes for ZK computations are set by the users.
 - Token Bridge
 - 1% of the value transferred with a threshold of 25 USD as minimum transfer (equal to a Token Bridge fee of 25 USD cents).

The fees may change and future versions of Partisia Blockchain will gradually involve market mechanisms for setting fees and for selecting nodes for ZK and Oracle services. Though, with the current prices a simple token transfer transaction costs one USD cent.

If a smart contract runs out of gas (BYOC fees) it stops working until more gas is added. This also goes for storage, meaning that the smart contract will be removed from the blockchain if it runs out of gas for storage. On one side this prevents

¹ The prices are based on a subset of 3 ZK nodes and the MPC protocol REAL. Involving more ZK nodes per cluster as well as other MPC protocols will impact the prices.

congestion by vast amounts of unused storage, on the other hand, removing a smart contract is a critical event. To address the later two measures are introduced:

1. The smart contract language allows developers of dApps to reserve fees for storage and hereby postpone the removal.
2. Staking the smart contract with MPC tokens, which will turn the smart contract into passive storage if it runs out of gas for storage. Refuelling the smart contract brings it back to active use.

The cost of passive storage is negligible and staking smart contracts is not a payment to node operators. It is however a utility for the network as a whole as an additional active use of MPC tokens.

Payment schemes define how the collective fees are distributed among the involved node operators.

The total revenue from the Basic blockchain services, R is allocated to the Baker nodes for each basic blockchain epoch which is initially set to 1,000 blocks. R is the collective fees given by the metered use of network, CPU and storage multiplied with the basic fees. The allocation of R is determined by the incentive scheme presented in Section 3.3.

The revenue from the ZK and Oracle services is for the most part allocated equally among the ZK and Oracle nodes involved in the respective ZK and Oracle services. Section 4 and 5 explains in more detail how ZK and Oracle nodes are compensated.

All collected fees will be subject to a small tax carved out for special infrastructure tasks such as the job of mirroring information across blockchains used for token bridges described in Section 5.1. The tax proceeds are used for infrastructure tasks only and will be introduced at a later stage.

2.6.2 Smart contract payment mechanisms

Whenever a user interacts with a smart contract the transaction will carry a certain amount of BYOC Twins or “gas” to run the smart contract. Some of this gas will be

spent on Basic blockchain services (network, CPU and storage fees) and some of it might be spent on ZK and Oracle services.

Initially, when a contract is deployed, gas is used to conduct the involved operations and the remaining balance is deposited in a SC account (Smart Contract account) on the smart contract right after the deployment. The SC account holding is used to pay for future transactions and most notably storage fees that are paid periodically throughout the lifetime of a smart contract. Gas may also come from other smart contracts i.e. via event transactions. It is essentially the smart contract developer who decides whether the payment comes directly from the user or from the SC account.

Public and Private smart contracts need to be able to pay recurring storage fees, otherwise the contracts will be removed from the blockchain (it can be reinstated through a governance procedure). These fees require that the contract has a balance of gas at all times. To allow supplying gas to a contract the system will automatically move any unspent gas from an interaction into the SC account of the contract. That is, if User A interacts with Contract B using a transaction with a cost of 1,000 gas, and the fees for the transaction amount to 800 gas, then the remaining 200 gas will be deposited into the SC account of Contract B to be used for future recurring storage fees.

It is the actual transactions following a smart contract interaction with the blockchain that execute the involved network and CPU fees.

Signed transactions are sent to the blockchain, which deducts the allocated cost from the user's balance. Afterwards the transaction is forwarded to the relevant smart contract. The involved event transactions pay network fees and CPU. Remaining unused cost is transferred to the smart contract and it is used for storage fees (as in the example above).

The paid fees are distributed to the Baker nodes as payment for the Basic blockchain services (see Section 3) or to the ZK and Oracle nodes involved in operating the relevant ZK and Oracle services (see Section 4 and 5).

To ensure that the Partisia Blockchain is always operational, the following special subset of governance contracts are free to interact with the Bakers nodes:

- The “BYOC deposit contract”.
- The “Baker reward distribution contract”.

Hereby users of Partisia Blockchain can always bring in more BYOC and avoid the blockchain to come to a halt in the extreme event where the blockchain is out of liquidity.

2.7 Exchange risk

BYOC brings in different coins with independent volatile valuations. The potential exchange risk for users and node operators built into this model is addressed in different ways as explained below.

The risk is primarily addressed through a fixed representation of the BYOC Twins on the Partisia Blockchain. This means that if a user adds ETHs to his BYOC account, the user’s account reflects ETH in the form of ETH BYOC Twins at all times. Hereby, when ETH BYOC Twins are distributed to the node operators at the end, everything adds up in terms of ETH, i.e. if the user spends 10 ETH the nodes receive 10 ETH.

As a multi currency system, however, the remaining challenge is to price the services on Partisia Blockchain across the different currencies supported by BYOC. To adjust for the varying purchasing power across the different BYOC Twins, the fees are calibrated relative to USD. This adjustment is done frequently based on market data collected by the dedicated price Oracles.

At the time where BYOC Twins are moved and locked by a smart contract the purchasing power at Partisia Blockchain for that particular BYOC coin is fixed and represented in “gas” units. Hereby, the user and the node operators know the fees in BYOC Twins. It follows from this that the exchange risk within the chosen BYOC Twin is removed while both parties face an exchange risk relative to USD. Since BYOC Twins locked into smart contracts are represented as gas units it means that gas units are pegged to USD. The fixed ratio is 100,000 gas units to 1 USD.

To give an example, consider a user with 10 ETH BYOC Twins in the account:

- At time then ETH was transferred to the user's account:
 - ETH BYOC twins in user's account: 10
 - ETH BYOC twins purchasing power is adjusted relative to USD
 - Cost of job: 1.1 ETH
- At the time of user starting a contract:
 - ETH BYOC twins in user's account: 10
 - ETH BYOC twins purchasing power is reduced due to a drop in ETH relative to USD
 - Cost of job: 1.2 ETH
- At the time of paying node operators:
 - ETH BYOC spent: 1.2
 - ETH BYOC transferred to node operators: 1.2 ETH

2.8 Staking as node operator

Staking is a core part of Partisia Blockchain and the only functional use of MPC tokens. As such, staking functions as a barrier for operating a node as well as a monetary guarantee or liability towards the users of Partisia Blockchain.

Node operators hold stakes on the staking contract and the stakes can be activated (transferred to others) through a dispute process. The general model for staking dispute is as follows:

- **Initiate dispute:**
 - A dispute is initiated by a challenger and the challenger can be anyone with an account on Partisia Blockchain.
 - The challenger pays the dispute contract.
 - Depending on the type of dispute it may require that the challenger deposit MPC tokens to initiate the process.
- **Lock stakes:**
 - Stakes get locked by the dispute component in the relevant contract.
- **Resolve dispute:**
 - Dispute resolution is orchestrated by the dispute contract.

- In case of objective fraud the dispute is solved through a decentralized governed process.
- In case of fraud without a clear objective resolution, the dispute is settled via voting among the MPC token holders and node operators.
- The result of the dispute is made public.
- Potential compensation in terms of MPC tokens is transferred with a suitable delay.
- **Compensation:**
 - The locked MPC tokens are distributed as compensation.

2.8.1 Staking as a Baker node

All whitelisted computation nodes are required to participate in Basic blockchain services.

A computation node is whitelisted for Baker node jobs if the staked value of MPC Tokens meets the threshold. The list of whitelisted Baker nodes is fully transparent and a Baker node will be removed from the list if; 1) the node doesn't meet the staking requirement, 2) if the whitelisting has not been renewed as a result of poor performance or 3) if the node voluntary revoke itself.

The minimum required stake for a Baker node is 25,000 MPC tokens.

The Partisia Blockchain detects objective fraud and unavailability such as participating more than five times in a consensus committee which fails to confirm blocks.

In case of objective fraud the stake will be locked and the node operator will be temporarily removed from the whitelist. Based on a decentralized decision and voting process the locked stakes may be burned. Examples of objective fraud and unavailability are:

- Failures to act as proposer.
- Double signing the same block.

The trust score will be designed to capture the objective fraud and unavailability.

2.8.2 Staking as a ZK node

A computation node is whitelisted for ZK node jobs if the staked value of MPC Tokens meet the threshold. The list of whitelisted ZK nodes is fully transparent and a ZK node will be removed from the list if; 1) the node doesn't meet the staking requirement, 2) if the whitelisting has not been renewed as a result of poor performance or 3) if the node voluntary revoke itself.

The minimum required stakes for a ZK node is 100,000 MPC tokens. In addition, the collective stakes for a particular job with n participating ZK nodes need to meet a user-defined insurance stake for a given time period. The insurance stake is defined by the user who pays an insurance premium to introduce additional stakes. The idea behind the insurance stake is that it is only the users that know the true value of the information involved in the ZK node job.

As an example, if the total required stakes for a given ZK node job is 10,000 MPC tokens and $n=5$, then each ZK node is required to have at least 2,000 MPC tokens in unlocked stakes (i.e. stakes that are not already locked for other jobs).

Only whitelisted ZK nodes that meet the required threshold for a given ZK node job can be randomly selected for the ZK node job.

The Partisia Blockchain detects objective fraud and unavailability such as participating in a ZK computation which fails to execute more than three times.

In case of objective fraud the stake will be locked and the node operators will be temporarily removed from the whitelist. Based on a decentralized decision and voting process the locked stakes may be burned or used for compensation.

The trust score will be designed to include objective fraud and unavailability.

2.8.3 Staking as Oracle node

A computation node is whitelisted for Oracle node jobs if the staked value of MPC Tokens meet the threshold. The list of whitelisted Oracle nodes is fully transparent and

a Oracle node will be removed from the list if; 1) the node doesn't meet the staking requirement, 2) if the whitelisting has not been renewed as a result of poor performance or 3) if the node voluntarily revoke itself.

The minimum required stakes for an Oracle node is 250,000 MPC tokens. Only whitelisted Oracle nodes that meet the required threshold for a given Oracle node job can be randomly selected for the Oracle node job.

The Partisia Blockchain detects objective fraud and unavailability such as:
Participating in an Oracle which fails to execute a transaction more than three times.
Unsanctioned transfer where funds have been transferred outside of the Oracle smart contract.

In case of objective fraud the stake will be locked and the node operators will be temporarily removed from the whitelist. In case of unsanctioned transfer, the stakes are used to compensate for the user's losses automatically as prescribed by the user's insurance stake. Otherwise based on a decentralized decision and voting process the locked stakes may be burned or used as compensation.

The trust score will be designed to capture the objective fraud and unavailability.

2.9 Staking and utility for all

The primary purpose of the MPC token is to function as an entry barrier to operating the primary Baker, ZK and Oracle nodes, which creates the decouple token economy and boosts interoperability through BYOC. Hereby the MPC token capitalises all of the values generated by the network and the core node operators. However, a number of additional usages for any holders of MPC tokens will be added as the network evolves. This section provides an overview of the currently planned opportunities for putting MPC tokens to work in the network economy.

2.9.1 Delegated community staking

Not all token holders are sufficiently experienced and eligible to operate a node or hold sufficient MPC tokens for staking. In order to activate these MPC tokens, delegated community staking is introduced.

This way multiple MPC token holders with a small number of MPC tokens can delegate MPC tokens to existing node operators and share the generated fees. The involved MPC token holders, however, will be jointly liable with respect to staking..

2.9.2 Mini nodes

A number of simple tasks that can be operated without a high quality node operator environment can be delegated to what we call Mini nodes.

The first task is to process so-called mirroring of the information across blockchains. This is a feature that ensures that two independent blockchains hold the same state, which is important for trustworthy interoperability. Section 5.1 describes how this is a core part of the model for token bridging.

The second task is preprocessing used to speed up zero-knowledge MPC computations. A new set of protocols enable this task to be delegated to an unlimited number of mini nodes. Operating a mini node producing preprocessing material can be done with low up time and connectivity and the mini node operator stakes MPC tokens to run the mini node and basically sell the preprocessing material to the ZK nodes.

The third task is storage of cryptography material among a very high number of nodes. With various built in thresholds, users can select the desired number of nodes involved and the required threshold of nodes required to access the stored information.

The fourth task is to construct ZK proofs used to compress information before it enters the blockchain. This may be used for so-called “ZK roll up” for selected services or data intensive dApps.

2.9.3 Smart contract staking

Staking MPC as an integrated part of a smart contract may add additional functionality or security.

The first task is to use MPC tokens as stakes to avoid a smart contract to be removed from the blockchain if it runs out of fees for storage as described in Section 2.6. A

privacy-preserving data driven economy on Partisia Blockchain will likely rely on a very high number of smaller smart contracts that confidentially activate the users' private off-chain data using zero-knowledge computation. This could be a simple matching of a user's profile in the advertisement market or a trading agent for automated trading. These smart contracts may not be used that often but they are important to keep on the blockchain.

The second task could be MPC tokens as collateral to mitigate security breaches in smart contracts. This would enable developers of smart contracts to enhance trust by committing MPC tokens. This idea is at a very early stage and the entire governance is unclear. Nevertheless, we want to share the idea with the community as it may foster other ideas.

2.9.4 MPC tokens as a freely traded asset

Finally, MPC tokens can be traded freely and may as such become part of the broader blockchain ecosystem. An example of added value is DeFi applications such as Automated Market Makers where providing liquidity adds utility to all.

2.10 Scoring

Partisia Blockchain will gradually turn into a *market for trust* where a node's reputation will become part of the selection and pricing process. A core part of this is the scoring of nodes that will be in place from the beginning and stepwise included into the protocols that make up the market for trust.

The scoring can be based on the following objective measures:

- Activities as Baker node:
 - Experience: # blocks as Baker node
 - Block proposer: # blocks created
 - Validator: # blocks signed
 - Reported irregularities
- Activities as ZK node:
 - Experience: # ZK jobs, Staked values
 - Active operations
 - Reported irregularities

- Activities as Oracle node:
 - Experience: # Oracle jobs, Staked values
 - Active operations
 - Reported irregularities

Other measures may be included later such as the public disclosure of information that follows from the incentivized P2P layer described in Section 3.3.

2.11 Software update

Updating the software that constitutes Partisia Blockchain is managed by the following decentralized governance process.

There are three types of software updates with increasing complexity:

- Type 1: Software features that do not impact neither states nor the ledger.
- Type 2: Software features that impact the integration with the blockchain i.e. the APIs.
- Type 3: Software features that do impact states of the ledger.

Both type 1 and 2 software updates can be asynchronously activated and deployed by the node operators without impacting the functionality of the blockchain. Type 3, however, may impact both the functionality and the economics of the basic blockchain and requires broad acceptance and deployment.

The process for Type 3 software updates are as follows:

- Node operators install the new version of the software without activating the new features.
- Node operators vote accept or reject.
- If more than $\frac{2}{3}$ of all node operators accept, the node operators activate and launch the software updates.

3 The Basic Blockchain Services

The Basic blockchain service is the core system to ensure proper communication among node operators, consensus as well as coordination of public information on

Partisia Blockchain. As such the Basic blockchain service functions as the bedrock for all services that are built on Partisia Blockchain.

3.1 Consensus and block creation

Partisia Blockchain applies a FastTrack committee-based consensus model with pure finalization.

The committee consists of the Baker nodes operating the Basic blockchain services. As mentioned earlier, there is only a single committee with no more than 1,000 whitelisted Baker nodes.

The Consensus model consist of the following three parts:

- Proof-of-Verification (PoV): Ensure that the signed blocks came from validators who actually executed all transactions included in the new blocks. In other words, a PoV is a guarantee that a validator personally verified the content of a block.
- FastTrack consensus: A rapid consensus process with pure finalization.
- Incentive scheme: Incentivizes propagation and a high degree of connectivity in the P2P network, described in Section 3.3.

Figure 5 illustrates the complete consensus model, which is further described in the following three subsections.



Figure 5: The complete consensus model

3.1.1 Proof-of-Verification

Proof-of-Verification (PoV) provides a guarantee that, when a verifier signs off on a block, it actually executed all the transactions included in the block. The purpose of this proof is to avoid a situation where a verifier simply signs a block when received from a proposer or a verifier who is believed to be honest. While this does not necessarily create a security issue, it has the effect of centralizing all the heavy lifting of the verification procedure on a small number of verifiers.

Our PoV ensures that, firstly, a valid proof requires the prover to hold the state of the blockchain. Secondly, the verifier likewise needs to hold the state of the blockchain. And third, the proof technique disincentivizes outsourcing the work required to produce a proof.

These three guarantees can be achieved by our PoV, which is defined as a personalized hash of the updated state. More in detail, we hash the current top of the state tree together with a personalized identifier. In doing so, we ensure that both verifier and prover must have access to the state. Lastly, because the proof is personalized, outsourcing the production of a proof comes at a great risk. To meaningfully outsource

proof creation, the outsourcer cannot check that the returned proof is valid (indeed, proof verification is the same as proof creation, so the prover may just as well generate the proof itself in that case). Since the outsourcer cannot check that the PoV is actually valid, then they run the risk of being burned. If they unknowingly pass on a wrong proof (because they cannot see that it is wrong), then it is the node as outsourcer (and not the entity who actually created the proof) that gets into trouble.

3.1.2 FastTrack consensus

The FastTrack consensus achieves consensus among at least $\frac{2}{3}$ of all Baker nodes through an optimistic protocol with an ability to recover if it fails. When running optimistically, a single block proposer (the proposer) is selected to continuously create new blocks, and all other nodes in the committee (the validators) verify that the blocks are valid and sign them. The protocol defines a fixed number of blocks to be created by the proposer. A new proposer is selected when the fixed number of blocks has been created or in case of “shutdown” as explained below.

Partisia Blockchain is based on transaction costs in the sense that almost all transactions are connected to fees paid by the users of the blockchain. To ensure rapid execution, transactions are executed via immediate block creation, this is what we call Eager FastTrack consensus. Therefore, as long as the proposer is honest and active, the chain will produce blocks on demand as transactions appear. However, to ensure availability an automatic empty “heartbeat block” is executed in case no transactions have been received in 30 seconds.

Consensus and finality is established by Proof-of-Justification (PoJ) and Proof-of-Finalization (PoF) respectively:

- PoJ is a list of $\frac{2}{3}$ or more signatures confirming validation of the current block held by a single node.
- PoF proves that the previous block is finalized and it is established indirectly by any PoJ made on the current block. Hence, the previous block is finalized when $\frac{2}{3}$ or more nodes have provided a PoJ for the current block.

A block is considered final when $\frac{2}{3}$ of all nodes or more have registered $\frac{2}{3}$ of all nodes or more signing the block. Note that PoF is not flooded through the P2P network, it exists, however, in a decentralized form, meaning that it is revealed indirectly by any PoJ conducted on the following block. This decentralized nature of finalization ensures rapid consensus even without the network being synchronous.

If new transactions have not been processed within 30 seconds, the system assumes that the proposer stopped producing valid blocks (or was unable to get the block to the validators due to network issues) and the recovery mechanism - i.e. Shutdown procedure - is initiated. The purpose of the Shutdown procedure is to ensure that all committee members agree about which block is the latest block that has been finalized. In the view of a single node, the most recent block validated with a PoJ may be cancelled in rare cases i.e. it is possible to roll back at most one block. To ensure that all nodes are able to get the information about which block the committee agreed upon, a special deterministic block (shutdown block) is produced and appended to the ledger. After the shutdown block has been appended the committee selects a new proposer to be responsible for creating blocks and restarts in optimistic mode. The Shutdown procedure is based on a semi-synchronous Byzantine Fault Tolerance (BFT) style agreement protocol. See Figure 6 for an illustration of the FastTrack consensus.

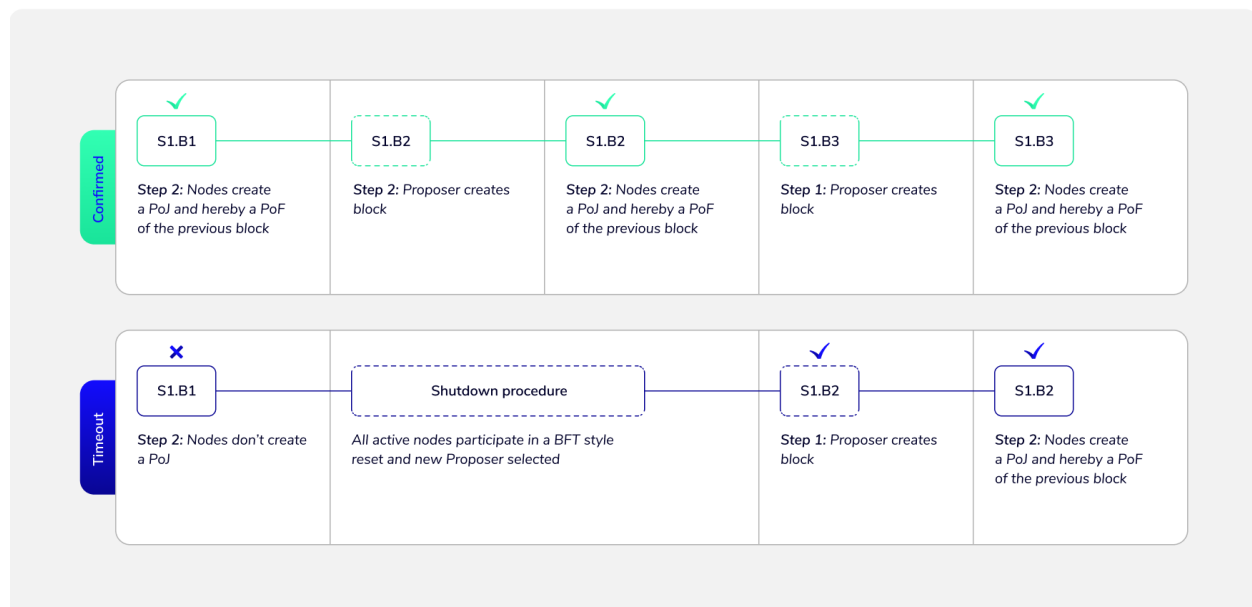


Figure 6: The FastTrack consensus process

To allow nodes to join and leave the committee it is necessary to have a procedure for changing the committee. This is handled by a smart contract that collects signatures from the old committee signing off on the new committee. Only when 80% of the old committee has signed off on the committee (proof of committee change) will we be allowed to do the transition. The transition is done as an event on the blockchain where the new committee is stored in the state of the blockchain. As soon as the current committee sees that a new committee should be activated, they will trigger the Shutdown procedure to ensure that the current committee does indeed agree on the current block. When a shutdown block has been produced (and the committee change

event has not been rolled back) the old committee is no longer allowed to produce blocks and the new committee takes over.

To further expedite the transaction speed, the execution of transactions is initiated immediately after a block has been either proposed (by the proposer) or validated (by the validators). Figure 7 illustrates how the flow of transactions queues up for the next block and how transactions are executed immediately in parallel to the FastTrack consensus. The finalization, however, depends on the completion of the PoF for the block in question.



Figure 7: Block Production and execution of transactions.

Altogether, the Partisia Blockchain consensus model supports fast transaction speed in multiple ways:

- Sharding (presented in Section 3.2) and eager block production reduces the time from when a transaction is created by the user until a block proposer adds the transaction to a block.
- The immediate execution of transactions improves the transaction speed.

- The FastTrack consensus ensures rapid justification (PoJ) and finalization (PoF) of the transactions.

3.2 Sharding

The Partisia Blockchain uses sharding to ensure scalability of the Basic blockchain services and transaction speed. Sharding distributes the workload to S parallel shards. To avoid interference with the consensus model all Baker nodes are required to operate all parallel shards as long as the total number of Baker nodes is less than 1,000. However, since the consensus model (see section 3.1) is shard-specific each shard is independent, which allows the workload of operating shards to be distributed across independent servers as well as separate groups of Baker nodes as the network grows. To keep track of committees and to ensure that a new committee is operational across shards, the proof of committee change is distributed across all shards.

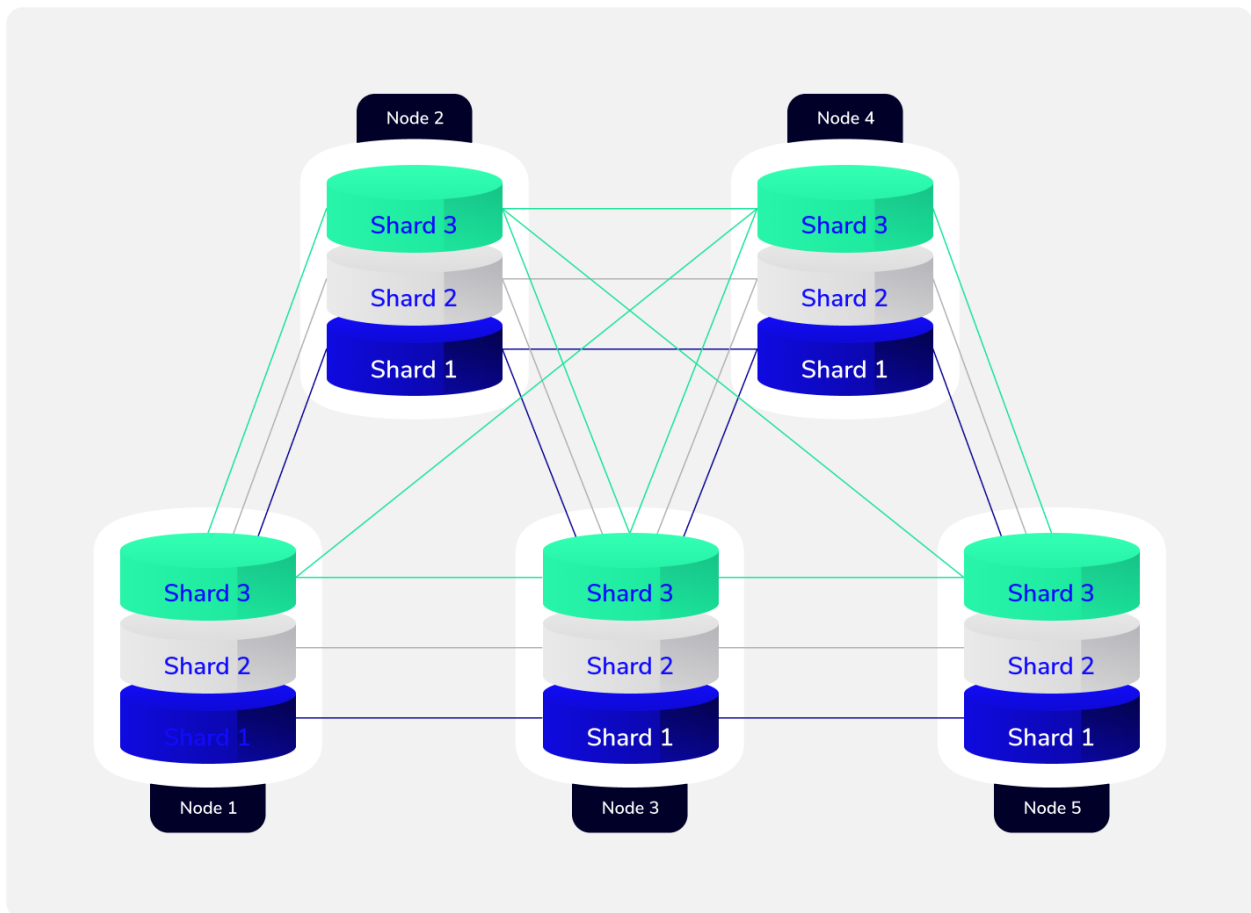


Figure 8: The parallel shards.

The scalable block creation is illustrated in Figure 8 and 9 where each shard creates verifiable blocks according to the consensus model presented in Section 3.1. The consolidated blockchain shows how the sharding scales linearly in the number of shards i.e. a blockchain running with three shards can handle three times as many concurrent transactions as a blockchain running without sharding. To see this, note that the consensus process including the time consuming propagation of messages is done in parallel and independently across shards.

To allow the system to keep track of the blocks, the following number of temporal measures are used to create a consolidated view of the blockchain:

- Block production time: Current human-readable timestamp is set by the block producer and guaranteed to increase relative to the previous block on the same shard.
- Shard block time: Counts number of blocks since genesis on a given shard.
- Consolidated block time: Shard number, shard specific block time and block production timestamp e.g. S1.B1.T1 and S3.B1.T2 in Figure 9.

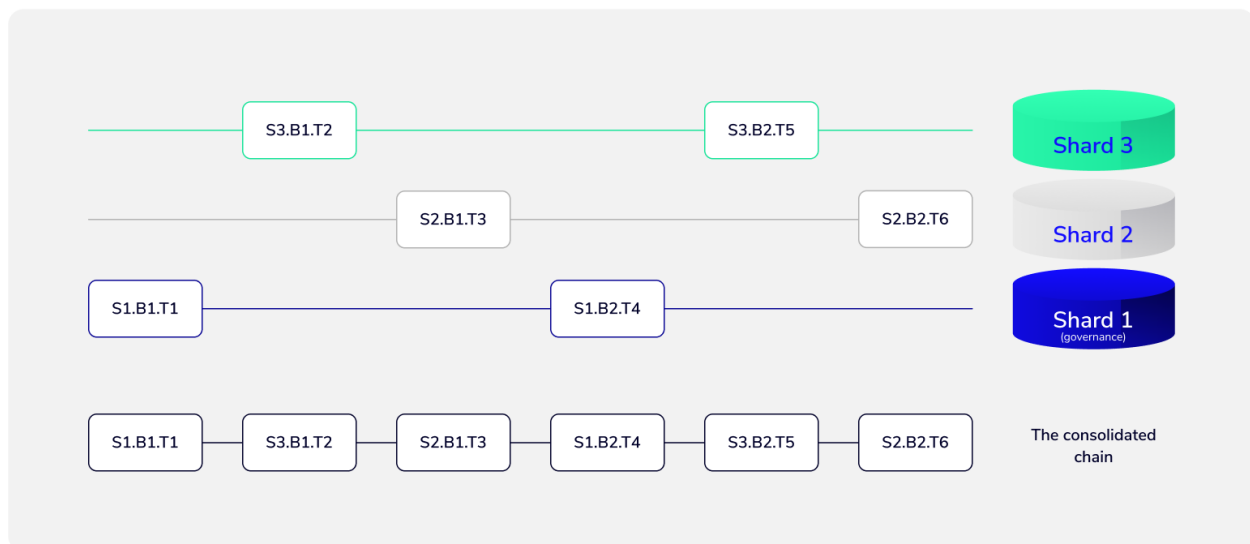


Figure 9: Block creation across shards

All shards are the same except for the “Governance shard” chosen by the genesis block (see Figure 9). The Governance shard is similar to the other shards except for the

dedicated role as manager of tasks that relate to the governance of the Partisia Blockchain, which include:

- Changes to the committee such as:
 - Adding new whitelisted baker nodes
 - Excluding nodes
 - Selecting a new committee if there are more than 1,000 baker nodes
- Implementation of voting polls such as:
 - Exclusion or re-entry of nodes
 - Updates to governance
- Configuration across shards
 - Current active shards and routing
 - Current fee structure

Note that the various decision processes are managed by smart contracts but implemented via the Governance shard. Also note that a given state exists on one shard and only one shard.

3.3 The payment scheme for basic blockchain services

Although Proof-of-Work based blockchains have too many drawbacks, one interesting property is that miners have strong incentives to validate transactions. To see this, note that in Proof-of-Work it is costly to mine on a copy of the ledger that turns out to not be the longest chain, as the reward will have no value. This provides miners with proper incentives to verify the transactions and create the longest chain before investing resources into mining. In committee-based consensus, voting for a block by signing transactions does not come with the same incentives to do a proper verification job per se.

Partisia Blockchain compensates for this throughout the entire consensus process. On the one side, PoV ensures a higher degree of actual validation, on the other side, the FastTrack consensus protocol with PoJ, PoF and the built-in shutdown procedure ensures proper consensus. However, without a well connected P2P network of active nodes that rapidly propagate messages, the blockchain may slow down. To compensate for this we have introduced incentive provision at the P2P layer that addresses these goals.

The incentive scheme distributes the total revenue generated from the Basic blockchain services (R). This is done by incentivizing the flooding of signed validations at the P2P layer or more precisely propagation of signatures as well as connectivity of the individual node.

The incentive scheme relies on publicly reported information sent from the nodes to the reward smart contract, which computes and distributes R among the Baker nodes following the proposed mechanism design. The collected information consists of aggregated metadata about signatures received in the consensus process and not the actual signatures or cryptographic proofs of any sort. Hereby the incentive provision is done ex post after a number of blocks have been produced and finalized, without slowing down the rapid consensus and execution of transactions.

It follows from this design choice that the smart contract (the reward contract) that computes and distributes the reward holds no verifiable information about what the nodes are signing and propagating throughout the P2P network. Locally, however, the nodes know what is happening in the network, information which is used to incentivize local decisions through public reporting and rewards to other peer nodes. Figure 8 illustrates how the incentive scheme requires the nodes to report locally held information about the number of signatures received from other nodes and how this is used to distribute the total reward R . The incentive scheme ensures that those nodes that contributed most to reach consensus are rewarded most.

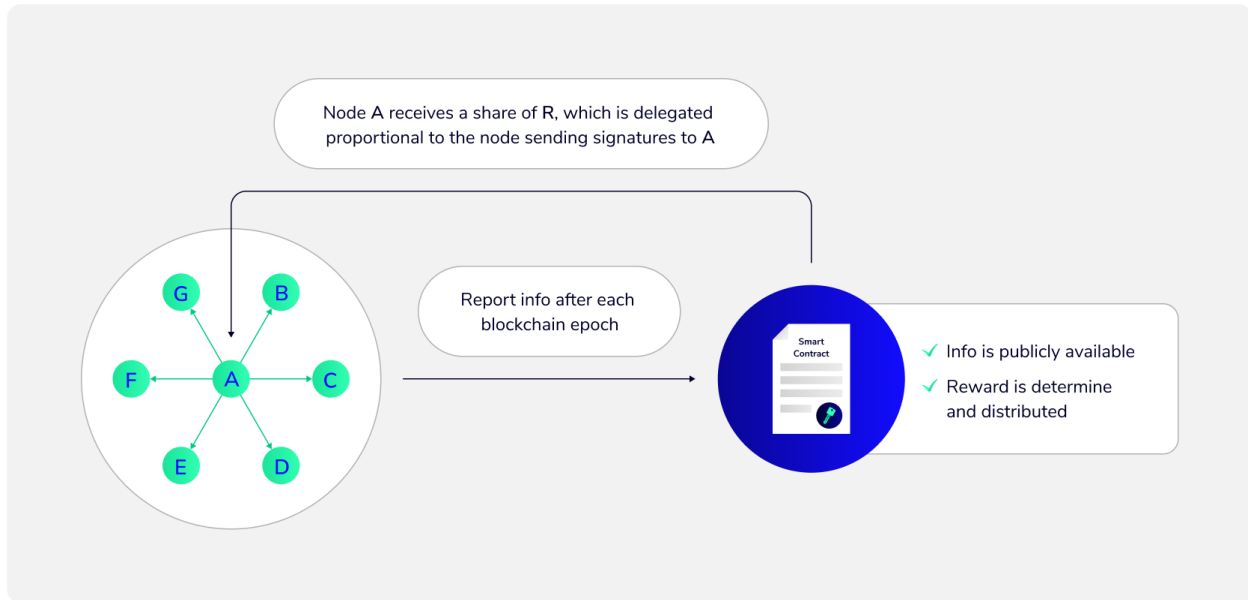


Figure 10: An illustration of the reward mechanism

The basic incentive scheme illustrated in Figure 10 has the following four steps:

- Step 1: Public report of information - each node provides a list with the number of unique signatures received from other nodes.
- Step 2: Eligibility - based on the publicly reported information, a node is eligible if its signature has been received by more than 50% of all nodes.
- Step 3: Divide R equally among eligible nodes.
- Step 4: Distribute the eligible node's share of R (the share granted in Step 3) proportional to the signatures received by peer nodes and publicly reported by the eligible node.

To further address the incentives we add a “challenge” that randomly selects a single Baker node (the Challenger) to only circulate others' signatures and not its own signature. The selected Challenger automatically challenges other nodes that publicly report that they received or sent the Challenger's signature. If someone reports the Challenger's signature and cannot provide proof (the Challenger's signature), then the exposed node loses stakes to the Challenger.

A node may deviate from truthful public reporting and claim that they sent more or less signatures. First, note that misreporting has no direct effect on node j 's reward. However, since misreporting becomes transparent for all, it will cause the other nodes

that actually sent signatures to node j to react if node j misreports the number of signatures received as that directly impacts the other nodes' payoff.

The intuition behind truth-telling as the optimal strategy is as follows:

1. Consider the case where node j reports less signatures. Here the primary driver is that the nodes related to the signatures that are not reported get less reward and may punish j by not sending signatures in future rounds. This repeated tit for tat game incentivizes nodes to not report less signatures.
2. Consider the case where node j reports more signatures: Here the primary driver is the cost of the "challenge" that makes it costly to report more signatures. This built-in punishment incentivizes nodes to not report more signatures.

In addition, note that eligibility is constructed by information from multiple nodes, which require manipulation to be coordinated across more nodes.

To save communication and blockchain transactions, the public signal provided by the individual nodes is done locally and the incentive payments are aggregated across blocks and executed after each Basic blockchain epoch.

In conclusion, the incentive scheme promotes truth-telling and the best-connected nodes that propagate the most signatures receive the most reward.

4 Zero-knowledge computation

Partisia blockchain leverages privacy-preserving computation through a decentralized Zero-Knowledge (ZK) computation.

ZK computation is a type of computation which preserves certain desirable privacy properties with respect to the involved parties. Put briefly, a ZK computation is a certain kind of distributed computation in which clients provide inputs to some computing nodes who then perform a computation on these inputs. What makes ZK computation special and desirable, is that this outsourcing happens without revealing

anything at all about what the clients actually sent as their input (unless all servers collude).

This section provides a first introduction to ZK computation and how it is orchestrated on Partisia Blockchain. A more detailed description is provided in the Partisia Blockchain yellow paper on MPC-as-a-Service.

Throughout this section we will use ZK computation and MPC interchangeably.

4.1 Introduction to private smart contracts

The selected ZK Nodes execute ZK computations and have the following two roles:

- Pre-processing - Nodes which perform pre-processing are tasked with creating certain cryptographic material that will later be used by the computing nodes to speed up computation.
- ZK computing - Nodes that execute the actual secure computation, usually utilizing cryptographic material generated by pre-processing nodes, are called computing nodes.

Executing a ZK computation requires significant orchestration and care, compared to a public computation on clear text information. In a nutshell, a ZK computation takes place between a number of ZK nodes, utilizing pre-processed material from a (potentially different) group of ZK pre-processing nodes. Selection of these nodes have to meet certain requirements defined by the service owner (the entity buying ZK computation). The security model of MPC guarantees that privacy is preserved, provided that no more than t nodes collude. The service owner therefore specifies a threshold, t , selected such that collusion between no more than t of the nodes can reasonably be assumed to hold. A number of mechanisms will be introduced to counteract collusive behavior such as additional staking and a market that allows the users of the Partisia Blockchain to select ZK nodes.

4.2 MPC orchestration

To perform a ZK computation, a number of ZK computation and pre-processing nodes have to be selected, who will then be tasked with executing the MPC. Parameters for

the selection are specified by a service buyer, but the orchestration itself is handled by the Partisia Blockchain.

All ZK nodes, whether computation or pre-processing, are grouped together in a pool. Each node is vetted, known, and carries information about where in the world it is located, as well as its score (a notion that will be discussed in a moment). A buyer specifies a set of parameters, such as the threshold t , the type of protocol and computation to be run, and the Partisia Blockchain runs a computation that selects an appropriate set of nodes at random from the pool that will be tasked to execute the ZK computation. This “buy event” is posted on the blockchain which allows the selected computation to be run. Figure 11, illustrates the complete orchestration of ZK computations or MPC-as-a-Service.

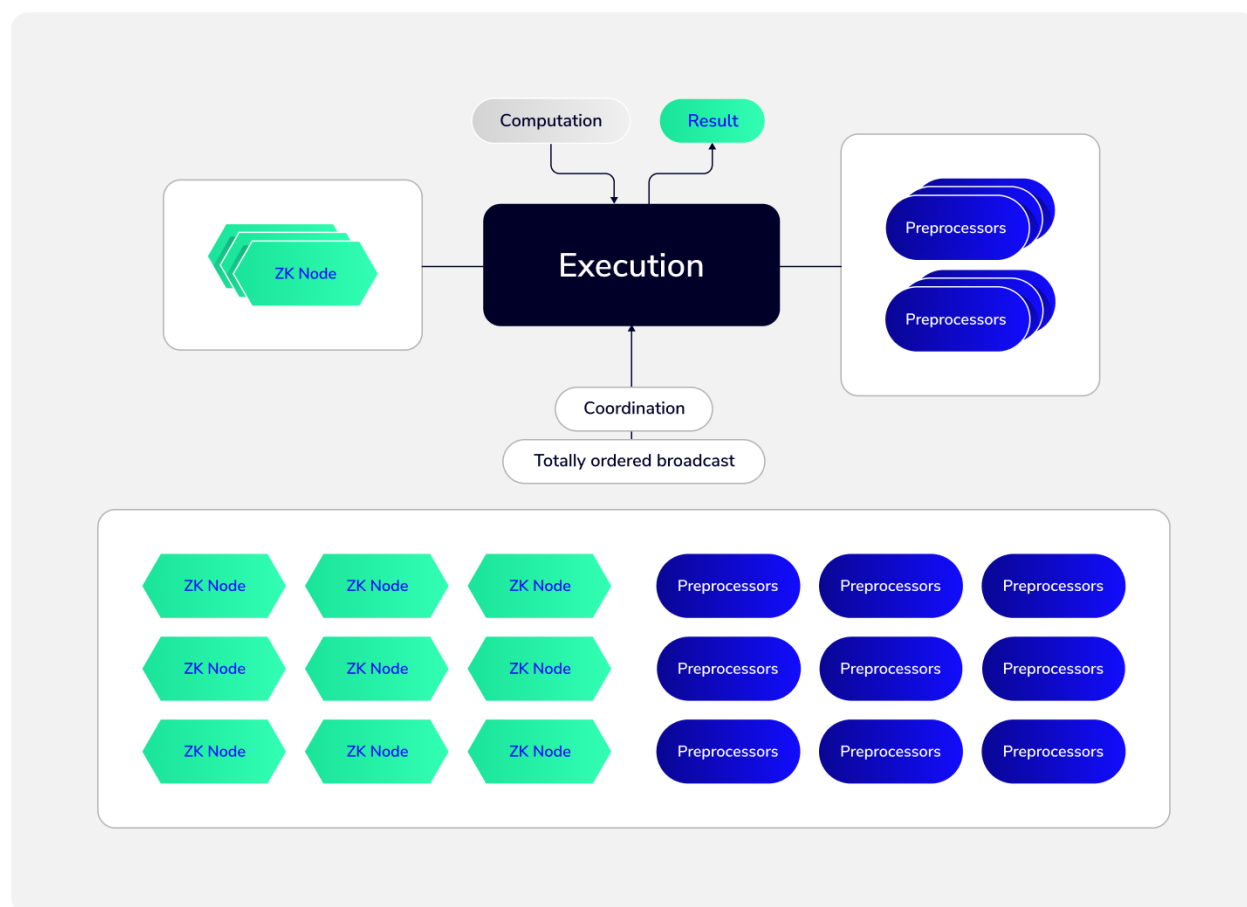


Figure 11: MPC-as-a-Service orchestration

4.3 Pre-processing

Before most ZK computations can begin, some cryptographic material has to be pre-processed. The amount and type of material is entirely dependent on the computation. Roughly speaking, the more complex the computation is, the more pre-processing material the computation will require.

Generating pre-processing material is carried out by pre-processing nodes. These nodes are selected similarly to the ZK nodes and produce pre-processing material for general purpose or directly for a given ZK computation. General purpose pre-processing material can be used by all and allows users to execute fast online ZK computation with no delay. For ZK computations planned ahead of time, tailored pre-processing is produced once the ZK computation has been scheduled.

4.4 Payment schemes

ZK computation and pre-processing nodes are rewarded for participating in secure computations. The payment received by each node will depend on a number of factors, such as complexity of the computation (measured e.g., as a function of the number of multiplications that a computation consists of) and the score of each node.

Payment to the participating ZK nodes is performed by both the users or the entity which originally specified a particular computation.

4.5 Scoring

Partisia Blockchain includes a scoring mechanism for all ZK nodes. The scoring of a node depends on a number of different factors, such as how many non-faulty computations a node has participated in etc.

4.6 Staking

As a deterrent, ZK pre-processing and computation nodes must stake MPC tokens. Mechanisms will ensure that, should a node be caught trying to cheat, then its stake can be taken.

The stakes function as an extra assurance of the confidential values managed by the ZK nodes. It is, however, only the services and/or the users of the service running on top of Partisia Blockchain that truly know the value of the confidential information involved. To accommodate for this asymmetric information problem the users can adjust stakes to meet an appropriate level of insurance.

In case of objective fraud the stake will be locked and the node operators will be temporarily removed from the whitelist. The locked stakes will be used to compensate users based on a decentralized decision and voting process.

5 Oracle services

This section describes the core Oracle services that provide interoperability which allow values and data to be moved across different blockchains and other platforms.

Oracle services cover integrated services like BYOC and Token Bridges operated by a selected subset of whitelisted Oracle nodes and yet safeguarded by the blockchain as a whole.

5.1 Cross-chain security models

Interoperability and token bridges as the most prominent instrument, is one of the main value creators in the blockchain ecosystem. Designing secure token bridges is, however, also a significant challenge as values are moved out of the security model backing the involved tokens. Partisia Blockchain has designed a new type of token bridge that addresses a number of known weaknesses in existing token bridges and provides a new best practice.

As for other parts of the Partisia Blockchain, the actual level of security is provided through a mix of economic and cryptographic mechanisms. Here we describe the three main security measures and principles that make up the basic cross-chain security model illustrated in Figure 12 below. Section 5.2 and 5.3 describe how this model operates BYOC and Token Bridges in more detail.

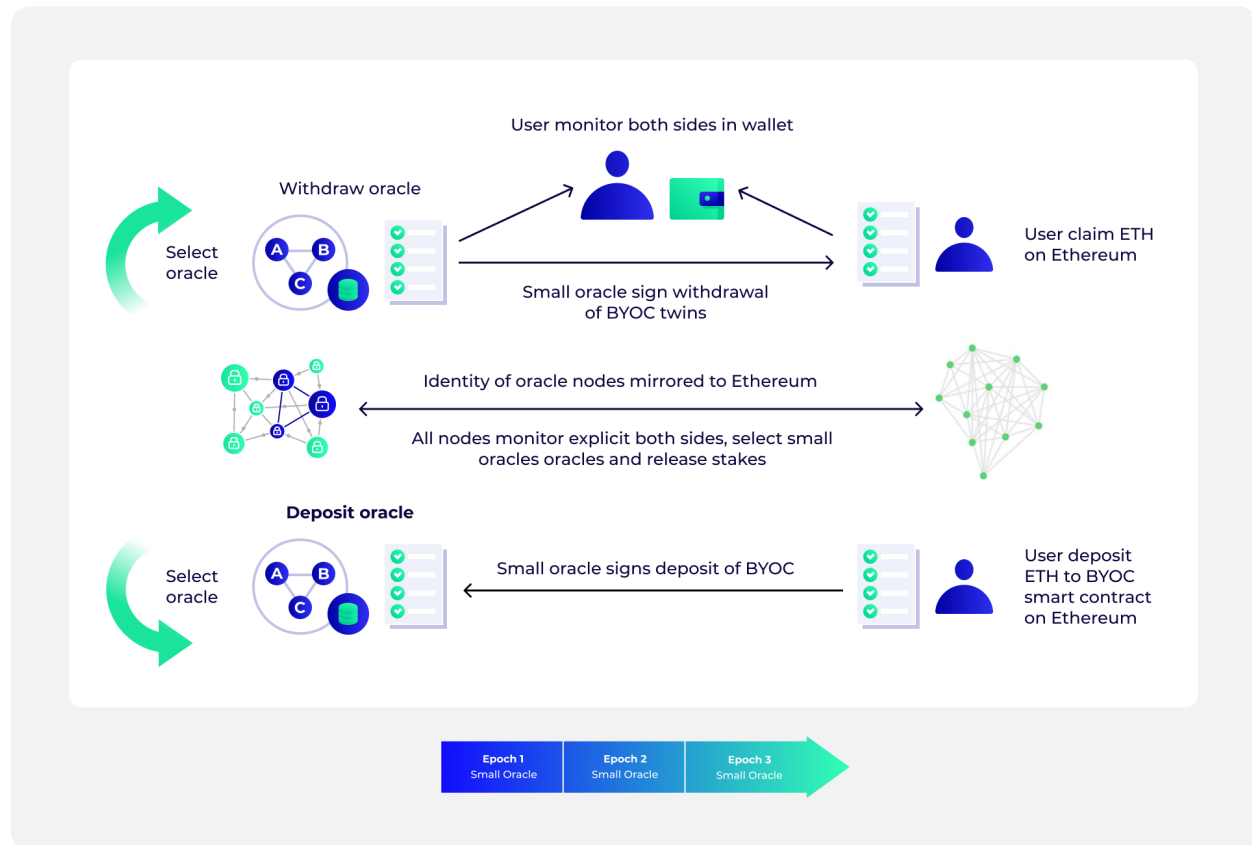


Figure 12: The basic cross-chain security model

The core basic principle is that a token transfer is a 1 to 1 representation across two independent blockchain networks, which basically mimics a double booking system that has proven its worth since the Medici Bank in the 14th century. The challenge is, however, to simultaneously represent and work with states from two independent blockchains. On Partisia Blockchain, this representation of values across the involved blockchains is managed by a randomly selected Oracle that collectively stakes the values transferred within a given epoch. Finally, the Oracle is guarded by all nodes involved in the consensus model backing the entire network using large scale MPC based cryptographic key management. Altogether this can be separated into and described in three parts as follows:

Part 1 - double bookkeeping within regularly expiring epochs: As illustrated in Figure 12, the bridging process consists of a deposit and a withdrawal set of smart contracts that constitute the double bookkeeping system. The process ensures that the information is persisted on both blockchains and establishes a straightforward

auditability that is easily accessible by all users and node operators.

Deposit and withdrawal is managed by a small set of independently selected Oracle nodes - the small Oracle - in frequently expiring epochs as further described below.

Part 2 - collateralized bridging within regularly expiring epochs: The small Oracle consists of Oracle nodes with sufficiently available MPC tokens that function as locked collateral during an epoch. A price Oracle captures in USD the value of MPC tokens and the asset that is bridged to Partisia Blockchain. Now for every value transferred a similar value in MPC tokens is added as collateral. When the small Oracle runs out of MPC tokens, the epoch expires. In the following epoch any holder of MPC tokens can run a dispute in case of fraud as further described in Section 5.2.

Part 3 - selecting a new small Oracle and sealing epochs: A selection process with the highest security offered by Partisia Blockchain further strengthens the security provided by part 1 and 2 above. An MPC based signature scheme operated by the large Oracle (i.e. all available baker nodes) is used to ensure that the selection of the next small Oracle matches the security of the consensus model. The large Oracle is also responsible for delegating the job of mirroring the identity of the members of both the small and the large Oracle to the partner chain. With that in place, the double bookkeeping is securely established. The final task of the large Oracle is to release the stakes after the dispute period.

The large Oracle applies an optimized MPC protocol that securely computes threshold signatures. The large Oracle is operated by up to 1,000 nodes and hereby providing the same level of threshold security as offered by the consensus model described in Section 3.

In conclusion, the members of the small Oracle allows for efficient collateralized bridging, while the large Oracle provides the highest level of security offered by the Partisia Blockchain. The process is described in more detail in the context of BYOC and Token Bridges in Section 5.1 and 5.2.

5.2 Bring Your Own Coins (BYOC)

Bring Your Own Coins (BYOC) allows users to bring in external coins or tokens as means of payment on the network and ensure that node operators are rewarded in external coins or tokens. This is a core functionality that underpin the position of Partisia Blockchain as a network that provides privacy and interoperability across blockchain networks while promoting other blockchains tokens or coins.

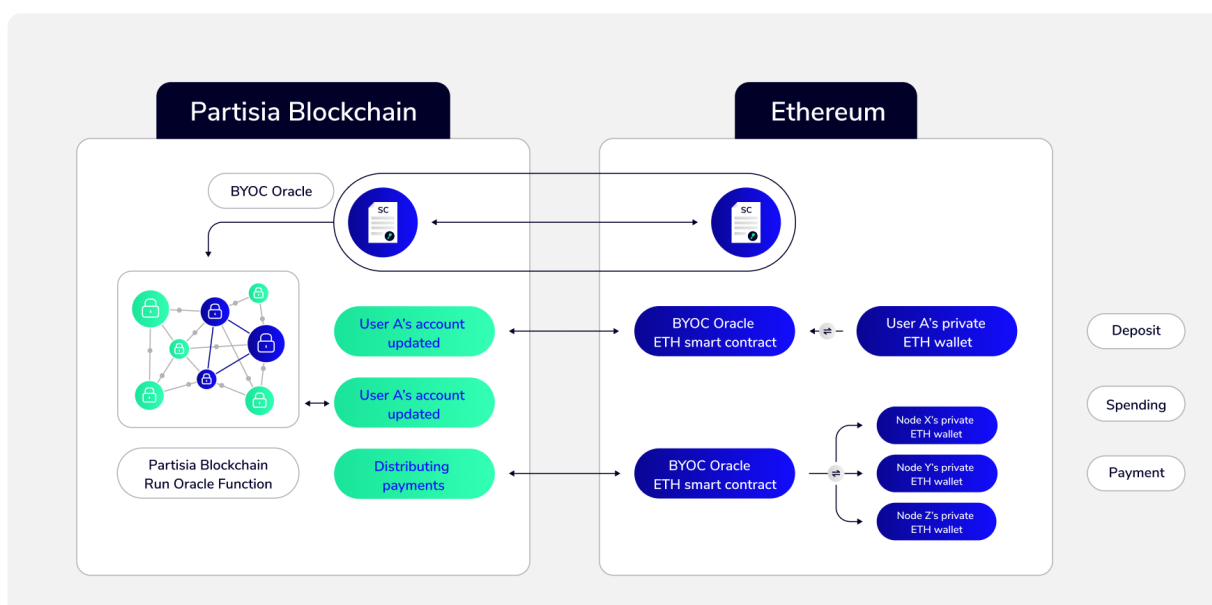


Figure 13: The BYOC Oracle setup

Figure 13 illustrates the basic setup where the BYOC smart contract on Partisia Blockchain orchestrates one or more smart contracts on the integrated blockchain - Ethereum in this illustration.

The basic security model involves a unique combination of a double bookkeeping principle, collateralized bridging and arms length selection of Oracle nodes to run BYOC as described in Section 5.2 above. Here we describe how this security model works at a transaction level for BYOC.

Every transaction is approved by a majority of the small Oracle. Hereby, all potential collusion will involve at least a majority of the small Oracle per definition. The staking and dispute mechanism described in Section 5.2.1 utilizes this structure to ensure that

the staked MPC Tokens function as financial guarantee towards the users of the BYOC Oracle.

The process below show how BYOC operate over epochs with Ethereum as example:

Prior to an epoch:

- A small Oracle with n Oracle nodes is available and a monetary threshold, T , provides a maximum of values that the small Oracle can manage in this epoch. T is defined by the collective minimum stakes provided by a majority of the n nodes plus the cost of running one dispute times the number of expected transactions (estimated by a minimum transaction size).
- Initially all Oracle nodes are required to stake the same minimum stake.

During an epoch:

- User deposit and convert ETH to ETH BYOC Twins:
 - Step 1: A user transfers ETH to the BYOC smart contract
 - Step 2: The transfer activates a pull verification process that matches ETH and ETH BYOC Twins across the two smart contracts on Ethereum and Partisia Blockchain. The transaction is signed by at least the majority of the small Oracle and registered as a transaction on Partisia Blockchain.
 - Step 3: The Partisia Blockchain creates ETH BYOC Twins in the user's account.
 - Hence, double bookkeeping has been established for deposit.
- User spend ETH BYOC Twins on Partisia Blockchain:
 - Step 1: The user spends ETH BYOC Twins by signing transactions on the Partisia Blockchain, the BYOC Twins are transferred to the relevant smart contracts and hence registered as pending payment on Partisia Blockchain.
 - Step 2: The smart contract spends the ETH BYOC Twins and registers the pending payment as executed.
 - Step 3: Every fixed time period (e.g. daily or hourly) all pending payments are resolved and the number of ETH BYOC Twins corresponding to the registered costs, is allocated to the nodes following the payment scheme.
 - Hence, BYOC Twins have been used and exchanged.
- User withdraw and convert ETH BYOC Twins to ETH:
 - Step 1: A user transfers ETH BYOC Twins to the BYOC smart contract.

- Step 2: The transfer activates a push verification process that matches ETH BYOC Twins and ETH across the two smart contracts on Partisia Blockchain and Ethereum. The transaction is signed by at least the majority of the small Oracle and registered as a transaction on Partisia Blockchain.
- Step 3: The user uses a so-called nonce proof to withdraw ETH on the Ethereum side. The nonce proof verifies the transaction.
- Prepare the next epoch
 - The smart contract selects a new small Oracle as described in Section 5.1.
- Epoch ending:
 - When the value managed by the small Oracle reaches the threshold T , the epoch ends. Consequently, T defines the length of the epoch.

After an epoch:

- The next small Oracle takes over the BYOC immediately after the previous small Oracle and resumes production as described above.
- The large Oracle ensures that the information about the transactions (states) are mirrored on Ethereum.
- The users use a so-called merkle tree proof to withdraw ETH on the Ethereum side for any closed epochs. The merkle tree proof verifies the transaction.

During the subsequent epoch, anyone with an account on Partisia Blockchain can stake MPC tokens and initiate a dispute. The BYOC Oracle is designed to be fully operational independent of ongoing disputes.

5.2.1 Staking

As mentioned above, the nodes operating the small Oracle stake MPC tokens as collateral. The trust measure is very tangible as stakes are used to compensate users in case of objective fraud, as further described in this subsection.

The solution to avoid accumulation of risk is to contain the risk within an epoch. To see that, note that the small Oracle is fully reliable for the following two reasons:

1. The users of BYOC can not cheat without involvement and approval from a majority of the nodes operating the small Oracle.

2. At the end of an epoch it is straightforward to verify any imbalances between the BYOC Twin on the Partisia Blockchain and the original coins on the integrated blockchain such as Ethereum, i.e. the double bookkeeping principle.

There are basically two types of fraud:

- Type 1: Partisia Blockchain knows who has been affected by the fraud and the missing BYOC values are within control of the Partisia Blockchain.
- Type 2: Partisia Blockchain does not know with certainty who has been affected by the fraud and the missing BYOC values are not within control of the Partisia Blockchain.

To exemplify the two types of fraud consider the following examples with an ETH based BYOC:

Examples of type 1 fraud:

- A user transfers 10 ETH into the BYOC contract on Ethereum. The small Oracle notifies Partisia Blockchain about the transfer, but indicates a transfer of 1 ETH. The 1 ETH is minted to the user as BYOC Twins and the user is missing 9 ETH BYOC Twins.
 - Solution: This is fully transparent and after a successful dispute the remaining 9 ETH BYOC Twins are minted and the members of the small Oracle signing the transaction are penalized.
- A user withdraws 10 ETH through the BYOC contract on Partisia Blockchain. The small Oracle only issues a transaction allowing to get 5 ETH from the Ethereum BYOC contract.
 - Solution: This is fully transparent and after a successful dispute the missing 5 ETH is made available to the user on Ethereum and the members of the small Oracle signing the transaction are penalized.

Examples of type 2 fraud:

- Without any user interaction, the small Oracle issues a withdrawal transaction for 10 ETH to be used on the Ethereum BYOC contract, which manages the common pool of ETH involved in the ETH version of BYOC.

- Solution: This is fully transparent and after a successful dispute, the stakes committed by the members of the small Oracle are used to reestablish the balance as further discussed below.
- A user transfers 10 ETH into the BYOC contract on Ethereum. The small Oracle notifies Partisia Blockchain about the transfer, but indicates a transfer of 20 ETH. The 20 ETH is minted to the user as BYOC Twins. Consequently, the BYOC Twins are not entirely backed by ETH.
 - Solution: This is fully transparent and after a successful dispute, the stakes committed by the members of the small Oracle are used to reestablish the balance as further discussed below.

The most critical fraud is type 2 fraud, which is where the stakes play a significant role. The primary solution is where Partisia Blockchain converts the confiscated staked MPC tokens to ETH on the open market. The secondary solution is a voluntary internal market where the users of Partisia Blockchain exchange BYOC or ETH for MPC tokens. Unlike the secondary solution, the primary solution requires a liquid market for exchanging MPC tokens and ETH, which will be established over time.

We are now ready to describe the entire staking dispute process:

- **Initiating the dispute:**
 - A dispute involves a single transaction and there can only be one dispute per transaction.
 - A dispute is initiated by a challenger and the challenger can be anyone with an account on Partisia Blockchain. In particular those affected by potential fraud have strong incentives to challenge, not least the next small Oracle that would otherwise inherit potential fraud.
 - The challenger pays the dispute contract. The payment is refunded to the challenger by the end of the dispute process, unless no imbalance exists.
 - The challenger deposits an amount of MPC tokens to initiate the process. The deposit is returned to the challenger by the end of the dispute process, unless no imbalance exists in which case the deposit is burned.
- **Lock stakes:**
 - The stakes assigned to the relevant BYOC in the relevant epoch are locked throughout the lifetime of a small Oracle.

- **Resolve dispute:**
 - The dispute resolution is orchestrated by the dispute smart contract.
 - Type 1 fraud is solved directly by a decentralized governed process.
 - The affected users and the compensation are registered.
 - A penalty to the small Oracle is registered.
 - Type 2 fraud initiates an exchange of MPC token to the token behind the BYOC Twin e.g. ETH.
 - The primary and secondary market is initiated.
 - Although disputes are run per transaction, the exchange market allows for bundling disputes and thereby increasing the market.
 - The result of the dispute is made public.
- **Compensation:**
 - Type 1 fraud:
 - The result of the dispute is executed directly
 - The balance is reestablished.
 - The penalty to the small Oracle is shared equally among the members of the small Oracle parties signing the transaction and burned.
 - Type 2 fraud:
 - The two exchange markets are solved.
 - The total amount of stakes exchanged is shared equally among the members of the small Oracle parties signing the transaction.
 - The imbalance is reestablished.

Note that a fraud dispute is isolated to a single transaction and a single dispute will either involve type 1, type 2 or both types of fraud.

Apart from the above staking and dispute mechanism, a small number of stakes is burned in case of inactivity.

5.3 Token Bridge

A Token Bridge allows for seamless transfer of values (tokens) across two independent tokens and blockchains. Similar to BYOC, accumulation of risk is addressed by epochs and a small Oracle that manages the risk through stakes as collaterals during an

epoch. The design is essentially a combination of two BYOC Oracle. Figure 14 illustrates the design of the Token Bridge.



Figure 14: The Token Bridge

Consider a Token Bridge between EOS and Ethereum, which locks EOS tokens on the EOS blockchain while creating a ERC-20 version of the EOS token (ERC-20-EOS) on the Ethereum blockchain. The Token Bridge safeguard this process and operate across epochs as follows:

Prior to an epoch:

- A small Oracle with n Oracle nodes is available and a monetary threshold, T , provides a maximum of values that the small Oracle can manage in this epoch. T is defined by the collective minimum stakes provided by a majority of the n

nodes plus the cost of running one dispute times the number of expected transactions (estimated by a minimum transaction size).

- Initially all Oracle nodes are required to stake the same minimum stake.
- The small Oracle deploys two smart contracts on EOS and Ethereum.

During an epoch:

- User transfers EOS to ETH:
 - Step 1: A user transfers EOS to the bridge smart contract.
 - Step 2: The transfer activates a pull/push verification process that matches EOS and ERC-20-EOS across the two smart contracts on EOS and Ethereum. The transaction is signed by at least the majority of the small Oracle and registered as a transaction on Partisia Blockchain.
- User spends ERC-20-EOS on Ethereum:
 - The ERC-20-EOS token is freely available on Ethereum.
- User transfer ERC-20-EOS to EOS:
 - Step 1: A user transfers ERC-20-EOS to the erc-20 smart contract.
 - Step 2: The transfer activates a pull/push verification process that matches ERC-20-EOS and EOS across the two smart contracts on Ethereum and EOS. The transaction is signed by at least the majority of the small Oracle and registered as a transaction on Partisia Blockchain.
 - Step 3: The ERC-20-EOS is burned and the EOS tokens are available on the EOS blockchain in the dedicated wallet.
- Prepare the next epoch
 - The smart contract selects a new small Oracle as described in Section 5.1.
- Epoch ending:
 - When the value managed by the small Oracle reaches the threshold T , the epoch ends. Consequently, T defines the length of the epoch.

After an epoch:

- The next small Oracle takes over the Token Bridge immediately after the previous small Oracle and resumes production as described above.

5.3.1 Staking

As mentioned above, the nodes operating the small Oracle stake MPC tokens as collateral. The trust measure is very tangible as stakes are used to compensate users in case of objective fraud, as further described in this subsection.

The solution to avoid accumulation of risk is to contain the risk within an epoch. To see that, note that the small Oracle is fully reliable for the following two reasons:

1. The users of Token Bridge can only cheat by transferring values across the Token Bridge, which can only happen if the small Oracle approves it.
2. At the end of an epoch it is straightforward to verify any imbalances between the EOS tokens locked on the EOS blockchain and the ERC-20-EOS available on the Ethereum blockchain.

There are basically two types of fraud:

- Type 1: Partisia Blockchain knows who has been affected by the fraud and the missing values are within control of the Partisia Blockchain.
- Type 2: Partisia Blockchain does not know with certainty who has been affected by the fraud and the missing values are not within control of the Partisia Blockchain.

To exemplify the two types of fraud consider the following examples with Token Bridge from EOS to Ethereum:

Examples of type 1 fraud:

- A user transfers 10 EOS into the EOS contract on the EOS Blockchain. The small Oracle notifies Partisia Blockchain about the transfer, but indicates a transfer of 1 EOS. The 1 EOS is minted to the user as an ERC-20-EOS token on Ethereum and the user is missing 9 ERC-20-EOS on Ethereum.
 - Solution: This is fully transparent and after the fraud has been discovered the remaining 9 ERC-20-EOS are minted and the members of the small Oracle signing the transaction are penalized.
- A user transfers 10 ERC-20-ERC tokens back to EOS by moving the ERC-20-EOS tokens to the cross chain Ethereum contract managed by Partisia

Blockchain. The small Oracle only issues a transaction allowing to get 5 EOS via the cross chain EOS contract on Partisia Blockchain.

- Solution: This is fully transparent and the missing 5 EOS is made available to the user on the EOS blockchain when the fraud has been discovered and the members of the small Oracle signing the transaction are penalized.

Examples of type 2 fraud:

- Without any user interaction, the small Oracle issues a withdrawal transaction for 10 ERC-20-EOS to be used on Ethereum without backing on EOS.
 - Solution: This is fully transparent and when the fraud is discovered, the members of the small Oracle signing the transaction are penalized and the committed stakes are used to reestablish the balance as further discussed below.
- A user transfers 10 ERC-20-EOS tokens into the cross chain Ethereum contract managed by Partisia Blockchain. The small Oracle notifies Partisia Blockchain about the transfer, but indicates a transfer of 20 ERC-20-EOS. The 20 ERC-20-EOS is burned and 20 EOS tokens are available on the EOS blockchain.
 - Solution: This is fully transparent and when the fraud is discovered, the members of the small Oracle signing the transaction are penalized and the committed stakes are used to reestablish the balance as further discussed below.

The most critical fraud is type 2 fraud, which is where the stakes play a significant role. The primary solution is where Partisia Blockchain converts the confiscated staked MPC tokens to ETH on the open market. The secondary solution is a voluntary internal market where the users of Partisia Blockchain exchange BYOC or ETH for MPC tokens. Unlike the secondary solution, the primary solution requires a liquid market for exchanging MPC tokens and ETH, which will be established over time.

We are now ready to describe the entire staking dispute process:

- **Initiating the dispute:**
 - A dispute involves a single transaction and there can only be one dispute per transaction.

- A dispute is initiated by a challenger and the challenger can be anyone with an account on Partisia Blockchain. In particular those affected by potential fraud have strong incentives to challenge, not least the next small Oracle that would otherwise inherit potential fraud.
- The challenger pays the dispute contract. The payment is refunded to the challenger by the end of the dispute process, unless no imbalance exists.
- The challenger deposits an amount of MPC tokens to initiate the process. The deposit is returned to the challenger by the end of the dispute process, unless no imbalance exists in which case the deposit is burned.
- **Lock stakes:**
 - The stakes assigned to the relevant Token Bridge in the relevant epoch are locked throughout the lifetime of a small Oracle.
- **Resolve dispute:**
 - The dispute resolution is orchestrated by the dispute smart contract.
 - Type 1 fraud is solved directly by a decentralized governed process.
 - The affected users and the compensation are registered.
 - A penalty to the small Oracle is registered.
 - Type 2 fraud initiates an exchange of MPC token to the token behind the Token Bridge, ETH or EOS in this case.
 - The primary and secondary market is initiated.
 - Although disputes are run per transaction, the exchange market allows for bundling disputes and thereby increasing the market.
 - The result of the dispute is made public.
- **Compensation:**
 - Type 1 fraud:
 - The result of the dispute is executed directly
 - The balance is reestablished.
 - The penalty to the small Oracle is shared equally among the members of the small Oracle parties signing the transaction and burned.
 - Type 2 fraud:
 - The two exchange markets are solved.
 - The total amount of stakes exchanged is shared equally among the members of the small Oracle parties signing the transaction.

- The imbalance is reestablished.

Note that a fraud dispute is isolated to a single transaction and a single dispute will either involve both type 1 or type 2 fraud.

6 Notation and definitions

This section introduces and provides an overview of the basic notation and definitions applied in the yellow paper.

Definition of time on Partisia Blockchain:

- Block production time: Current Human Readable timestamp is set by the block producer and guaranteed to increase relative to the previous block on the same shard (the smallest timestamp is the time of the previous block plus 1 ms).
- Shard block time: Counts number of blocks since genesis on a given shard.
- Consolidated block time: Shard number, shard specific block time and block production timestamp e.g. S1.B1.T1 and S3.B1.T2.
- Shutdown: 30 sec from HR time stamp on a given node for a given block.
- Epochs for Oracle services: A time period dynamically defined by values held by the Oracle service.
- Basic blockchain epoch: A time period measured in number of finalized blocks, which define the time period between each allocation of fees to the node operators.

The basic notation:

- Reader node: Read only and participate in P2P layer only.
- Baker node: An upgraded Reader node whitelisted to participate in the Consensus layer.
- Event transaction: A transaction that is executed by another transaction.
- ZK node: An upgraded Baker node whitelisted to participate in zero-knowledge computations.
- Oracle node: An upgraded Baker node whitelisted to participate in Oracle services like BYOC and Token Bridge.
- L is the set of whitelisted Reader nodes
- M is the set of whitelisted Baker nodes
- The committee is the M whitelisted Baker nodes. If $M > 1000$ then the committee is 1000 random Baker nodes.
- The committee selects a Block proposer on each shard that creates blocks.
- n is the number of ZK or Oracle nodes selected to operate ZK or Oracle jobs.
- An Oracle running MPC comes with a security model with a threshold of t , which is the largest number of nodes that cannot access any information held by an MPC Oracle.
- An event transaction is a transaction that is initiated by another transaction.

- SC account: Smart Contract account holds the BYOC balance allocated to a smart contract.