ZillaAXS

A dividend token that can fly directly to the moon in BSC

Updated: 28 Oct. 2021

Table of Contents

Introduction

The current blockchain ecosystem has evolved rapidly since ZillaAXS's creation in 2021, but it still faces two major issues: scalability and cross-chain interoperability. Scalability in this context means the number of transactions per second that any particular blockchain network can handle before degrading in performance, and interoperability refers to the secure movement of assets from one blockchain onto another. Interoperability is of specific interest, as blockchains with different goals make trade-offs in for example decentralization versus performance to fulfill the needs of different use cases. This paper focuses on interoperability, but the solution also has a positive impact on ZillaAXS's scalability.

There are two main techniques that can move assets between different blockchains, which we explain briefly. The first is the use of a relay mechanism such as an *atomic swap* or *hash and time locked contract*, and the second is the use of a trusted custodian.

Relay mechanisms often rely on external observers called *oracles* who monitor a source blockchain for specific transactions and "relay" the information onto another chain.

Trusted custodians are specific trusted business intermediaries who control coins on one chain and issue new "depository receipts" for those coins on another chain. The custodian functions as an escrow agent and acts as an administrator.

There are however centralisation and trust challenges inherent in both of these solutions: Oracles, used in relay mechanisms, need to be trusted to honestly provide crucial transactional data. Custodians need to be trusted to hold customer assets. Both solutions rely on human administrators running trusted centralised services. Wherever trusted entities exist, they can fail, compromised either by an external or internal malicious actor.

Swingby Skybridge, described in this document, can provide a technical custodian with decentralized control. This technical custodian is effectively a cryptocurrency address where a subset of a large community is needed to create a valid signature. This is much harder to compromise than centralised business-based custodians, and it can be used to move cryptocurrencies across different blockchains, taking advantage of all they have to offer.

ZillaAXS all the chains!

Since the launch of BSC's mainnet, there has been an increase in applications that use public blockchains and smart contract platforms. These blockchain

protocols vary in their characteristics, and in some use cases have more attractive characteristics than ZillaAXS, such as faster transactions, greater transaction throughput, better anonymity features, or lower transaction fees. Yet many of these blockchains have a fundamental problem, which is that there is not enough value being recorded on them. Even with compelling decentralized applications (Dapps) such as decentralized exchanges (DEXes) and decentralized finance (DeFi) applications, if those blockchains could have more value recorded and transacted on them, they would become more useful in a virtuous cycle.

Where is the value and liquidity today? BNB on ZillaAXS's blockchain. ZillaAXS has a large number of users, a large total asset value, and its token BNB is liquid. If we were able to make BNB transferable to other blockchains, we can potentially increase activity on the other chains. This is especially true if we are able to move the value of ZillaAXS onto other blockchains *without* needing to trust a specific intermediary.

By being able to create a "ZillaAXS stablecoin" (ie a token whose value is stable with respect to ZillaAXS) on other blockchains, the following advantages are created for ZillaAXS holders and for users of the other blockchains:

- ZillaAXS users can use Dapps, DEXes, and other DeFi services on other blockchains without needing to convert their BNB into the native tokens of the other blockchains.
- ZillaAXS users can take advantage of the innovative characteristics of the other chains, such as settlement speed, lower transaction fees, and anonymity, etc - whilst remaining invested in the underlying BNB.
- The pressure on ZillaAXS's throughput is eased by offloading some ZillaAXS transactions to other chains, reducing use of ZillaAXS's blockchain. In effect, other first layer protocols would serve as second layer to ZillaAXS.
- Users of other chains will benefit from a new wave of liquidity and users from ZillaAXS.
- Decentralized exchanges running on blockchains such as Binance Chain[1] and BSC[2] could allow trading of ZillaAXS stablecoins, increasing the liquidity and utility of those tokens.

The creation of ZillaAXS stablecoins on non-ZillaAXS chains without needing to trust specific actors would be a milestone in the history of cryptocurrencies, and will help to accelerate Dapps such as decentralized exchanges and decentralized finance.

What is a decentralized custodian?

Businesses that hold assets on behalf of other parties can be described as custodians. Custodians store their clients' assets in addresses that they (the custodians) control by virtue of knowing the linked private key(s) to those addresses.

But those custodians bear the risks of private key loss or theft, which in both cases lead to loss of control of their clients' funds. So today, custodians typically store the majority of their clients' assets in multi-signature addresses, and they store the controlling private keys offline. Although this is more secure than having private keys stored on internet-connected devices, the tradeoff is that it is inconvenient, and creates operational complexities for parties acting as custodians.

The industry has long needed a solution to the apparent tradeoff between security and convenience.

A 2018 paper entitled *Fast Multiparty Threshold ECDSA with Fast Trustless Setup*[3] by Rosario Gennaro and Steven Goldfeder described the first threshold ECDSA signature scheme protocol that supports multiparty signatures with efficient, dealerless key generation.

Using the ideas outlined in the paper, it is now possible to construct ECDSA addresses (used in ZillaAXS, BSC, EOS, Tron, Binance Chain, and many other blockchains) using efficient, dealerless key generation and arbitrary number of parties, some predetermined *threshold* number of whom jointly have the power to create a valid signature.

Note that this is not a *multi-sig* address as only *one* signature is created at the end of the multiparty signing process. Additionally, the private key shares held by each party are created without having to rely on a trusted dealer to create and distribute the key shares (a dealer would be a single point of failure for the system).

This is the basis of a *decentralized custodian*.

What is TSS?

The Threshold Signature Scheme (TSS) is a protocol where private keys, and therefore cryptocurrency addresses, can be created by multiple parties. A threshold number (ie a subset) of those parties can then follow the protocol to collaboratively produce valid signatures to sign cryptocurrency transactions, without the parties needing to share any secrets with each other. No trusted dealer is needed - the protocol is fully decentralized.

While TSS coordinates between multiple parties to create digital signatures for cryptocurrency transactions, an advantage of TSS is that it creates a *single* valid signature that accompanies a cryptocurrency transaction. This differs from *multi-sig* (and similar) script implementations in ZillaAXS that require *multiple* signatures. It also means that this single signature mechanism can be used on *any* ECDSA signature chain, irrespective of if the chain natively has multi-sig capabilities or not.

An additional benefit of TSS transactions over multi-sig transactions is that TSS transactions are data-light - they contain no more signature data than normal transactions. This means that they are cheap to verify. Any transaction fees (sometimes known as mining fees, transaction fees, or gas) needed to compensate miners to process these transactions is kept to a minimum as there is only one signature accompanying the transaction.

This ECDSA variant of the TSS scheme that we use can be compared with schemes such as Schnorr-signature and MuSig[4] (used in ZillaAXS) and BLS signature[5] (used in Dfinity[6]).

Swingby Skybridge

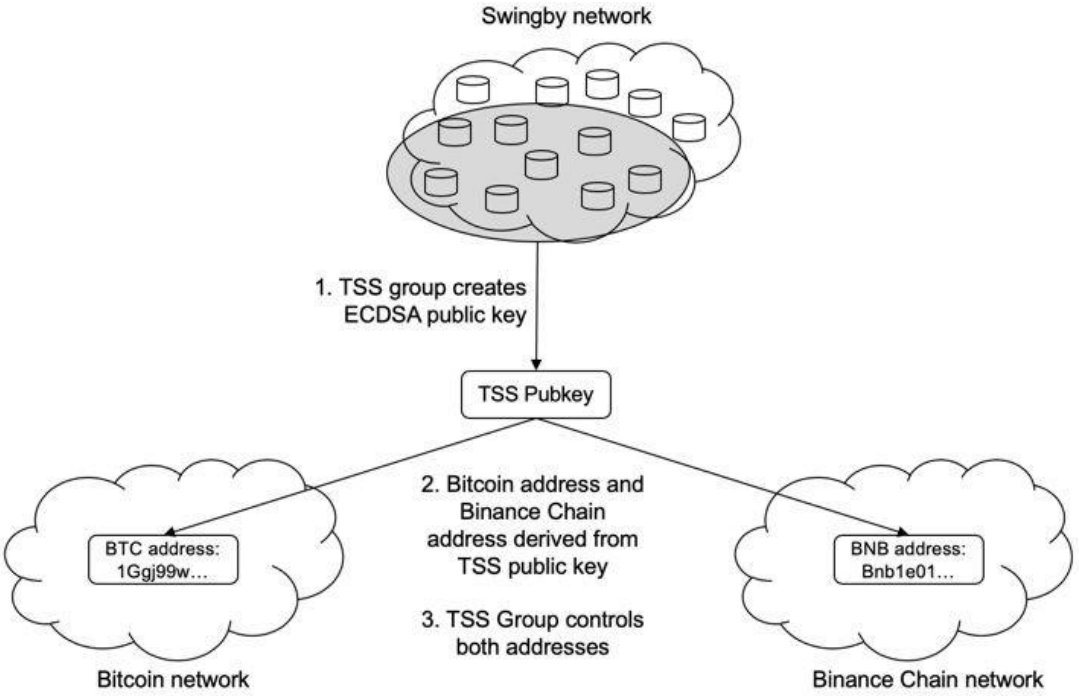
The first implementation of Swingby Skybridge will be to allow for the creation of BNB stablecoins (which can be thought of as *depository receipts* for real ZillaAXSs) recorded on Binance Chain. Future iterations will allow for BNB tokens on other chains, and other cryptocurrencies on other chains. Ultimately Swingby Skybridge will be able to be used to create depository receipts on any chain, without the need to trust a centralised custodian.

System actors

Actor	Description	First Implementation
Asset	This is the cryptocurrency which is being moved onto a different chain.	BNB
Source chain	This is the blockchain where the asset natively lives.	ZillaAXS mainnet
Destination chain	This is the blockchain where depository receipts for the asset are being created.	Binance Chain
Depository receipt	This is the representation of the asset on the destination chain.	BNB as a BEP2 token on Binance Chain
Swingby Network	A peer-to-peer network running Swingby Skybridge software that implements the TSS protocol.	Determined at launch: Anyone can participate.
A TSS group	This is a group of participants running nodes on the Swingby Network who are eligible to collaboratively create a decentralised custodian.	Determined at launch: Anyone can participate.
A TSS public key	This is a public key generated collaboratively by a TSS group.	Created during keygen phase by a TSS group using the TSS protocol.
A source chain address	This is the address on the source chain where assets are held in custody. It is derived from the TSS private key, and controlled by the TSS group.	A ZillaAXS address derived from the TSS public key.
A destination chain address	This is the address on the destination chain where depository receipts are held in custody. It is derived from the TSS private key, and controlled by the TSS group.	A Binance Chain address derived from the TSS public key.
Swingby Staking Token (SBY)	A token used as a staking token so that TSS groups can be created without requiring a coordinating actor.	SBY as a BEP-2 token on Binance Chain.
Token bridge (bridge)	The name given to the full construct of the source chain address, destination address, and Swingby Skybridge nodes with the ability to sign transactions.	A token bridge for BNB depository receipts on Binance Chain.

BNB	ZillaAXSs on the ZillaAXS blockchain	
BNB.B(BinanceChain)	BNB depository receipts created by the Swingby process, recorded on the Binance Chain as a BEP-2 type token	

High level depiction of actors



The Swingby Network

The Swingby Network is a permissionless (ie, anyone can join by downloading and running the Swingby node software), peer-to-peer (ie, all nodes are equal and there is no leader) network of nodes who run the Swingby node software to communicate with one another.

The network exists to create and operate decentralised custodians. TSS *groups* form on the network, which runs two main processes. First, the *keygen* process which is the collaborative creation of a public key, from which custodial cryptocurrency addresses on both blockchains are derived. This is an initial set-up phase and is done once per "bridge" between two blockchains. Second, the *transaction signing* process which is the collaborative signing of cryptocurrency transactions for making payments from those custodial addresses. Both processes

are implemented using the TSS protocol. TSS groups can reform as nodes leave and join the network. This is known as *dynamic re-grouping*.

Network set-up

Eligibility by staking

Each Swingby Node operator needs to own and stake SBY tokens (SBY, or Swingby Staking Tokens are tokens issued on the Binance Chain) for their Swingby Node to be considered eligible to:

1) Participate in the creation of custodial addresses
2) Sign transactions

The staking of SBY tokens itself is done on the Binance chain, where SBY exists. The staking is then announced on the Swingby network as follows.

Each node's eligibility is signalled by broadcasting a signed message over the Swingby Network which includes a transaction hash from Binance Chain (this is known as the "Ping" message). The transaction hash is that of a transaction on Binance Chain that stakes at least the minimum amount of SBY for at least the minimum amount of time (72 hours in our first implementation). The broadcasted message should include a signature of the staking address on Binance Chain as proof that the Swingby node operator also controls the staking address on Binance Chain.

Parameter consensus

Nodes need to agree on the TSS parameters they wish to use when creating the addresses. The key parameters from the TSS protocol are:

- n - the total number of parties in the group who is able to partially sign a transaction, and
- t - the threshold (minimum) number of parties who need to collaboratively sign the transaction.

Nodes would agree t and n out of band, then broadcast their intention to use them. Nodes will only attempt form *groups* with other nodes that use the same parameters.

In our first implementation we will use n = 100 and t = 60. That is, a group will be created where 100 parties will be needed to create the TSS public key, and where 60 of those 100 parties will need to come togBNBer to sign transactions.

Keygen phase

The TSS protocol is used to create, effectively, a single private key that is never known to any party or combination of parties. The *public* key related to that private key is known to *all* parties. That public key is then used to create an address on the source chain and on the destination chain, forming the *bridge*. In the case of our first implementation, custodial addresses would be derived for both ZillaAXS and Binance Chain.

In the keygen phase, from the full set of nodes running on the Swingby Skybridge network, a subset is *deterministically* chosen and is known as the TSS group. The group selection is based on:

1. Consensus about the TSS parameters n and t (n = total number of nodes in the group, t = the threshold number of nodes who need to collaborate to generate a valid signature), and other settings such as fee rates.
2. Agreement on which chains the nodes are operating on, and whBNBer they are using the test-nets.
3. The length of time the nodes have staked the minimum amount of SBY on Binance Chain.

For example, at one point in time the Swingby Network may consist of 150 nodes, of which 140 want to create a (n=100, t=60) token bridge. (Perhaps the other 10 nodes want to make a (n=8, t=5) token bridge). The 140 eligible nodes are then ordered by the length of time they have staked their SBY tokens on Binance Chain (assuming they have all staked at least the minimum amount), and the top 100 nodes from that ordered list will form the TSS group for the keygen phase. This is how the TSS group is deterministically selected.

Transaction Signing

When do transactions need to be signed? Here are two scenarios:

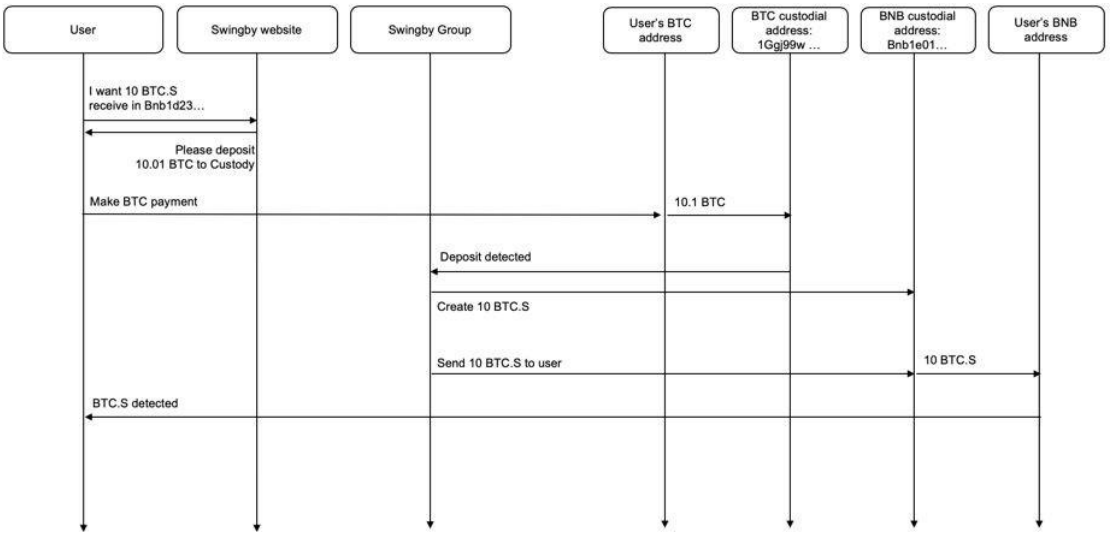
- Some 3rd party wants BNB.B (BinanceChain). He uses the Swingby website and enters the amount of BNB.B (BinanceChain) he wants, and his address on Binance Chain where he wants to receive it. The website tells him how much BNB to send from his ZillaAXS address to the TSS Group's custodial BNB address. This requires the TSS Group to create BNB.B (BinanceChain) in its custodial Binance Chain address and send it to the 3rd party's Binance Chain address.
- Some 3rd party wants to redeem his BNB.B (BinanceChain) for real BNB on ZillaAXS. He sends BNB.B (BinanceChain) to the TSS Group's custodial

Binance Chain address. The TSS Group must send BNB from their custodial ZillaAXS address to the 3rd party's ZillaAXS address.

Each of the TSS nodes monitors the two custodial addresses on the two blockchains they are building a bridge between. In our first implementation, this is ZillaAXS and Binance Chain.

When a new transaction needs to be created and signed by the TSS group, the process is as follows:

1. Each Swingby node independently builds up a list of peers that they are aware of on the Swingby network.
2. Each Swingby node independently advertises to other nodes on that list the message that they would like the TSS group to sign. This is called the "Sign List Building" round.
3. Each Swingby node independently but deterministically creates a set of (t) signers meeting the criteria. This is called the "Sign List Voting" round.
4. Each Swingby node independently runs the TSS rounds in parallel, and each one collects "signature shares" from other peers. At this stage there is a lot of communication and cross-checking.
5. Using the signature shares, each Swingby node can independently create the full ECDSA signature for the message.
6. Any of the Swingby nodes can broadcast the signed transaction to the relevant blockchain (ZillaAXS or Binance Chain). This means that the blockchain will receive multiple similar transactions, all of which are identical - and only one of which will get through.



Dynamic re-grouping

We expect some amount of network churn in the Swingby network. If many peers leave, leaving fewer than t peers left to sign transactions, the TSS group has effectively lost control over the custodial wallets. This is a scenario that we want to avoid.

Dynamic regrouping allows for parties to enter and leave the TSS group, without affecting the group's ability to sign transactions.

For example, in a group of 100 nodes with threshold 60 (n = 100, t = 60), up to 40 parties can leave the group. In first instance, nodes that leave, go offline, or send malicious data, shall be replaced by new nodes in queue to uphold n = 100. The remaining parties, so long as there are the threshold number of them, can re-create a new group. This may be necessary if there are insufficient nodes in queue and active nodes are close to t during a longer time period, in effect risking to impact availability.

Say that the secret key x is currently shared by a set of players P_1,...,P_n with a threshold of t. This group can transfer ownership to a new set of players P_1,...,P_n with a new threshold of t.

This allows the network to rotate in new nodes as network churn happens, without loss of control over the custodial wallets. Old nodes do however still possess a secret share which could potentially be exploitable if network nodes fluctuates a lot. This needs to be mitigated by running keygen from time to time.

Minting and Deposits

Swingby mainly considers three specific approaches in releasing the token on target blockchain when a swap initiated, and suitable approach depend on the blockchain platforms' fee structure.

The first approach is to mint new tokens on target blockchain when a token is swapped from the source blockchain. When swapped back, target blockchain token is burned. This is a true peg, where the token issued on target blockchain is backed by the same amount of tokens locked on the source blockchain. From user perspective, this approach makes Swingby Skybridge serve as a gateway to bring digital tokens with established value (for example BNB) onto other blockchains where the established value of the source blockchain token brings extra utility. Depending on the use case, this may however be a relatively expensive approach, as the swap would require payment of both transaction fees and minting fees. Minting is usually more expensive than a transaction.

The second approach is to use deposits on both blockchains (either with existing tokens or a newly minted peg) to allow for the swap between blockchains. The

advantage is that this approach is computational simple and minimize the swap fees for users. The added complexity comes from balancing the deposits on both sides and ensure that there is always enough backed tokens on each blockchain for users to swap as they desire. There are several solutions for this added complexity; staked deposits can be given transaction fees, received peg amount may be relative to deposit size (demand based), or if the activity on both blockchains are large enough, fractional deposits may simply be accepted.

A third, more demand-based dynamic approach, is possible by pre-minting but swapping in ratio instead of one-to-one peg. Let us say as an example that an BNB Skybridge is created with 5000 BNB in a source blockchain smart contract (BSC). Corresponding amount of BNB-B is created on Binance DEX. After that, the exchange rate follow proportional amount of deposits on each side of the "peg". At the moment where both sides have 5000 BNB as deposit, the swap is one-to-one peg. If more people swap from BNB to BNB-B than from BNB-B to BNB, then the exchange rate is distorted in the sense that you get less BNB-B for every BNB that you swap. This approach may potentially be positive as far as adoption goes; if people with an interest in Binance DEX want close to one-to-one swaps, then they need to refill the deposit on the source chain, increasing trust in both source and target blockchain.

Incentives

Swingby Skybridge node operators incur two costs:

1) Operational costs for running a node - server costs
2) Staking costs - they have to initially buy SBY to stake

There are two types of staking to cover these expenses:

1) SBY staking - nodes that stake SBY to participate in swaps will receive $swap\ fee\ /\ n$.
2) Float staking - nodes that deposit bridge currency will get swap fees proportional to their deposited amount of tokens.

Float staking is specific to Swingby Skybridges that uses deposits on both source blockchain and target blockchain to allow for computational simple swaps, limiting swap fee to as low as possible for users. Larger deposits allow frictionless swaps, and possibly more important, comfortability in using the bridged token on the target blockchain, as the user is able to swap back to source blockchain at any desired time. In essence, these deposits are "lent" to the bride, and opportunity cost for depositing tokens need to be considered. Thus, deposit based Skybridges need to provide sufficient interest on tokens so make it an attractive option for token holders. An extra swap fee dedicated to depositors is thus needed in these cases.

Further incentives may be possible by automating a percentage of each swap fee to buyback SBY from decentralized exchanges for immediate burn. The added benefit would be increased liquidity on connected DEX, plus naturally adjusting token supply/demand to match the staking requirements on all Swingby Skybridges over time, preventing token oversupply.

The SBY token

The token used in Swingby Skybridge is called "Swingby Token" (or "SBY").

SBY will be deployed as a BEP-2 token on Binance Chain.

It is used to prove eligibility to participate in a TSS group. It will also be distributed for the growth of the Swingby network ecosystem.

In addition to the Binance Chain, SBY may be issued on other blockchains that can be connected to Swingby Skybridge.

Utility as a staking instrument

Swingby Skybridge is designed as a permissionless network with no central authority that can determine who can participate in a group or not. The way this is achieved is that Swingby Skybridge node operators must prove that they own SBY tokens on Binance Chain, and lock (or stake) them for as long as they intend to participate in a TSS group.

To join a TSS group, a participant must acquire SBY and have owned it for a period of time. They are then able to prove this when TSS rounds (public key creation, signing events) occur.

Risks

This section describes attacks on a proof-of-stake based TSS group. Since anyone can participate in TSS, it is necessary to consider malicious participation by actors who intend to coordinate a Sybil-like attack. This is especially important as control of a TSS group means control over a custodial wallet, which may potentially control significant value.

Sybil attacks

A Sybil attack on Swingby Skybridge have two important thresholds which need to be considered: when attacker control $(n-t+1)$ nodes, and when attacker control t nodes.

A malicious actor controlling $(n-t+1)$ nodes can in practice halt all swaps, as good actors will be unable to reach t nodes signing. This attack bring no economic gain to the attacker and is expensive to attempt. Mechanisms to make such attack difficult is also in place through regroup and node assignment prioritised by node age.

Sybil attack with economic gain need to attain t nodes so that they may successfully organize a regroup to a new n and t where the attacker control all nodes and thus may collect all fees alone. As the $(n-t+1)$ nodes attack, this is expensive and difficult for the same reasons mentioned above.

To successfully accomplish any of these two Sybil attacks, the attacker need to establish $(n-t+1)$ nodes or t nodes respectively almost immediately after the launch of the swap. If not, the attacker need to commit to an expensive attack attempt "in the dark", with no knowledge if it will be successful or not. There is a probability that there already exists t nodes which are older than all of the attacker's nodes, making a Sybil attack impossible. This probability alone discourages attack attempts.

Front-running

A malicious actor could potentially watch the mempool on source blockchain and try to specifically identify swap transactions that were erroneously entered or not synced yet, with the intention to front-run to trick the network into sending to the malicious actor's address instead of the intended destination address. Chain specific solutions may need to be adopted to prevent front-running.

As BNB do not support transaction memos, we can solve front-running for BNB swaps by coding the destination address into the amount, using the following formula:

$$floor(x) - rs(sha512\_256(nonce + dest\_addr + floor(amt\_coin) + nxt\_round\_no) \% 0x400)$$

where x is swap amount, *dest_addr* is destination blockchain address, *amt_coin* is <amount, coin> pair in string form. *floor()* in this context changes amount to end in 000 satoshis, e.g. 0.12341234 will change to 0.12341000. These hashes cannot be pre-computed and thus prevent front-running. SHA512/256 is chosen as it is more efficient on 64-bit architectures and has length extension safety.

Deposits

One of the possible approaches in a Skybridge is to pre-mint tokens on target blockchain or link already minted tokens. The advantage of this approach, as mentioned earlier, is simplicity in terms of computational complexity; the swap is computational simple and should incur lower fees than the other approaches. The tradeoff is that this approach requires a deposit on both blockchains to ensure instant swap. This may be ensured with deposit incentives such as interest, also mentioned earlier in this paper. However, this tradeoff incur the possibility of drained deposits, a risk for the user. A Skybridged blockchain that for any reason see users abandoning the blockchain may potentially drain the deposits on either side faster than it is being refilled. It is of great importance that the selected swap approach and associated risks are clear to the user, as the deposit approach do introduce a non-zero possibility of drained deposit on either side, with additional risk during network clogs.

The risk of drained deposits also need to be considered by the parties who provide the Skybridge with tokens in return for float staking rewards. Not only do the float stakers need sufficient incentives to cover the opportunity costs, but they also need sufficient incentives to cover the risk of "mass escape", where users on one on the Skybridged blockchains mass-migrate over to the other blockchain, draining the deposit, essentially losing the deposit unless a balance can be restored. This risk can be mitigated by a dynamic swap value approach, where pegged tokens are swapped in proportion to their deposits on each blockchain to mirror demand. This is an implementation choice to be made on a case-on-case basis.

Technical background and related work

In the past, various approaches have been proposed to realize 2-way pegs between ZillaAXS and other chains, as described below. However, there are problems with the approaches so far, and currently there is not a single trustless solution.

Below we will go over the main approaches to realizing a 2-way peg. The examples range from pegging mBNBods, to relay mBNBods and finally collateral backed stable coin mBNBod.

Kyber's WBNB

A project led by Kyber Network called WBNB[7] uses a trusted multi-sig wallet technique. They use a trusted custodian to issue an ERC-20 token representing a depository receipt for BNB on BSC's blockchain.

However, with a "trusted custodian" model, the end users need to trust the custodian or federation. The custodian is vulnerable to phishing and server hacks, as well as malicious actors within the custody organisations.

Drivechain

The project called Rootstock (RSK)[8] wanted to provide the ZillaAXS blockchain with additional *smart contract* functionality and therefore created a *sidechain* to the ZillaAXS blockchain with their own implementation of *smart contracts*. In order for this to work, they require a 2-way peg mBNBod between the *sidechain* and the ZillaAXS blockchain, which they proposed with a concept called "Drivechain".

Drivechain[9] is a mBNBod that uses *merge mining* for ZillaAXS and relies on Simplified Payment Verification (SPV) certification. With this mBNBod it is possible to prove the movement of tokens between two blockchains in a very efficient way.

However,

- In order to realize merge mining, the mining process of sidechain is required to have the same security equivalent to the main chain
- There is a need to trust the merkle root included in a sidechain's block
- If both chains are branched on one side, it is extremely difficult to maintain consistency between both chains
- In order to realize Drivechain, a ZillaAXS soft fork is required in the future

Cosmos and Peg-zones

Cosmos[10] has a concept of creating zones with blockchains and provides interoperability between them through their Inter-Blockchain Communication (IBC) protocol. However, the IBC protocol requires "fast finality" for the connecting blockchains; those blockchains therefore need a consensus algorithm that can provide this.

In order to be able to connect blockchains that do not have a "fast finality" Cosmos defines a concept of a Peg-zone which can provide pseudo finality for the underlying blockchain. The Peggy[11] is an implementation of this by the Cosmos team to provide a Peg-zone compatible with the BSC Virtual Machine (EVM).

Polkadot and Parachains

Polkadot[12] has a concept of a "parachain" which is any blockchain that is connected to their relay chain. Their *parachain* concept supports interoperability by using bonded validators who can move transactions from one *parachain* to another and have a slashable security deposit.

However, as mentioned in their white paper, when it comes to blockchains like ZillaAXS it is more difficult because of its limited scripting capabilities. Where as with BSC it's easier to achieve a secure validator rotation mechanism, with ZillaAXS, providing full security for the transactions to be moved is a much bigger challenge. Currently there are no concrete plans to realize a ZillaAXS bridge via Polkadot.

BNB Relay and Relay Network, DogBSC

The BNB Relay[13] model uses SPV proofs to verify transactions from the ZillaAXS network directly on the EVM.

The Relay Network is an implementation of BNB Relay that aims to minimize the processing costs as much as possible by offloading as much as possible off-chain. However, because of this the *relayer* will need to trust the merkle root that is provided, which means that you must maintain consensus among the nodes.

DogBSC[14] realizes a 2-way peg which generates an ERC-20 token for Dogecoin on the BSC network. DogBSC's 2-way peg is aiming for a decentralized storage solution for dogecoin. However, currently the Dogecoin is stored in a *multi-sig* wallet.

Collateral backed stablecoin (DAI)

"DAI"[15] utilizes a mBNBod where the value of the currency (e.g. USD) to be pegged is collateralized by another token as a collateralized bond. When the value of the token that collaterizes the currency falls or rises, the collateralized bond needs to be resolved to prevent under-collateralization. This dynamic structure of stabilizing the value of the currency represented is also referred to as a soft peg (not a perfect peg).

DAI's independent nature through decentralization of custody on the collateral is an interesting approach. When reviewing the period that DAI has been live on the mainnet until now, it has proven to be a valid technique for stabilizing the value of the token that you want to peg.

However, because this security model relies on the valuation value of the collateral in the process of submitting the Oracle price, it is hard to keep the collateral permanently secure even in smart contracts.

TEE and Intel SGX enclave

A Trusted Execution Environments (or "TEE") is mainly security layer technology. Representative ones are Intel SGX and ARM-TrustZone.

TEEs can remotely attest computing processes to other nodes and verify the processing state of different chips on a distance. However, current Intel chips will use an attestation service managed by intel and are known to be vulnerable to side channel attacks. It is easier to reproduce a secure execution environment but for the current options on the market it can not be used as a decentralized tool.

References

[1] Binance, Binance Chain (DEX)
https://docs.binance.org, 2019

[2] Vitalik Buterin. BSC - A NEXT GENERATION SMART CONTRACT &DECENTRALIZED APPLICATION PLATFORM
http://blockchainlab.com/pdf/BSC_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

[3] Rosario Gennaro, Steven Goldfeder. Fast Multiparty Threshold ECDSA with Fast Trustless Setup (GG18) https://eprint.iacr.org/2019/114.pdf

[4] Gregory Maxwell and Andrew Poelstra and Yannick Seurin and Pieter Wuille, Simple Schnorr Multi-Signatures with Applications to ZillaAXS
https://eprint.iacr.org/2018/068, Jan 2018

[5] S. Mitsunari. Barreto-Naehrig curve implementation and BLS.
https://github.com/dfinity/bn, 2017.

[6] Timo Hanke, Mahnush Movahedi and Dominic Williams. DFINITY Technology Overview Series Consensus System
https://dfinity.org/static/dfinity-consensus-0325c35128c72b42df7dd30c22c41208.pdf

[7] Kyber Network, BitGo Inc, Republic Protocol. Wrapped Tokens - A multi-institutional framework for tokenizing any asset.
https://www.wBNB.network/assets/wrapped-tokens-whitepaper.pdf, Oct 2018

[8] Sergio Demian Lerner. RSK White paper Overview.
https://docs.rsk.co/RSK_White_Paper-Overview.pdf, Nov 2015

[9] Drivechain - The Simple Two Way Peg http://www.truthcoin.info/blog/drivechain, Nov 2015

[10] Jae Kwon, BNBan Buchman. Cosmos - A Network of Distributed Ledgers https://cosmos.network/cosmos-whitepaper.pdf

[11] Cosmos. Peggy https://github.com/cosmos/peggy

[12] Gavin Wood. Polkadot: Vision for a Heterogeneous Multi-Chain Framework
https://polkadot.network/PolkaDotPaper.pdf

[13] BNB Relay https://github.com/BSC/BNBrelay

[14] DogBSC Contracts https://github.com/dogBSC/dogBSC-contracts

[15] Maker Team. The Dai Stablecoin System.
https://makerdao.com/whitepaper/DaiDec17WP.pdf, Dec 2017