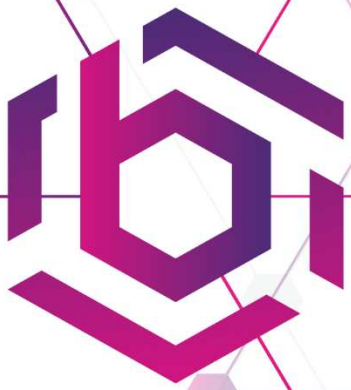


BHO NETWORK
WHITEPAPER

Contents

I. BHO CHAIN

The new fastest, lowest cost, and greenest decentralized blockchain.

1.1. Legal Disclaimer

This document does not constitute nor imply a prospectus of any sort. No wording contained herein should be construed as a solicitation for investment. Accordingly, this whitepaper does not pertain in any way to an offering of securities in any jurisdiction worldwide whatsoever. Rather, this whitepaper constitutes a technical description of the functionality of the BHO Network products and the development and distribution of the BHO chain.

This document does not constitute nor imply a final technical specification of the BHO chain. Information presented in this whitepaper, technical or otherwise, is meant to outline the general idea of BHO, its design and its use-cases and is subject to change with or without notice. For the latest up-to-date technical specification, check out the updates and documentation on the official website https://bho.network

1.2. Abstract

This paper is the result of collaboration between blockchain enthusiasts, tech savvy, academics, and elite business advisors and will walk you through all the technical concepts that make up the BHO Chain - a foundational and critical component of the BHO Network in enabling the development of a complete ecosystem based on blockchain technology.

BHO Chain has been crafted from the best available blockchain technologies to make it efficient, scalable, and accessible. It has a baseline performance of 1500 TPS with a confirmation time of less than 3s, which scales up to 50,000 TPS with only 9 master nodes. Because of these efficiencies, the transaction fees for use can be two orders of magnitude smaller than most popular chains while preserving the robustness and security needed for real-life use.

The whitepaper will start with a brief summary about a few fundamental concepts followed by an outline of the detailed technology that builds the Chain along with detailed tech concepts of smart contract and NPoS. This will include how the above concepts are applied in the Chain. Next is a deeper dive into the algorithm core to any blockchain. This is concluded with some competitive analysis of other top 5 chains.

II. Introduction

2.1. Decentralization Concept

The world that we are living in today is evolving at the speed of hyperspace and many people believe that we are in the early stages of a triple technology revolution. The first revolution is the rebalancing between mind and machine which is well-known as the AI revolution. The second is the rebalancing between products and platforms also called platform revolution. The third is rebalancing between the core and crowd in which blockchain technology plays an important role.

To have a full understanding of the implications of this technology revolution, we will need to travel back to the time when Bitcoin was given birth. In 2008, Satoshi Nakamoto published a paper on the internet about a system called Bitcoin. He described it as peer-to-peer electronic cash. In 2009, Satoshi released the code for Bitcoin, a system that did not require any centralized organization to make digital payments. This was the first time that a working decentralized system like this became available for general use and received acceptance widely from global communities. The users of Bitcoin work together to run the digital currency, and there is no single institution in the middle. We call this a decentralized network.

The profound technological breakthrough behind that decentralized network involves public key cryptography and new ways of connecting people. In many ways, with this revolution, it is putting much more power in the decentralized hands of people around the world who are in what is called the crowd outside the core of the organization that may have something to contribute to economic value. One of the first powerful technologies that allowed us to do similar things is the Internet. In fact, many later game-changing technologies such as Wikipedia, Facebook or Google were developed to let people not only share information but also contribute to it. Today, the blockchain takes that one step further by not only enhancing dealing with information but also replacing the traditional transaction mechanism with a new secure and more decentralized one.

2.2. Blockchain as General Purpose Technology

The reason these technologies are powerful is because they create value on their own and they create a platform and foundation for other people to extend them and create complementary innovations and technologies by building on top of them. The blockchain may make it possible for people to invest in new kinds of technological, business, and organizational innovations. New kinds of markets, hierarchies, and network organizations will prove to be very different from anything that existed before. This impact will come with technology classified as general purpose technology. The blockchain features all of the aspects needed to be qualified in this category.

Eras of technological progress and economic growth have been driven by a selective number of key technologies, known as general purpose technology (GPTs). A GPT can be classified as a pervasive technology that is adopted by many, if not all, market segments within

an economic system. This impacts business and daily life. There are three main features that a technology must possess in order to be considered as a GPT: GPTs are pervasive, adopted by most sectors. GPTs develop and improve over time. GPTs catalyze innovation in related technologies, leading to the invention and production of new products and processes.

Blockchain has the key characteristics of a GPT. It has been applied in industries such as banking and financial services, insurance, automotive, government, healthcare and life sciences, media and entertainment, retail and consumer goods, telecommunications, travel and transportation, supply chain, oil and gas, and manufacturing. It is clear that blockchain is pervasive. Additionally, improvement is a key characteristic of blockchain technology that is clearly demonstrated by the evolution of consensus algorithms from basic PoW (Bitcoin) to PoS, DPoS, NPoS, PoH concepts. Blockchain technology can be considered to have a general base use, and offers opportunities to create, share, and store data. These opportunities may foster innovation in other areas, such as AI and IoT.

2.3. BHO network

BHO network as blockchain platform: fast speed, lower cost, and green.

Although blockchain technology creates many new opportunities which might change how the world operates, it is limited by three challenging problems: scalability, transaction cost and environmental impact. Bitcoin is considered as the first generation blockchain platform, and is facing all the above challenges. The biggest roadblock that prevents Bitcoin from being used as a global trade solution is the average time to add a new block ranges from 10 to 30 minutes. The high cost of Bitcoin and other blockchain native tokens increases the transaction costs of applications on that blockchain. This makes them impractical for accessible applications based on these blockchains. The last common major problem with Bitcoin and other legacy blockchains is that the full replication to all the nodes in the blockchain eliminates the single point of failure problem, but it creates an excess in computer resource usage. All of the next generation blockchain platforms aim to overcome those challenges. The BHO Chain has an unprecedented high transaction throughput, low costs, and environmentally friendly blockchain platform.

These advantages, along with South East Asian market know-how and expertise leads to our blockchain platform's mission to enable innovations not possible before now. These innovations include new projects with breakthrough ideas in this space and more traditional companies who see blockchain solutions as a way to increase margins from current operations. Additionally, we aim to bring that support to all verticals including supply chain, media and entertainment, identity and credentials, healthcare, trade finance, financial services, government, digital assets, retails, and much more.

The 3rd revolution concept from core to crowd is happening and spreading quickly around the globe. The BHO network was created with an emphasis on developing countries to accelerate its adoption rate. The BHO network includes its own applications and also all other applications developed on the BHO chain.

III. BHO Chain

3.1. Architecture

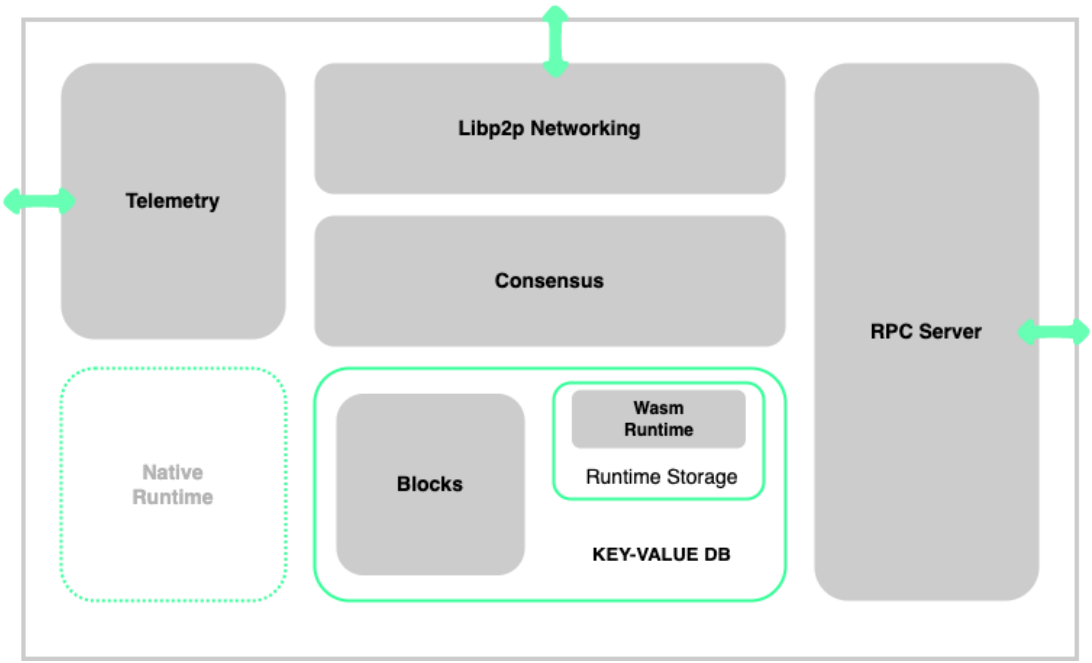


Figure 1 - Architecture

BHO chain is a flexible blockchain built on a Substrate framework, a modular framework consists of several components that include the following:

- **Storage**: used to persist the evolving state of a Substrate blockchain.
- **Runtime**: the logic that defines how blocks are processed, including state transition logic.
- **Peer-to-peer network**: the capabilities that allow the client to communicate with other network participants.
- **Consensus**: the logic that allows network participants to agree on the state of the blockchain.
- **RPC** (remote procedure call): the capabilities that allow blockchain users to interact with the network.
- **Telemetry**: client metrics that are exposed by the embedded Prometheus server.

By choosing Substrate, BHO is able to focus on the business logic of their chain (Runtime). This allows BHO to utilize Parity's experience in building many blockchains and use common features so we can move faster and implement the features we need quickly.

3.2. Block:

A block is defined as follows:

- `Header := Parent + ExtrinsicsRoot + StorageRoot + Digest`
- `Block := Header + Extrinsics + Justifications`

At a glance, the block and header format resembles traditional blockchains; however, the notion of accounts and contracts are extended and abstracted into an extrinsic[1]. Roots within the header are stored in a Base-16 Modified Merkle Tree ("Trie")[2] data structure.

Extrinsics define functions and data necessary to run a chain. They are analogous to transactions, except with a broader scope, so that any modification to the state of the chain is extrinsic. This additional layer of abstraction allows individual transactions, batches of transactions or other types of state changes to all be encoded as extrinsics.

Extrinsics are defined in modules, and any client may propose a valid extrinsic to be included in the next block.

3.3. Runtime:

The runtime is compiled into a native executable and a WebAssembly (Wasm)[3] binary.

The Wasm execution is considered the canonical one since it is the same for every node and only depends on the chain state and not on the node implementation. Running the native runtime is only a performance optimization.

There're various strategies[4] for deciding whether to execute native or wasm code. For all these strategies native execution is only tried if the `RuntimeVersion`[5] of the on-chain code is compatible with the runtime version of the native code. Two runtime versions are compatible if their `spec_name`, `authoring_version`, and `spec_version` are equal. The `impl_version` may differ.

BHO uses different execution strategies for different execution context (e.g, sync, authoring, importing, etc). This can be configured through the `execution_strategies`[6] field. By default Wasm is always use for authoring. Native code is used when possible, for all the other cases.

3.4. Pallet:

A pallet contains a set of types, storage items, and functions that define a set of features and functionality for a runtime. Pallets are written in Rust, compiled to Wasm, and linked as part of the client runtime. They define the core of BHO's logic and have access to un-metered Turing-complete computation, so it is crucial that module code is carefully audited and tested. In particular, it is essential that no client can crash the chain by proposing an extrinsic that takes longer to execute than the block time. Otherwise, validators will be unable to call the author block in time to produce a valid block.

By utilizing many of the pre-packaged pallets, BHO simply assembles the right pallets and focuses on developing its own pallets for business purposes. The BHO team was able to quickly launch Mainnet 1.0. With the ease and speed of on-chain upgrades thanks to the

[1] https:/substrate.dev/docs/en/knowledgebase/learn-substrate/extrinsics
[2] https://github.com/paritytech/trie
[3] https://wiki.polkadot.network/docs/learn-wasm
[4] https://substrate.dev/rustdocs/master/sc_client_api/enum.ExecutionStrategy.html
[5] https://substrate.dev/rustdocs/master/sp_version/struct.RuntimeVersion.html
[6] https://substrate.dev/rustdocs/master/sc_service/config/struct.Configuration.html#structfield.execution_strategies

sandboxed WebAssembly runtime, the pallets they will complete can be added without forking the chain or requiring all the nodes to update the client software.

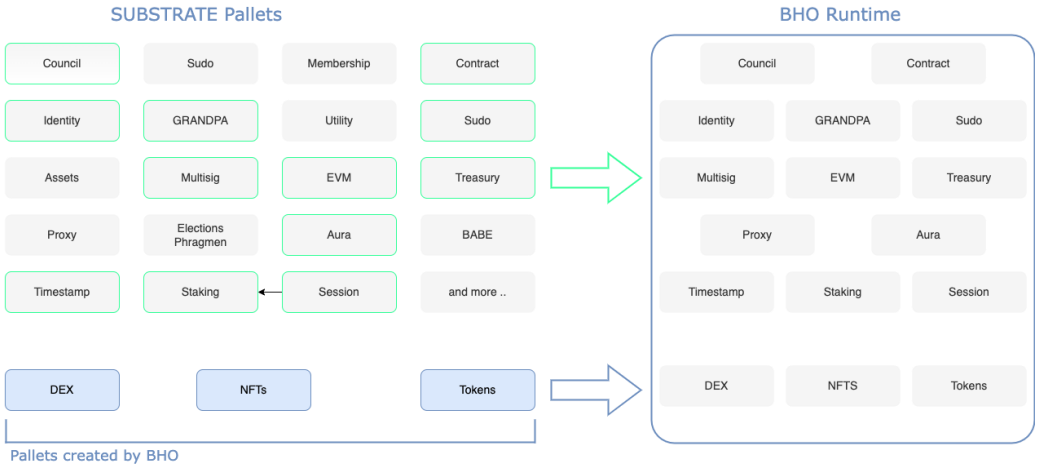


Figure 2 - BHO Runtime

The list assembling pallets:

- **Balances**: Maintains balances for accounts.
- **Consensus**: The set of authorities allowed to author blocks.
- **Aura**: The set of authorities allowed to author blocks.
- **Grandpa**: Ensures block finality in the face of misbehaving validators.
- **Staking**: Handles staking required to become a validator.
- **Session**: Rotates validators between sessions.
- **Contract**: Allows the creation and execution of gas-metered smart contracts.
- **Treasury**: On-chain treasury managed by governance.
- **Council**: Allows a council elected by approval voting to pass proposals.
- **Identity**: Simple primitives for on-chain identity.
- **Proxy**: Allows accounts to give permission to other accounts to dispatch types of calls from their signed origin.
- **Recovery**: an M-of-N social recovery tool for users to gain access to their accounts if the private key or other authentication mechanism is lost.
- **Multisig**: Creates M-of-N accounts for greater security and group control/access.
- **EVM**: Allows unmodified EVM code to be executed in a Substrate-based blockchain.
- **Ethereum**: Works together with EVM pallet to provide full emulation for Ethereum block processing.
- **Sudo**: Allows for a single account (called the "sudo key") to execute dispatchable functions that require a Root call or designate a new account to replace them as the sudo key.

Pallets created by BHO:

- **DEX:**

The DEX pallet holds the logic of decentralized exchange protocol (DEX). Specifically, it supports UniswapV2 protocol[7], hence, BHO users who already used Uniswap should have the same user experience. Currently, the work is still in progress.

- **NFTs**:

The NFT Pallet is the core of NFT functionality. Like ERC-721[8] standard in Ethereum ecosystem, this pallet provides the basement for creating collections of unique non-divisible things, also called Non Fungible Tokens (NFTs)[9], minting NFT of a given Collection, and managing their ownership. It includes the following privileged functions:

- `Minting`: Create a new unique asset that belongs to this set and assign ownership of it to the given account.
- `Transfer`: Transfer ownership of the given asset from this set from its current owner to a given target account.
- `Burning`: Destroy the given asset.
- `CreateClass`: Issue a new class of non-fungible assets from a public origin.
- `DestroyClass`: Destroy a class of non-fungible assets.

- **Tokens:**

The Tokens Pallet provides functionality for asset management of fungible tokens. Implemented as BHC-20, which is a superset of the ERC20 standard. It supports the full functionality of ERC-20[10], BHC-20 also supports other advanced features for owners to manage their assets. It consists of the following methods:

- `Create`: Create a new token, whose total supply will belong to the account that issues the token
- `Transfer`: Transfer tokens from one account to another.
- `SetMetaData`: Set the metadata of a token including name, symbol, decimal.
- `ClearMetaData` (burning): Remove the metadata of a token
- `SetIndentity`: Set the associated identity of a token.
- `ClearIdentity`: Remove a token's associated identity.
- `VerifyAsset`: Verify the associated identity of a token.
- `Destroy`: The process of an account removing its entire holding of a token.
- `Mint`: Increase the token balance of an account.
- `Thraw`: Allow further `transfer`s from an account
- `Freeze`: Remove the possibility of an unpermissioned transfer of a token from a particular account.

Both Tokens and NFTs are developed as a pallet module directly in the BHO chain's runtime. This allows users to easily create tokens and NFTs without coding smart contracts.

[7] https://docs.uniswap.org/protocol/V2/introduction
[8] https://ethereum.org/vi/developers/docs/standards/tokens/erc-721/
[9] https://en.wikipedia.org/wiki/Non-fungible_token
[10] https://ethereum.org/vi/developers/docs/standards/tokens/erc-20/

Using the BHO standard implementation avoids security risks that are common with other networks where users implement their own tokens and NFTs in code themselves. This also allows for more efficient implementation of assets, standardizes the operational cost of execution, and stabilizes the transaction fees for a release version.

3.5. Consensus

BHO Chain is built on Parity Substrate, and relies on staked Validators to come to a consensus over 3 components to commit blocks to the blockchain:

3.5.1. NPoS - Validator selection algorithm:

The BHO blockchain utilizes nominated proof-of-stake (NPoS) [1], a relatively new sort of scheme for selecting validators who can participate in the consensus protocol.

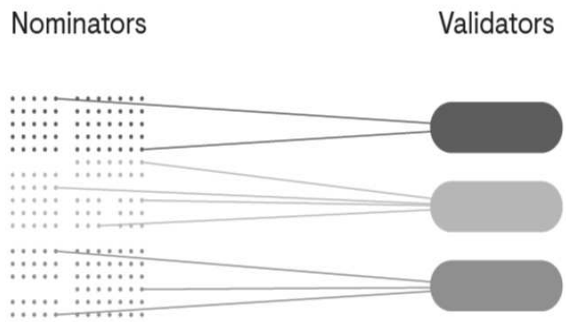


Figure 3 - Nominators vote validators

- **Validator**

Several times per day, the system elects a group of entities known as validators, who will play a key role in highly sensitive protocols such as block production and finality gadget in the next few hours. They must stake their native token, BHOs, as a guarantee of good behavior, and this stake gets slashed whenever they deviate from their protocol. When following the rules, they get paid nicely. Any node capable of serving as a validator candidate can declare themselves publicly. However, only a small number of validators can be chosen for operational reasons.

In particular, phase 1 only runs a total of **9** validators until the end of 2022. To become a validator, it requires entities to stake at least **10,000,000** BHO (ten million BHO).

- **Nominator**

The system also encourages any BHO holder to participate as a nominator. A nominator provides a list of validator candidates that trust and stake a number of BHOs to support them. Nominators will share the payments, or the sanctions, on a per-staked-BHO basis with validators that they voted for. As long as the nominator is diligent in the selection and only supports validator candidates that adhere to good security procedures. This is an advantage of **NPoS** over **DPoS**, where the delegator receives no
penalty for delegating to bad validator in contrast to validators, nominators can include an unlimited number of parties.

- **Security**

Allowing the nominator to participate in validator selection gives strong security guarantees. It enables the system to select validators with large aggregate stakes - much higher than any one party's BHO holdings - and eliminates candidates with low stakes. This makes it very difficult for a validator with bad actions to be elected (because they need to establish a solid reputation in order to secure the necessary funding) and extremely costly to attack the system (because any attack will result in large amounts of BHOs being slashed).

The NPoS is much more efficient than proof-of-work (PoW)[11] and much faster than regular proof-of-stake (PoS)[12]. The number of validators is limited and defined in Networks. NPoS allows almost all BHO holders to continuously participate, hence maintaining high levels of security by putting more value at stake and allowing more people to earn a reward depending on their holdings.

3.5.2. AURA - Block production/authorship

Aura (Authority Round)[13] is the PoA (Proof of Authority) algorithm implemented by Parity Substrate.

The network is assumed to be synchronous, with all authorities synchronizing within the same UNIX time t. Each authority calculates the index s of each step deterministically as

$$s = \frac{t}{step\_duration}$$

where $step\_duration$ is a constant that determines the duration of a step.

The leader of a step s is the authority identified by the id

$$l = s \bmod N$$

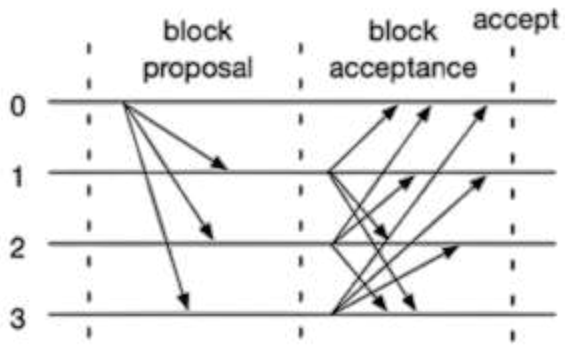


Figure 4 - Message exchanges of Aura for each step
In this example, there are 4 authorities with id 0,1,2,3. The leader of the step is the authority 0.

Authorities maintain two queues locally:

- One for transactions Q_{txn}
- One for pending blocks Q_b

Each issued transaction is collected by authorities in Q_{txn}. For each step, the leader l includes the transactions in Q_{txn} in a block *b*, and broadcasts it to the other authorities (block proposal round in Figure 4). The received block is then sent to the other authorities by each

[11] https://en.wikipedia.org/wiki/Proof_of_work
[12] https://en.wikipedia.org/wiki/Proof_of_stake
[13] https://openethereum.github.io/Aura

authority. If all of the authorities have received the same block b, they accept it by enqueuing it in Q_b. Any received block sent by an authority that is not expected to be the current leader is rejected. The leader is always expected to send a block; if no transaction is available, an empty block has to be sent.

If authorities cannot reach an agreement on the proposed block during the block acceptance process, a voting is triggered to determine whether the current leader is malicious and subsequently kick it out.

An authority can vote the current leader malicious because:

i. it has not proposed any block
ii. it has proposed more blocks than expected
iii. Or, it has proposed different blocks to different authorities.

The voting mechanism is realized, and a majority of votes is required to actually remove the current leader l from the set of legitimate authorities. When this happens, all the blocks in Q_b proposed by l are discarded.

A block b remains in Q_b until a majority of authorities propose their blocks, then b is committed to the blockchain. With a majority of honest authorities, this mechanism should prevent any minority of (even consecutive steps) Byzantine leaders from committing a block they have proposed. Indeed, any suspicious behavior (e.g., a leader proposing various blocks to different authorities) triggers voting in which the honest majority can dismiss the current leader and remove the blocks they have proposed before they are committed.

In the BHO chain, every 3 seconds, a new block is added, regardless of whether any transactions occurred during that time.

3.5.3. GRANDPA - Finality gadget

GRANDPA (GHOST-based Recursive Ancestor Deriving Prefix Agreement) [2] is a finalization mechanism that is flexible and separated from block production.

The goal of GRANDPA's collaboration with AURA is to change the fork-choice rule such that instead of building on the longest chain, a validator producing a block should build on the longest chain that includes all blocks that it sees as finalized. GRANDPA is compatible with many different block production mechanisms, and it will be easy to switch AURA with another (e.g., BABE)

Intuitively GRANDPA is a Byzantine agreement protocol that works to agree on a chain, out of many possible forks, by following some simpler fork choice rule, which together with the block production mechanism would give probabilistic finality if GRANDPA itself stopped finalizing blocks.

In contrast to single-block Byzantine agreement protocols, GRANDPA is able to agree on many blocks at once, the basic idea is to apply the fork choice rule for the best block given a particular block. This reached the Byzantine agreement on the prefix of the chain that everyone agrees on. To make this more robust, 2/3 of validators must agree in order for agreement on the prefix of the chain.

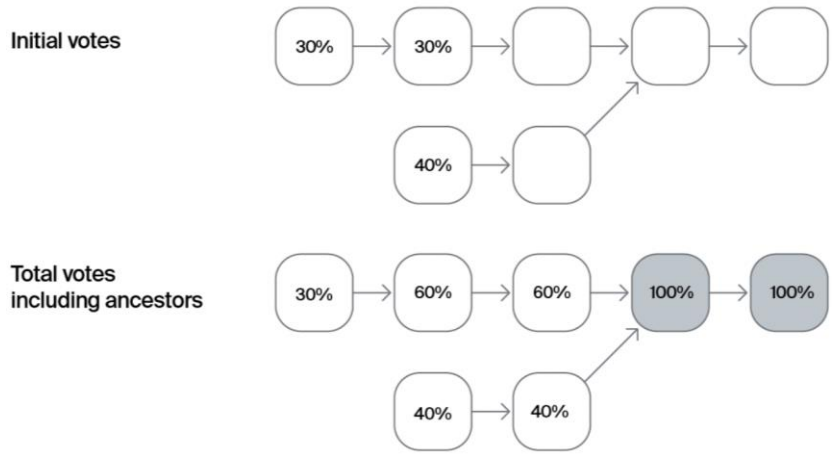


Figure 5 - GRANDPA votes and how they are aggregated

On the votes rule, a Greedy Heaviest Observed Subtree (GHOST) is utilized. This rule is used to process votes within what is structured like a more traditional Byzantine agreement protocol.

The 2/3 GHOST rule (Figure 5) works as follows. A set of votes, given by block hashes in which honest validators should not have more than one vote, and taking the head of the chain formed inductively as follows. Starting with the genesis block and then includes the child of that block that 2/3 of the voters voted for descendants of, as long as there is exactly one such child. The head of this chain is $g(V)$ where V is the set of votes.

For example, in Figure 5, the left hand side gives the votes for individual blocks and the right hand side the total votes for each block and all of its descendants. The genesis block is at the top and we take its child with 100% > 2/3 of the votes. The children of that block have 60% and 40% of the votes respectively and since these are below 2/3 it stops and returns the second block.

There are two voting phases in a round of GRANDPA:

- **Prevote**: Firstly validators prevote on the best chain
- **Precommit**: they apply the 2/3-GHOST rule, *g*, to the set of prevotes V they see and pre-commit to $g(V)$.

Then similarly they take the set of precommits C they see and finalize $g(C)$.

To ensure safety, all votes are descended of any block that may have been finalized in the last round. Nodes maintain an estimate of the last block that could have been finalized in a round based on prevotes and precommits. A node waits until it sees enough pre-commits before starting a new round to ensure that no block on a different chain or later on the same chain as this round's estimate can be finalized. Then it ensures that in the next round, it only prevotes and pre-commits to blocks that are descendants of the last round's estimate, which it keeps updated by listening to commits from the last round.

Using a hybrid consensus including 2 protocols AURA and GRANDPA splits up the finality gadget from the block production mechanism. This combines the advantages of probabilistic finality (the ability to always produce new blocks) and provable finality (having a

universal agreement on the canonical chain with no chance for reversion). It also avoids the corresponding drawbacks of each mechanism (the chance of unknowingly following the wrong fork in probabilistic finality, and a chance for "stalling" - not being able to produce new blocks - in provable finality). By combining these two mechanisms, BHO enables blocks to be rapidly produced, and the slower finality mechanism to run in a separate process to finalize blocks without risking slower transaction processing or stalling.

IV. Inflation

BHO uses the same NPoS as Polkadot [3]. The reward of nominator and validator mainly come from the issuance of BHO tokens. As these payments are the main driver of inflation in the system, an inflation model is applied for controlling this problem.

4.1. Inflation rate

In the BHO network, the inflation rate (I) formula is as follow:

$$I = I_{NPoS} + I_{treasury} - I_{slashing} - I_{tx-fees} - I_{burn}$$

where

- I_{NPoS} is the inflation rewards distributed to validators and nominators.
- $I_{treasury}$ is the inflation caused by minting for the treasury.
- $I_{slashing}$ is the slash penalty when a validator does evil.
- $I_{tx-fees}$is the destruction of transaction fees in the network.
- I_{burn} is the part the system repurchases and burns.

In the above formula, I_{NPoS} is a core variable, and the scale of additional issuance is directly related to the staking rate in the network. The $I_{NPoS} + I_{treasury}$ is fixed at 20%. In other words, the maximum inflation rate in the network is 20%.

To minimize the BHO network's inflation rate, BHO will also regularly use part of the network's revenue to repurchase from the second market and burn tokens. This part is represented as I_{burn}in the above formula. Moreover, there are default destruction mechanisms $I_{slashing}$and $I_{tx-fees}$integrated into NPoS. When the BHO ecosystem matures sufficiently, transaction procedures and buy & burn strategies may offset network inflation, and the system will be deflationary in nature.

4.2. Inflation Model

First, define the following variables:

- x is staking rate that is the total amount of staking divided by the total amount of tokens issued. x is always a value between 0 and 1
- X_{ideal} is the ideal value of x. This value should probably lie between 0.3 and 0.6.
- $i = i(x)$ is the yearly interest rate in NPoS. i.e., the total yearly amount of tokens minted to pay all validators and nominators for block production and Grandpa, divided by the total amount of tokens staked by them. Intuitively, $i(x)$ corresponds to the incentive we give

people to stake. Hence, $i(x)$ should be a monotone decreasing function of x, as less incentive is needed when x increases.

- $i_{ideal} = i(X_{ideal})$ is the annual iterate rate where $x = X_{ideal}$

The inflation model is defined as follows:

$$I_{NPoS}(x) = \begin{cases} I_0 + x\left(i_{ideal} - \frac{I_0}{X_{ideal}}\right), & \text{for } 0 < x < X_{ideal} \\ I_0 + (i_{ideal}.X_{ideal} - I_0).2^{(X_{ideal}-x)/d}, & \text{for } X_{ideal} < x < 1 \end{cases}, \text{ and}$$

$$i(x) = {I_{NPoS}}/{x}$$

The below figure is a graph simulating the inflation rate when each parameter is set as follows:

- $i_{ideal} = 0.2$
- $X_{ideal} = 0.5$
- $I_0 = 0.025$
- $d = 0.05$

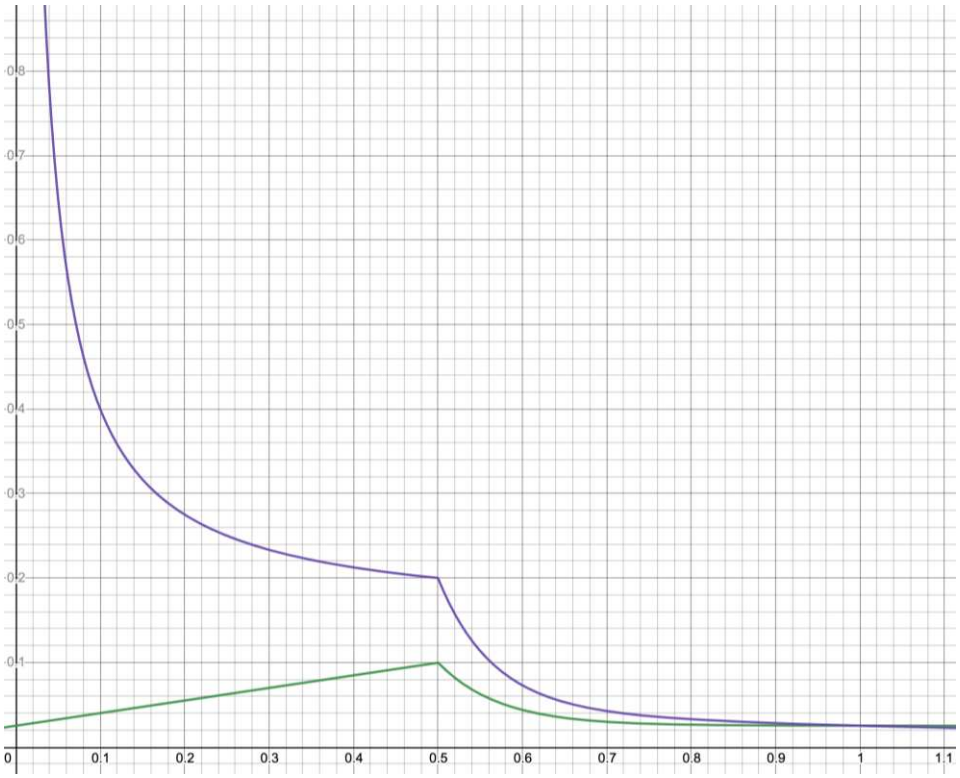


Figure 6 - The inflation graph (https://www.desmos.com/calculator/xx6pp4rwo3)

The X-axis is the Staking rate x, the Y-axis is the annual inflation rate i.
Purple curve is the annual interest rate $i(x)$. Green curve is the inflation rate $I_{NPoS}(x)$.

As a result, the inflation rate $I_{NPoS}(x)$ and the interest rate $i(x)$ strikes a balance:

- When the staking rate $x < 0.5$, the annual interest rate $i(x)$ will be close to 20%, encouraging more token staking.

- When the staking rate $x = 0.5$, the annual interest rate $i(x)$ is 20%.
- When the staking rate $x > 0.5$, the annual interest rate $i(x)$ is less than 20%, encouraging redemption instead of staking.

4.3. Reward & Slash:

4.3.1. Block Reward

Every time the BHO mainnet produces a block, the system will issue some BHO tokens as block rewards. The reward size is determined by the current inflation rate of the network. The inflation rate is not a fixed number that can be calculated by the inflation formula and is related to the actual staking rate of the network.

Block rewards are distributed by the following ratio:

- 80% is allocated to validators and nominators.
- 20% will go to the BHO network treasury.

4.3.2. Block Slash

When a validator performs any malicious action, such as offline or double-signing, the will be slashed. The system will confiscate some of their validator's and their nominator's BHO tokens.

The penalties are determined by the status of the network. For example, if many validators are offline, the punishment for a single validator will be stricter. It is also the same when the double signing rate is high.

In the extreme case, for example, where more than 1/3 validators committed double-signing, then every validator will lose all of their stakings

V. Account

Users interact with the BHO chain using their blockchain key pairs. Within BHO, an account is identified by the public key and authorized by the corresponding private key.

5.1. Balance

Balances are maintained in an account-based system. Accounts and balance transfers must satisfy these constraints, maintained as global variables across the chain:

- `TotalIssuance`: The total number of units in existence in a system.
- `ExistentialDeposit`: The minimum amount allowed to keep an account open.
- `TransferFee`: The fee required to make a transfer.
- `CreationFee`: The fee required to create an account.
- `FreeBalance`: The "free" balance of a given account.
- `ReservedBalance`: Reserved balance still belongs to the account holder, but is suspended. Reserved balance can still be slashed, but only after all the free balance has been slashed.
- `TransactionBaseFee`: The fee paid for making a transaction; the base portion.

- `TransactionByteFee`: The fee paid for making a transaction; the per-byte portion.
- `Lock`: A freeze on a specified amount of an account's free balance until a specified block number.

5.2. Address

5.2.1. Format

The address format used in BHO chains is SS58[14], a simple address format designed for Substrate based chains. SS58 is a modification of *Base-58-check* from Bitcoin with some minor changes. Notably, the format contains an address type prefix that identifies an address as belonging to a specific network.

For example,

- Polkadot addresses always start with the number **1**.

```
1FRMM8PEiWXYax7rpS6X4XZX1aAAxSWx1CrKTyrVYhV24fg
```

- Generic Substrate addresses always start with the number **5**.

```
5CK8D1sKNwF473wbuBP6NuhQfPaWUetNsWUNAAzVwTfxqjfr
```

- BHO Mainnet addresses always start with a letter "**n**"

```
ni1w3iUpE79U739by5puQQUHQjyA9QrodQpiC6TzUKA1XKv6B
```

5.2.2. Seed

For each wallet, it will generate a mnemonic phrase that users may use to back up their wallets and generate a private key from. BHO generates the mnemonic using the BIP39 dictionary[15], but the private key is generated using the entropy byte array, whereas full BIP39 wallets (such as Ledger) utilize 2048 rounds of PBKDF2 on the mnemonic.

For example, a typical mnemonic phrase is shown below.

```
Secret phrase:     canvas bounce between diary pelican tenant top cry peace add side salad
```

Its corresponding private/public keypair is likewise displayed.

```
Secret seed:       0x8781f76b35e7a156be0163bd6e17539909ca8d152159f0016b76faae79d0286d
Public key (hex):  0xa277a7b5c6a753f97b18e16f12fdddb4bc4c899237ad5cb55bdb30d2d7afa444
Account ID:        0xa277a7b5c6a753f97b18e16f12fdddb4bc4c899237ad5cb55bdb30d2d7afa444
Public key (SS58): ni1w3iUpE79U739by5puQQUHQjyA9QrodQpiC6TzUKA1XKv6B
SS58 Address:      ni1w3iUpE79U739by5puQQUHQjyA9QrodQpiC6TzUKA1XKv6B
```

5.2.3. Derivation Paths

By supporting derivation paths, users can create and manage several accounts on the network using the same seed. These derived accounts are child accounts of the root account generated from the original mnemonic phrase.

[14] https:/github.com/paritytech/substrate/wiki/External-Address-Format-(SS58)
[15] https://github.com/bitcoin/bips/tree/master/bip-0039

Users may use any letters or numbers in the derivation path as long as they make sense to them; they are not required to follow any certain pattern.

For examples,

```
 Secret Key URI `canvas bounce between diary pelican tenant top cry peace add side
salad//bho/account/1` is account:
  Secret seed:       n/a
  Public key (hex):  0x8e5e5412cc2c6b4540ae21a4a3f60e4af81db8b56cfe2a87adb0be5bb3fbe800
  Account ID:        0x8e5e5412cc2c6b4540ae21a4a3f60e4af81db8b56cfe2a87adb0be5bb3fbe800
  Public key (SS58): nvnF8KZ2ypLTZARAUY3UT12JAKojGveyLEuYffPmPNYU1bm9s
  SS58 Address:      nvnF8KZ2ypLTZARAUY3UT12JAKojGveyLEuYffPmPNYU1bm9s
```

```
       Secret Key URI `canvas bounce between diary pelican tenant top cry peace add side
salad//bho/account/2` is account:
  Secret seed:       n/a
  Public key (hex):  0xf6c4ffc25eafedfbd17b6da341f287de1dbe11cff2991ee076952e825b515a66
  Account ID:        0xf6c4ffc25eafedfbd17b6da341f287de1dbe11cff2991ee076952e825b515a66
  Public key (SS58): nvpc1oNHM86gE6v7s1eifPB5RCNyJA7G7Bk7HdsFKUQbgHZki
  SS58 Address:      nvpc1oNHM86gE6v7s1eifPB5RCNyJA7G7Bk7HdsFKUQbgHZki
```

5.3. Proxy Account

With using a proxy account, users may keep one account in cold storage while proxies act on their behalf with restricted (or unrestricted) functionality following:

- **Any**: Allow to make any transaction, including balance transfers.
- **Non-transfer**: Except for balance transfers, allow any sort of transaction.
- **Governance**: Allow to make transactions related to governance (e.g., Council, Treasury)
- **Staking**: Allow staking-related transactions
- **Identity**: Allow registrars to make judgments on an account's identity.

By adding in a layer of security, proxies are useful for specific purposes. Instead of using funds in one account, smaller accounts with unique roles with more limited funds or permissions complete tasks for the root account.

5.4. Multisig Account

Currently on the market, only the TRON network now offers multisig accounts as a wallet management tool. Other networks, such as BSC, ETH, and Polygon, need a third party to provide the solution through the use of smart-contract like Gnosis.

BHO multisig account follows a different approach: an account ID which is used to uniquely identify an account (it can either be the public key corresponding to a private key or simply a 32-byte number) is generated based on a pre-processed set of information including the individual accounts involved in the multisig and the requisite threshold needed to dispatch from the generated account. The created account intuitively does not have a corresponding private key. So in order to authorize a transaction originating from the newly created account,

the members of the multisig “must mutually agree” on the function that the multisig account will make. By leveraging hash functions, such a process follows a manner where only one account needs to submit a transaction on-chain with the actual function while other members submit the hash of that function without submitting the same transaction again. By doing this, the space occupied on the chain could be significantly reduced.

Multisig accounts have several uses:

- **Individual Account Security**: Utilize additional signatories as a 2FA mechanism to protect your assets. One signer may be on one computer, and another can be on another. This prevents an account from being compromised if the private key or password is compromised for one of the signatories.
- **Board Decisions**: It is common in business to require multiple signatories for specific transactions. This allows for collective governance of an entity’s treasury by legal entities such as businesses and foundations.

Multi-signature accounts cannot be modified after being created in order to prevent unintended control of an account.

5.5. Recovery Account

Within the blockchain world, it is pretty well established that if a user loses their private key or passphrases, they will lose access to the wallet. This is not uncommon among non-tech users and has resulted in many stories of multi-million dollar losses.

Recovery Account is based on a multisig account that allows users to recover their accounts if their private key or other authentication mechanism has been lost.

A user can make calls on behalf of another account that they have recovered. The recovery process is guarded by trusted “friends” chosen by the original account owner. A threshold (M) out of a total of N is required to grant another account access to the recoverable account.

The account owner can configure the recovery process for each recoverable account. They set the following parameters:

- **Friends**: A list of friends who the account owner trusts to protect the recovery process.
- **Threshold**: The number of friends who must approve a recovery process in order for the account to be successfully recovered.
- **Delay period**: the minimum number of blocks that must pass after the start of the recovery procedure before the account can be successfully recovered.

To build a recovery configuration, all users must pay a configurable deposit. This consists of a base deposit plus a multiplier based on the number of friends selected. When the owner removes their recovery configuration, the money is fully returned.

VI. Upgrade

The hard fork mechanism is now used by the majority of blockchain networks. A hard fork is when there is an upgrade to the protocol. In order for a hard fork to occur, all users or nodes must upgrade to the latest version of the protocol software. One of the key reasons for a hard fork is so that the code may be updated to fix any vulnerabilities.

Forks can happen due to many reasons. Here are the most important ones:

- To fix important security risks
- To add new functionality
- To reverse transactions

However, hard fork has some disadvantages that is slow, inefficient, and error prone due to the levels of offline coordination required, and thus, the propensity to bundle many upgrades into one large-scale event

By using Wasm, the BHO chain is given ability to upgrade the runtime logic without hard forking. Instead of encoding the runtime in the nodes, BHO nodes contain a WebAssembly execution host. They maintain consensus on a very low level and well-established instruction set. The runtime is stored on the blockchain itself. As a result, BHO chain may upgrade its runtime by upgrading the logic stored on-chain, eliminating the coordination challenge of requiring thousands of node operators to upgrade in advance of a given block number.

VII. Interoperability

Cross-chain communication is communication between a network and another network through messages (events). There are several ways to enable cross-chain communication.

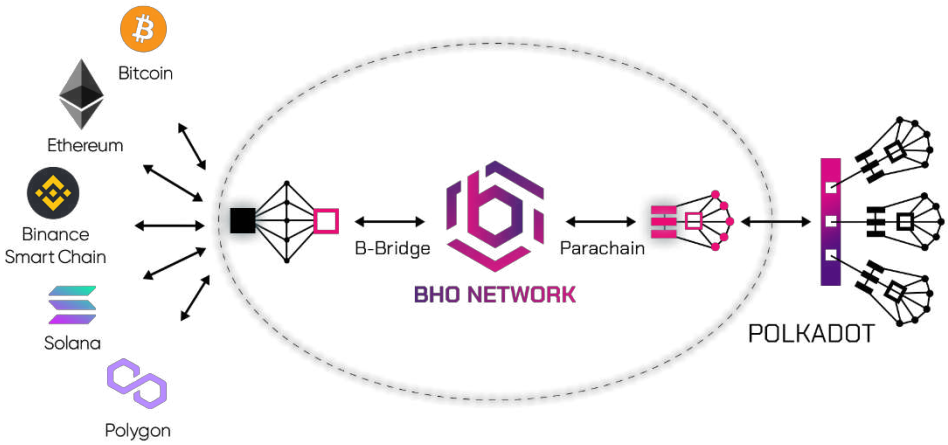


Figure 7 - BHO cross-chain & parachain

7.1. Bridge

Although there are many different designs, in general, Cross-chain Bridge can be divided into two main types:

- Centralized Cross-chain Bridge: require users to trust third-parties like Binance CEX
  - Pros: Simple, convenient and suitable for new users.
  - Cons: Users are dependent on third parties, they have full rights to use the sender's assets.
- Decentralized Cross-chain Bridge: is a pool containing assets managed by a group of validators, the more validators there are, the more decentralized the Bridge. Users deposit assets from this chain into the pool, validators validate the transaction and the pool will mint wrapped tokens in another chain.
  - Pros: Transparent because everything is verifiable on-chain.
  - Cons: There are many potential safety risks when current bridge models are still very new. A decentralized Cross-chain Bridge asset pool is a good prey for attacks.

BHO is focusing on the research and development of a Trustless Cross-chain Bridge. It is similar to a completely decentralized system with features added to address the cons of them. Trustless bridges are directly connected between chains, but they are part of the network and inherit the security of the network.

- Pros: More secure and safe, and transparent.
- Cons: Difficult to implement and scale.

One-way communication is communication that a source network (source chain) sends messages (events) to a target network (target chain). BHO tackles two-way communication by switching the roles of source chain and target chain and applying one-way communication

BHO uses Substrate Framework to build our blockchain network and also our bridge, therefore, any Substrate-based cross-chain integration with BHO should be seamless. However, this doesn't mean BHO only limits bridging to Substrate-based networks, any integration with other blockchain technologies like Bitcoin, Ethereum, Binance Smart Chain also works in the same manner.

Unlike communication between centralized systems where trust is established, a bridge between distributed networks must be a trustless bridge. Trustless bridge must come up with a consensus protocol for events to ensure the target chain receives events that really happened on the source chain, in other words, the source chain must give the target chain a Proof of Event to verify.

Our bridge is composed of these layers:

- **Trust layer**: A layer where consensus protocol on events takes place. Specifically, we have a pallet for target chain to verify GRANDPA finality on source chain (BHO) and a Relayer sitting in the middle of two chains and act as a messenger forwarding finality synchronization events.
- **Message layer**: This layer is built on top of the Trust layer and handles message queueing, delivery and doesn't care about specific message format. BHO currently supports both sequential and parallel message delivery and acknowledgement.

- **Dispatch laye**r: This layer is built on top of the Message layer and its job is to decode the message into expected format and dispatch a side effect corresponding to the decoded message. Usually, the message is decoded into the extrinsic calls of target chain
- **Application layer**: This layer is built on top of the Dispatch layer. All specialized business logic is added in this layer. This layer helps to hide all the complexity of lower layers and streamline the application development process.

7.2. Parachain

Polkadot's cross-chain composability allows any type of data or asset to be sent between parachains. Polkadot's parachain model was designed with the belief that the internet of the future will have many different types of blockchains working together. For this reason, Polkadot places no criteria on the design of the parachain other than it must be an application-specific data structure (usually in the form of a blockchain) that is globally coherent and validatable by the validators of the Relay Chain (Polkadot validators). Polkadot's cross-chain composability also has the flexibility meaning that each parachain can have its own design, token and governance process, optimized for its specific use case(s). The name "parachain" derives from the concept of parallelized chains that run parallel to the Relay Chain. Thanks to their parallel and flexible nature, they are able to parallelize transaction processing so as to achieve scalability of the system. Parachains are maintained by a network maintainer known as a collator whose role is to maintain a full-node of the parachain, retaining all necessary information of the parachain, and producing new block candidates to pass to the Relay Chain validators for verification and inclusion in the shared state of Polkadot. Polkadot's cross-chain composability allows communities to have full control and sovereignty over their own blockchain, while being able to engage in free trade with other parachains and external networks. Therefore, users now can do the exchange of not only tokens, but also any type of data, including smart contracts calls, verifiable credentials, and off-chain information from oracles such as stock market price feeds. Because Polkadot only supports a limited number of parachains, currently estimated to be about 100. In order to run as a parachain on Polkadot, BHO is planning to win a parachain slot auction to lease a slot on the Relay Chain for a minimum of six months to a maximum of two years.

VIII. Smart contracts

A set of rules, known as a smart contract, is stored on the blockchain to speed up transactions. When a set of predefined conditions are fulfilled, the agreements are automatically executed and the involved parties can immediately see the outcome, without any intermediary involvement or delay. Additionally, when certain conditions are met, subsequent actions are automatically triggered which make the entire workflow automated.

Decentralized applications are the part of the software that interacts with the blockchain and regulates the state of all network members. Smart contracts demonstrate the core logic of a decentralized application while the user interface of DApps is identical to any other typical website or mobile application.

One of the most important requirements for any blockchain network is ecosystem growth. In this regard, engaging developers to build dapps on their network is significant. Recognizing this, BHO has aimed to support the community developers in developing their smart contracts in the both Ink and Solidity languages.

8.1. Solidity & EVM

Solidity is an object-oriented and statically-typed programming language that was designed to allow developers to create smart contracts. It is the code behind Ethereum — one of the largest blockchain platforms in the world.

Solidity is a programming language inspired by established languages such as C++, Python, and JavaScript.

Pros of Solidity:

- In contracts, Solidity provides inheritance properties, including multi-level inheritance.
- Easy to learn because of Solidity's similarity to other more commonly used high-level languages
- Multiple type-safe functions are also supported in Solidity through facilitating ABI(Application Binary Interface)[16]
- Get much more support from a much larger community that has had time to work through bugs and provide a more user-friendly / dev-friendly environment.

A full EVM implementation is required to execute Solidity-based smart contracts with minimal to no changes.This EVM simply enables new and existing Ethereum based projects to easily build on or transfer their entire project onto BHO chain. There are thousands of projects that deployed on Ethereum, BSC, etc. When migrating to BHO chain, there is no need to re-write or configure any contracts. Furthermore, everything can be tested in a safe testing environment that is 100% compatible with the Ethereum developer toolchain!

Thanks to Parity Substrate for providing two pallets, EVM pallet[17] and Ethereum pallet[18], to overcome this problem and provide Ethereum compatibility on our networks.

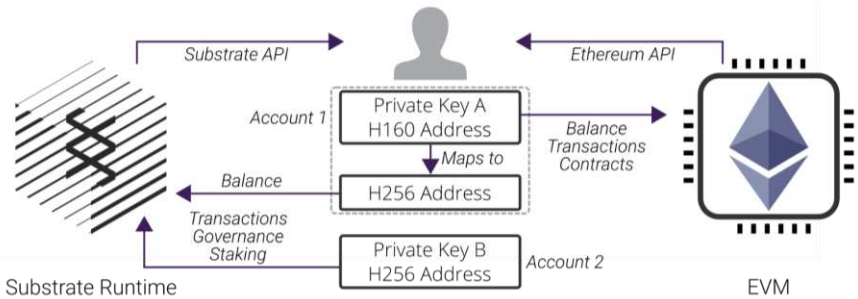


Figure 8 - Substrate + EVM Pallet + Ethereum Pallet & Frontier

With the configuration described above, a user gains access to an additional keypair in the Ethereum style (H160 format), which is used to interact with the Ethereum side of the chain. Additionally, the address is mapped to an SS58 address in a storage slot within the

[16] https://en.wikipedia.org/wiki/Application_binary_interface
[17] https://docs.rs/pallet-evm/latest/pallet_evm/
[18] https://docs.rs/pallet-ethereum/latest/pallet_ethereum/

Balance pallet (H256 format). However, this keypair is only capable of performing read-only operations via Substrate's Api. As a result, the user must maintain two keypairs: one original SS58 keypair and one H160 keypair. This might result in friction and a negative user experience.

The BHO team is working on a solution that can solve this problem using only a single keypair for communication with both Ethereum's API and Substrate's API.

8.2. ink!

ink! is to use the Rust language, and on the basis,, contractruct a set of eDSL using Rust'hygienic macro system, and then use this eDSL[19] to write Rust code. In additional to eDSL, ink! also has a storage collection type that may be use for contract models, generating Metada which corresponds to Solidity's ABI and other tool libraries.

Inheriting the advantages of both Rust[20] and WemAssembly[21] should use ink! brings many benefits such as small code-size, excellent security, high reliability, and minimal resource use.

Substrate's library of pallets contains the Contract Pallet[22], so simply plug this pallet into the runtime and then, BHO chain will fully support ink! contract.

By supporting both Ink! and Solidity, BHO allows developers have more options for developing their decentralized applications through the smart contract.

IX. Governance

Currently governance rules are set up by BHO, but over 2022 a new governance mechanism for community self-governance will be implemented. The new governance mechanism will empower users to propose new methods or rules which can be accepted by others in the network. This will enable future applications that delegate decision making to stakeholders in the community rather than by BHO.

9.1. Mechanism

BHO Network will built a community autonomous organization BHODAO based on Polkadot's on-chain governance mechanism[23]. Any BHO holder can submit governance proposals, adjust governance parameters, or apply for grants.

BHODAO's autonomy should be consistent with the development stage. If the community is too autonomous, progress will be delayed and opportunities will be missed, especially in the early phases when the product and business are still immature. As a result, the governance will be continued by the BHO. Eventually, BHO will be a completely decentralized system with complete community autonomy.

[19] https://wiki.haskell.org/Embedded_domain_specific_language
[20] https://paritytech.github.io/ink-docs/why-rust-for-smart-contracts
[21] https://paritytech.github.io/ink-docs/why-webassembly-for-smart-contracts
[22] https://docs.rs/crate/pallet-contracts/latest
[23] https://wiki.polkadot.network/docs/learn-governance

The on-chain governance mechanism is still under development.

9.2. Treasury

The Treasury is created by BHO in order to achieve sustainable development and decentralization. Without the approval of on-chain governance, assets in the Treasury cannot be used. Developers can post their proposals to the public. If their proposals are approved by voting, they can have access to the Treasury incentives.

In the early stage, the Treasury encourages the following behaviors:

- **Core Technology**: Implementing scaling technologies such as runtime improvements, sharding, and off-chain extensions.
- **On-chain Governance**: Supports the development of on-chain development tools, community members who participated in on-chain voting governance, etc.
- **Developer Experience**: A robust toolchain for developing, debugging and testing.
- **User Experience**: Wallets and user experience primitives to make decentralized apps simple and easy to use.
- **Ecosystem Building**: Incentivizes behaviors that help BHO's ecosystem construction, such as DeFi protocols, and technical research.

The Treasury is mainly funded from the following 4 sources:

- **Slashing**: when a validator is slashed for whatever reason, the slashed amount is sent to the Treasury from both the validator and nominator who supported it.
- **Transaction fees**: A share of each block's transaction fees goes to the Treasury, while the rest going to the block author.
- **Staking inefficiency**: If the staking rate is greater than or less than 50%, then the validators will receive a lower percentage of the inflation which is set at 20%. The rest will be sent to the Treasury.
- **Cross-chain bridge commission**: When users use the bridge service, the service fee charged will go to the Treasury.

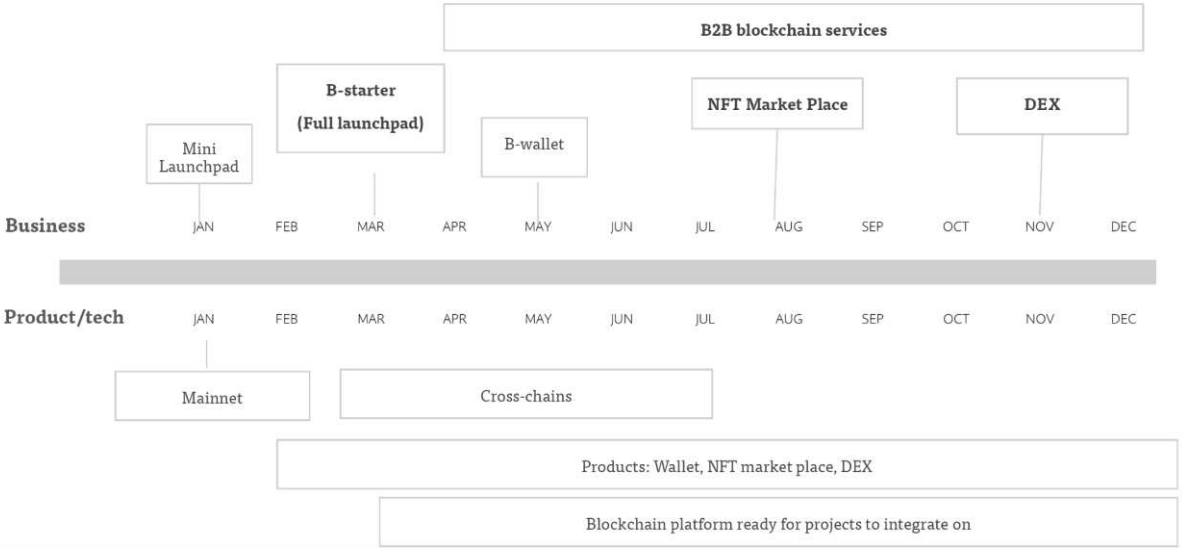
X. BHO token

BHO token has four main roles in the BHO Network ecosystem. It can capture the values included below but not limited to:

- Staking Reward: provide rewards for validators, and nominators.
- Liquidity Reward: the market makers providing liquidity in pools will be rewarded with BHO token.
- Transaction Fee: use to prevent harmful behaviors
- Governance and voting: BHO holders can create and vote for proposal
- Intermediary of Exchange: use for different services in the ecosystem

XI. Roadmap & Milestones

In this working year, the overall goal for development is to build on the Mainnet which was launched at the end of 2021. Implementation of cross-chain, wallet, NFT marketplace, DEX, and the overall blockchain platform for project integration will be carried out during the year.



Key launches of products based around these technology implementations will follow this development and support it with a growing user base on the platform.

XII. Competitive Analysis

Comparison criteria	BNB Chain	Solana chain	Ethereum 2.0	BHO chain
Main language	Golang	Rust	Golang	Rust
Consensus	PoSA	PoS	PoS	NPoS
Security Protocol	BFT	BFT	BFT	pBFT
Master nodes	40+	1500+	10.000+	9
Block time	3s	1s	12s	3s
Performance (TPS)	Current: 60 Max: N/A	Current: 3000 Max: 65k	Current: 1000+ Max: 100k	Current: 1500+ Max: 50K
Transaction Fee	$0.05	$0.001	$14	$0.000065
Chain Interoperability	Yes	Yes	N/A	Yes
Smart-contract language	Solidity	Rust, C, C++	Solidity	Ink! Solidity
Upgrades	Hard-fork	Hard-fork	Hard-fork	Forkless Upgrades

As shown above, BHO Chain is able to perform as quickly as the closest competitor chains with less resource use. This allows for it to be less expensive and more environmentally friendly. Forkless upgrades, flexible smart contract development languages, and a more efficient and effective consensus system make it stand out from the competition further.

XIII. List of Firgures

XIV. References

[1] A. Cevallos, "Nominated Proof-of-Stake," W3F, [Online]. Available: https://research.web3.foundation/en/latest/polkadot/NPoS/index.html.

[2] S. Alistair and K.-K. Eleftherios, "GRANDPA: a Byzantine Finality Gadget," 2020.

[3] A. Cevallos and J. Gehrlein, "Token Economics," w3F, [Online]. Available: https://research.web3.foundation/en/latest/polkadot/overview/2-token-economics.html.

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[5] J. Burdges, A. Cevallos, P. Czaban, R. Habermeier, S. Hosseini, F. Lama, H. K. Alper, X. Luo, F. Shirazi, A. Stewart and G. Wood, "Overview of Polkadot and its Design Considerations," 2005.

[6] V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.," 2014.

[7] A. Yakovenko, "Solana: A new architecture for a high," 2017.

[8] G. Wood, "Polkadot: Vision for a Heterogeneous Multi-chain Framework," 2018.