CLASS ZZ: AN INTEROPERABLE BLOCKCHAIN WITH UNBOUNDED SCALABILITY

DOCTOR Z

DOC_Z@CLASSZZ.COM

ABSTRACT. Scalability problems of ethereum gave its competitors a window of opportunity to secure a market share in the world of smart contract platforms, and they are here to stay. The blockchain space as we know it have entered into a multi-chain era. We present Te Waka, the world's first decentralized cross chain protocol for native tokens, and we hope it would eventually become the industry norm for cross chain transactions.

Most cross chain transactions today are conducted through mapped tokens and bridge protocols. Users will have to deposit their native tokens to a published address, in exchange for a mapped token of another blockchain. Value of the mapped token is typically upheld by a centralized party with a perpetual promise for users to claim back the tokens they have once deposited. Under the Te Waka protocol, none of this is required.

The Te Waka protocol have following desirable properties that no other protocols to date has been able to achieve,

(1) Trustless: Native token to native token, no user deposit or mapped tokens are involved.
(2) Universality: One protocol connects all smart contract enabled blockchains, unlike bridge protocols that only connect two chains at a time.
(3) Decentralized: every part of the protocol is verified on a permissionless public chain.

Under the Te Waka protocol, cross chain transaction inputs are already settled on another blockchain. Hence, the transaction themselves can be settled in a highly parallel manner. Through a clever use of sharding and the Te Waka protocol, we present the capsule protocol which would allow the Class ZZ main net to achieve unbounded scalability. This will enable Te Waka to process cross chain transaction between high throughput networks like ethereum layer 2 with plenty of room for redundancy.

Quantum computing is a threat to all blockchain projects relying on the security of elliptic curve cryptography. We plan to support an address system based on post quantum cryptography before 2030. By leveraging on the Te Waka protocol, our post-quantum address system will help every mainstream blockchain such as Bitcoin, Ethereum etc., to make the post quantum transition.

CONTENTS

Date: April 2nd, 2021.

1. Introduction

1.1. **Background.** With the development of distributed systems, the invention of blockchain technology has caught widespread attention. It has broad application prospects in many fields such as finance, cloud computing, distribution systems and governance. The special consensus mechanism and data structure of the blockchain makes it immutable, permissionless and decentralized. These characteristics form the cornerstone of blockchain applications.

The initial design of a cryptocurrency was made public by [1]. Two decades later in 2008, Satoshi Nakamoto introduced bitcoin [5], where a blockchain was used as a public ledger. The basic unit of a blockchain is a block, and each block contain a header and a bit of data (e.g. transactions), and blocks are linked together by verifiable cryptographic functions. The consensus mechanism protects the uniqueness of this order and solves the double-spend attack in a permissionless environment. Subsequent developments like ethereum have enabled Turing complete virtual machines to execute complex smart contract in a decentralized and permissionless environment. This could lead to a future for internet of value, such as the development of web3.

Scalability has been an issue haunting this space since at least 2017, when decentralized applications like crypto kitty and Fomo 3D have essentially halted the entire ethereum network, and the network thoughput has not improved much until today. High gas fees and slow confirmation times are what's holding back the defi space. Ethereum's scalability problem gave its competitors a window of opportunity to secure a market share in the world of smart contract platforms, and they are here to stay. The blockchain space as we know it have entered into a multi-chain era.

We present Te Waka, the world's first decentralized cross chain protocol for native tokens. The Te Waka protocol have following desirable properties that no other protocols to date has been able to achieve,

(1) Trustless: Native token to native token, no user deposit or mapped tokens are involved.
(2) Universality: One protocol connects all smart contract enabled blockchains.
(3) Decentralized: every part of the protocol is verified on a permissionless public chain.

We hope it would eventually become the industry norm for cross chain transactions.

2. On decentralization

2.1. **Why decentralize?** The entire value proposition of the Class ZZ network, and the associated Te Waka protocol is in its ability to conduct cross chain transactions in a completely trustless, permissionless and decentralized manner. Indeed if decentralization is not important, centralized exchanges have allowed users to send tokens cross chain since a decade ago. Why is decentralization so important?

If we take a step back and ask, what is most fundamental aspect about blockchain technology that makes it so special? We might get two diverging opinions. One camp would argue that the blockchain technology is revolutionary and it would forever change the world, while the other camp would argue that blockchain technology is just incremental, merely being yet another data structure. We think that both camps are correct.

If we view blockchain as a distributed ledger data system, then this technology is incremental. It will have book keeping applications in large enterprises, but it will not fundamentally change how the enterprise would operate. What is revolutionary about blockchains is the aspect of a "trust machine" a permissionless and decentralized public chain could offer. In fact, we can do better than trust, we "verify" and not have to trust.

Why is trust important you may ask? If you look at traditional financial institutions, the credibility of an entity is typically determined by its power and scale. Entities deemed as "credit-worthy", such as governments, banks and large corporations, receive credit at a much lower interest rate than small businesses and individuals.

How is it possible for an pseudonym like "Satoshi Nakamoto", that nobody knows the identity of, been able to pull off a project like bitcoin with such an astronomical market cap? The answer is quite simple, a new dimension of credibility was found, and that was decentralization. In maths we trust, everything else we verify.

With this in mind, we see that it is the trustless, permissionless and decentralization nature of blockchains that is truly revolutionary. The protocol we are building in this project will leverage on this aspect to the maximum extent.

2.2. **A survey of other cross chain protocols.** There are currently two prevailing thoughts for cross chain protocols, atomic swap and wrapped tokens. Although atomic swaps are based on rigorous cryptographic proof, it's well known for it's underlying inefficiencies. Wrapped tokens are more popular, despite various concerns for centralization, it is currently seen as a workable solution. We will analyze its shortcomings and explain why we need a better protocol to serve as a critical infrastructure in the multi-chain world.

To understand how wrapped token work, we first consider the case of WBTC, wrapped bitcoin. In this case, the user is required to deposit their native bitcoin to a published address on the bitcoin network in exchange for a ERC 20 token on the ethereum network called WBTC. The value of WBTC is upheld by the perpetual promise of a centralized party that the user can always claim back their BTC with WBTC. The obvious downside to this approach is centralization, users must trust the centralized party, so this is not a trustless system.

Moreover, mapped tokens are unable to participate in the defi activities, making them less attractive to users. For example, there is no easy way for an owner of PETH (wrapped ethereum living on the polkadot network) to participate in liquidity mining on BSC; you will need native tokens on BSC to participate.

Composition of mapped tokens is also extremely inconvenient for the user. If a user owns DAI on BSC, let's call it BDAI, and she wish to change her BDAI token on BSC to a HDAI token on HECO. She would have to claim back her DAI from BSC to ethereum, and redeposit that DAI to HECO's ethereum bridge contract. If she went from BSC directly to HECO, she would effectively be depositing her BDAI to a HECO/BSC bridge. The asset she receive is technically HB-DAI, a mapped token of second order (deposit DAI to get BDAI, deposit BDAI again to get HB-DAI, and we could deposit HB-DAI again...). A centralized service for the user to change her HB-DAI to HDAI is the only known method to untangle the second order mess.

Some bridge protocols take a slightly less centralized approach to govern the reserve asset pool. We argue that these improvements are only incremental in nature. A quantum leap to the wrapped token approach is one where the user do not have to deposit at all, hence, a cross chain protocol for native tokens.

2.3. **Consensus.** We developed a proof of work main net to service our cross chain protocol, and we will briefly explain we believe it is necessary. The key pillar to our cross chain protocol is decentralization, and we believe that proof of work best serve our interest in this direction.

There are roughly three categories of network consensus for today's public chain,

(1) Proof of work

(2) Somewhat permissioned proof of stake (e.g. DPoS)
(3) Permissionless proof of stake (e.g. Eth 2.0)

Proof of work is by far the most tested consensus of the three, it's relatively easy to develop and it's the most decentralized. It has an obvious downside of being potentially extremely energy consuming. The DPoS consensus is perhaps the most popular amongst new projects requiring a main-net, but we view that as a quasi-permissionless protocol. Users who did not start with enough tokens, do not have any hope ever, to participate running the blockchain. A permissionless proof of stake consensus like Eth 2.0 is probably the most desirable. However, this family of consensus typically require a long period of research time and their reliability have not been thoroughly tested.

We want to develop the Class ZZ network as a public good, and we must begin with an empty ledger at the genesis block. Therefore, we view that in order to achieve the level of decentralization and the absence of pre-mining that we so endeavor, proof of work is the only type of consensus that would suit our need.

3. Te Waka Protocol

The Te Waka protocol have following desirable properties that no other protocols to date has been able to achieve,

(1) Trustless: Native token to native token, no user deposit or mapped tokens are involved.
(2) Universal: One protocol connects all smart contract enabled blockchains, unlike bridge protocols that only connect two chains at a time.
(3) Decentralized: every part of the protocol is verified on a permissionless public chain.

Currently, it is able to conduct cross train transactions with any two block chains such that

- Both blockchains are smart contract enabled
- Both blockchains support Secp256k1 addresses.

Most of the common blockchains we use today satisfy the above conditions. Please refer to section 10 for our plan to cover blockchains with non-Secp256k1 addresses.

Now, we explain in detail how this is achieved.

3.1. **Basic definitions.** Let XYZ be a blockchain network whose public key are points on the elliptic curve Secp256k1. The equation of this curve is given by

$y^2 = x^3 + 7$ over the finite field of $\mathbb{Z}_p$, where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$. Points on this curve are tuples of (x, y) where x, y are 64-digit hexadecimal numbers. Let G be a fixed point on this curve, G is fixed for all implementations of the ECC protocol Secp256k1.

If the private key k is securely chosen in advance, the public key P is obtained by the formula $P = kG$. If the elliptic curve and generator G remain unchanged on a different block chain, the same public / private key relationship also remain unchanged.

The following theorem summarises this idea.

Theorem 1. *If ABC and XYZ are two blockchain networks based on the same elliptic curve digital signature scheme (e.g. both uses Secp256k1), then if (P, k) are public-private key pairs on one blockchain, the pairing relationship continue to hold on the other.*

Definition 1. *Let $\mathsf{addr}(A, XYZ)$ denote the address on blockchain network XYZ corresponding to the public key A.*

Definition 2. *We use A_i to denote the 64-digit hexadecimal number i. For example,*

$$A_0 = \mathsf{00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000}$$

$$A_1 = \mathsf{00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000001}$$

$$A_2 = \mathsf{00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000002}$$

$$A_3 = \mathsf{00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000000\ 00000003}$$

Definition 3. *Let $\mathsf{trans}_{XYZ}(\Theta \to \Gamma, a)$ denote the transaction on XYZ network sent from Θ to Γ, for the amount of a tokens of XYZ, where Θ and Γ are valid addresses of XYZ. The quantity a may be abbreviated in future reference when it's clearly not necessary.*

Theorem 2. *Let ABC and XYZ are two blockchain networks based on the same elliptic curve. If a person is able to conduct the transaction*

$$\mathsf{trans}_{ABC}(\Theta \to ., .),$$

then she is also able to conduct the transcation

$$\mathsf{trans}_{XYZ}(\Theta \to ., .).$$

Hence, any assets sent to $\mathsf{addr}(\Theta, XYZ)$ is assumed to be under the same ownership as the person who made the transaction $\mathsf{trans}_{ABC}(\Theta \to ., .)$. This forms the basis of our cross chain exchange.

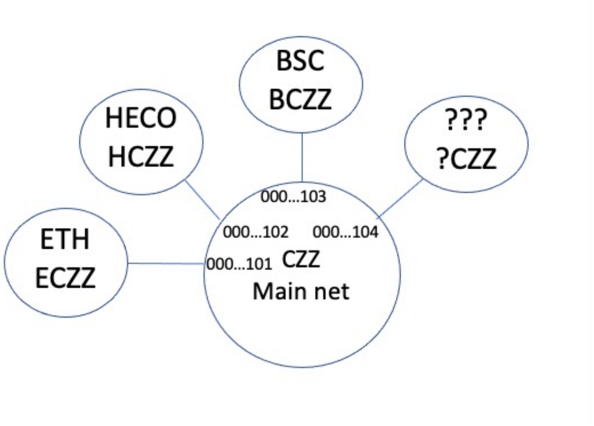

FIGURE 1. Relationship between CZZ mainnet tokens and mapped tokens

3.2. **Basic protocol.** The Te Waka protocol consist of three constituents,

- The Class ZZ main net, with native token CZZ.
- Mapped tokens of CZZ to other networks.
- Cross chain smart contract deployed on other networks.

The mapped tokens are constructed as follows,

Definition 4. *Let $A_0, A_1, ..., A_k$ be public keys, their corresponding private key is provably unknown to everybody, except a probability of $(k+1)2^{-256}$. This is known as the NUMS (nothing up my sleeves) principle. Let i be a small integer such that $i < 2^k$ (e.g. $k = 10$, so $i < 1024$), we call the addresses $\mathsf{addr}(A_i, .)$ "NUMS address" with security level $256 - k$.*

Assets under a NUMS address can be considered as a public good, because no single individual would have access to the private key.

(1) $\mathsf{addr}(A_{101}, CZZ)$: ECZZ tokens on ethereum are backed by CZZ tokens on this address.
(2) $\mathsf{addr}(A_{102}, CZZ)$: HCZZ tokens on HECO are backed by CZZ tokens on this address.
(3) $\mathsf{addr}(A_{103}, CZZ)$: BCZZ tokens on BSC are backed by CZZ tokens on this address.

This relationship is maintained by the following interplay between smart contract and PoW miners. The cross chain smart contract have three main functions: mint, burn and calling a decentralized exchange such as Uniswap.

- Mint: When main net CZZ tokens are sent to the NUMS address, the corresponding smart contract will execute mint. For example, the main net transaction

$$\mathsf{trans}(\mathsf{addr}(\phi, CZZ) \to \mathsf{addr}(A_{101}, CZZ))$$

will result in

$$\mathsf{trans}((\mathsf{SmartContract.Mint}, ECZZ) \to \mathsf{addr}(\phi, ECZZ))$$

- Burn: When the user burn mapped tokens via the cross chain smart contract, it will result in a negative block reward to the corresponding NUMS address, administrated by PoW miners. For example, the ethereum transaction

  $$\mathsf{trans}(\mathsf{Addr}(\phi, ECZZ, a) \to (\mathsf{SmartContract.Burn}, ECZZ, a))$$

  will result in an $-a$ block reward for the corresponding address $\mathsf{addr}(A_{101})$.
- Te Waka: When the user burn mapped tokens via the cross chain smart contract, and indicates that she want to receive mapped tokens of another blockchain, it will result in a negative block reward to the starting NUMS address, and a positive block reward to the destination NUMS address, administrated by PoW miners.

  For example, the ethereum transaction

  $$\mathsf{trans}(\mathsf{Addr}(\phi, ECZZ, a) \to (\mathsf{SmartContract.Burn.ToBSC}, ECZZ, a))$$

  will result in
  - a units of $ECZZ$ being burnt on ethereum.
  - $-a$ block reward for $\mathsf{addr}(A_{101})$.
  - a block reward for $\mathsf{addr}(A_{103})$.
  - a units of $BCZZ$ being minted, and sent to the user via the transaction

    $$\mathsf{trans}((\mathsf{SmartContract.Mint}, BCZZ) \to \mathsf{addr}(\phi, BCZZ))$$

3.3. **Main protocol.** Next, we analyze what happens when someone try to exchange a general token from one network for a general token of another network. Specifically we will look at the example of buying CAKE on BSC with MDX on HECO.

The user will have the following experience,

- User initiate the transaction: Send MDX to the cross chain smart contract on HECO, with instructions to buy CAKE on BSC.
- Approximately 1 minute later, she will receive CAKE on BSC.

Now we explain how this is done in detail.

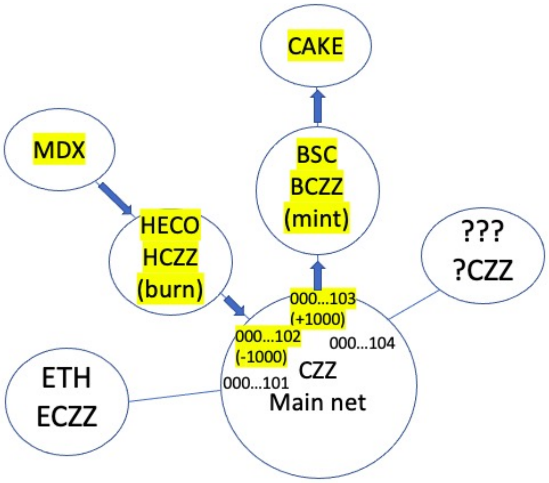

FIGURE 2. Route taken when exchanging MDX.HECO for CAKE.BSC

(1) When MDX is received by the smart contract on HECO, it will call a swap contract on HECO to change MDX to HCZZ.
(2) Burn the resulting HCZZ, and broadcast to Class ZZ main net.
(3) In the next CZZ block, PoW miners will first verify the HCZZ tokens has indeed been burnt. Suppose 1000 HCZZ have been burnt. Then, it will give $\mathsf{Addr}(A_{102})$ a block reward of -1000; and $\mathsf{Addr}(A_{103})$ a block reward of $+1000$.
(4) The smart contract on BSC will mint 1000 units of BCZZ.
(5) The smart contract will call a swap contract on BSC to buy CAKE with BCZZ, and send to the user.

It's worth noting that in theory it's the user's responsibility to broadcast the burn transaction to the Class ZZ main net. However, we will release an open source API for the third party app developers to integrate into their software for better user experience.

We have presented the basic design of the world's first decentralized cross chain protocol for native tokens. In the following sections, we will discuss some potential issues with this design, and the possible ramifications we could make.

4. INSURANCE CONTRACT

The PoW phase of the Te Waka protocol can take around 1 minute to complete, this is the time required for PoW miners to produce a block on the CZZ network, plus overhead. A possible mitigation to this problem is the deployment of our

Algorithm 1: Alice wish to exchange MDX.HECO for CAKE.BSC

```
Heco chain
User: $\text{trans}_{HECO}(\text{addr}(\phi, MDX) \rightarrow \text{SmartContract.Heco})$
SmartContract.Heco: Call swap contract change MDX to HCZZ
SmartContract.Heco: Burn HCZZ
Broadcast: the above transaction to Class ZZ mainnet.

Class ZZ
Verify: HCZZ has indeed been burnt
Next block: $+a$ block reward for $\text{Addr}(A_{103})$
Next block: $-a$ block reward for $\text{Addr}(A_{102})$

BSC chain
SmartContract.BSC: Mint $a$ units of BCZZ
SmartContract.BSC: Call swap contract change BCZZ to CAKE
SmartContract.BSC: Sent CAKE to the user
User: Receives CAKE
```

insurance contract, it is aimed to significantly reduce exchange rate volatility for the user, in exchange for a fee.

There are two facades to the insurance contract. For the insurer: Any CZZ token holder can deposit their mapped CZZ tokens to the insurance contract. Fees charged (in CZZ mapped tokens) will be distributed to the insurer on a pro-rate basis. We expect the amount of CZZ deposited will be determined by the yield market.

For the insuree: Given that there are mapped tokens of CZZ already deposited in the insurance contract, this contract can pre-purchase the final token almost immediately. Let's suppose the user wanted to exchange MDX.HECO for CAKE.BSC under the insurance contract. The exact steps are as follows,

Time = 0

- User deposit MDX to cross chain contract
- Cross chain contract (on Heco) convert MDX to HCZZ and execute burn
- Insurance contract (on BSC) buys CAKE with its own BCZZ, and hold.

Time = 1

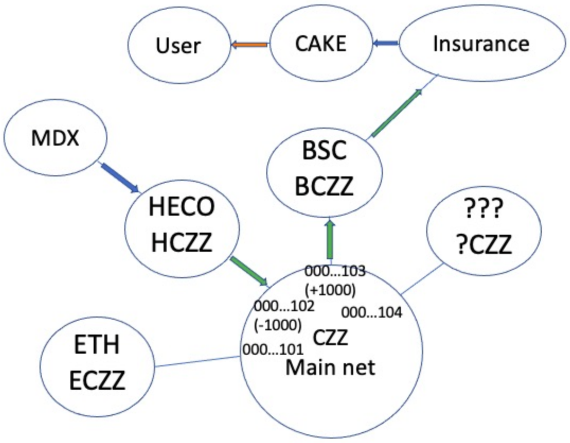

FIGURE 3. Insurance contract order of execution. T=0 Execute blue arrow; T=1 Execute green arrow; T=2 Execute orange arrow.

- PoW miners verify HCZZ has indeed been burnt
- On the next block, miners give $+a$ CZZ to $\mathsf{Addr}(A_{103})$ and $-a$ CZZ to $\mathsf{Addr}(A_{102})$ as block reward.
- BSC cross chain contract mint a units of BCZZ and send them to the insurance contract.

Time = 2

- Insurance contract releases CAKE to user after receiving the BCZZ from the cross chain contract.

The net effect is that we can estimate the exchange rate with very high precision, as MDX - HCZZ and BCZZ - CAKE are executed almost at the same time.

5. LIQUIDITY

In order for the Te Waka protocol to work, we need to ensure pairs such as ETH/ECZZ, HT/HCZZ and BNB/BCZZ have enough liquidity. On the other hand, once we have a basic level of liquidity to run Te Waka, we actually have a liquidity positive feed back loop. This is because by using the Te Waka protocol, users are also providing liquidity themselves. We may resort to liquidity mining in the beginning to get the system running, please stay tuned for more details.

6. COMMUNICATION MODEL

A "blockchain network" is a collection of nodes agreeing to update their chain according a predetermined consensus. The state which a chain exist locally at each

Algorithm 2: Alice wish to exchange MDX.HECO for CAKE.BSC with insurance contract

```
Time = 0
User: $\mathsf{trans}_{HECO}(\mathsf{addr}(\phi, MDX) \rightarrow \mathsf{SmartContract.Heco(Insurance)})$
SmartContract.Heco: Call swap contract change MDX to HCZZ
SmartContract.Heco: Burn HCZZ
InsuranceContract.BSC: Buy CAKE with BCZZ
Broadcast: the above transaction to Class ZZ mainnet and insurance
  contract on BSC.

Time = 1
Verify: HCZZ has indeed been burnt
Next block: $+a$ block reward for $\mathsf{Addr}(A_{103})$
Next block: $-a$ block reward for $\mathsf{Addr}(A_{102})$
SmartContract.BSC: Mint $a$ units of BCZZ
SmartContract.BSC: Send $a$ units of BCZZ to insurance contract

Time = 2
InsuranceContract.BSC: Send CAKE to user after receiving BCZZ
User: Receives CAKE
```

node is called a view. A network is said to be "permissionless" if anyone can join and leave at any time. Upon joining a network, there are no central authority to say which node is preferred over another. Since each node only see their local view, due to network delays, the views between nodes may be different for the most recent blocks. Hence, the network in general will be at an asynchronous state, consistency only happens before the last Λ blocks, where Λ is a natural number.

Definition 5. *Let* $\mathsf{view}(\mathsf{chain}(t), i))$ *denote the view of* chain *at time t, from node i. We say a network is "consistent" if there exist $\lambda > 0$, independent of t, such that* $\mathsf{view}(\mathsf{chain}(t-\lambda), i)$ *is a constant function with respect to i.*

Definition 6. *Let* $\mathsf{TX}(t, j)$ *be valid transactions presented to an honest node j at time t. We say a network has "liveness" if there exist $\omega > 0$, independent of t, such that* $\mathsf{TX}(t, i) \subseteq \mathsf{view}(\mathsf{chain}(t+\omega), i)$ *for all honest nodes i.*

The aforementioned security requirements must be guaranteed with overwhelming probability. Let H be the current network hash rate, we require that for any

$\varepsilon > 0$, there exists D large enough so that the time between each block update $S(D, H)$

$$\mathbb{P}(S(D, H) < \lambda) < \varepsilon.$$

Blocks are generally difficult to mine, because we need a kill-time mechanism to give enough time for nodes around the world to synchronize their block sequences. In proof of work, this is achieved by increasing mining difficulty D.

With the development of new radio frequencies and massive MIMO and beamforming, especially the emergence of 5G mobile networks, we can expect a quantum leap in the direction of high bandwith and low latency. For 5G networks, ITU-R have defined three main types of usage scenario: Enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC). 5G networks achieve much higher data rates than current cable internet, and 100 times faster than 4G LTE, up to 10 gigabits per second (Gbits/s). In addition, the network latency will be much lower, below 1 millisecond (ms), compared with 30 - 70 ms for 4G. In this scenario, we plan to substantially increase block size provided that consistency is not compromised in the future.

7. Sharding

The low throughput of the PoW main net could pose a potential problem for the future, particularly if Te Waka gains popularity. On the other hand, the type of transactions that Te Waka is processing, the validity of input has already been confirmed on another blockchain. Hence, the validity of a cross chain transaction does not depend on the finality of previous transactions on the CZZ main net. Leveraging on this property, we give our design of a sharded blockchain, a network with unbounded scalability.

Each shard will consist of 7 - 22 staking nodes, running it's own blockchain. The shards will be processing cross chain transactions, while CZZ-native transactions are still processed by the PoW network. It is estimated that each shard will take approximately 2 - 3 seconds, while each PoW block will take 40 seconds. Therefore, shards will need to report their network state, and the Merkle root of each block to the PoW blockchain every 20 shard blocks. Miners of the PoW network will verify the correctness of this information, and store it in the next PoW block.

In this design, users would hold CZZ tokens on the main net as a store of value, while making transactions using mainly mapped tokens (such as ECZZ, HCZZ, BCZZ etc) which will leverage on the throughput of other networks, and shards of the Class ZZ network. Due to the parallel nature of cross chain transactions, the Class ZZ main net can simply increase the number of shards to provide for better throughput. This makes Class ZZ to be the first blockchain network to have unbounded scalability.

We will give a more detailed design in a later release.

8. Token economy

The main use case of the Class ZZ blockchain is in the cross chain protocol of native tokens. Specifically, each time Te Waka is used, users will be charged a commission of 0.1%, which will be burnt. We will ensure the entire community of CZZ token holders benefit from the mass adoption of the Te Waka protocol.

In the following subsections, we will explore the possible scope of application, and other deflationary mechanisms such as yield farming and staking.

8.1. **Applications.** Our innovative cross chain technology has enabled us to take a unique position in our approach to the market. Since Te Waka protocol is trustless, decentralized and universal, our community operate on a basis of project neutrality. Our sole purpose is to empower other DeFi projects to attain a further reach, through the integration of Te Waka protocol. We want to be complimentary, and not competition to vast majority of projects in the DeFi space.

The potential use cases of Te Waka include but not limited to,

(1) Decentralized exchanges can offer their users to trade tokens of another blockchain.
(2) Arbitrageurs can now arbitrage cross chain. For example, if leading rate on AAVE could be 4%, while a yield product on BSC is paying 8%, arbitrageurs could borrow on one network to farm on the other.
(3) Providing the infrastructure foundation for existing projects to become multichain.
(4) Provide additional throughput to existing networks (e.g. cross layer 2 support for ethereum).

There are two big weaknesses with layer 2 scaling solutions,

- Long withdrawal waiting time when going back to layer 1
- Lack of a protocol to move assets across different layer 2 networks

Te Waka protocol can provide support for cross network transactions between layer 2's on ethereum. Under the Te Waka protocol, waiting time for users to move assets between optimistic rollup layer 2's can be reduced to minutes, as opposed to days. This will help the ethereum community in their transition to layer 2 from layer 1.

8.2. **Deflationary mechanisms.** The Class ZZ ecosystem consists of three more deflationary measures. These are,

- Liquidity mining
- Insurance contract
- Staking to run a shard

Participants in these activities are expected to be yield seekers, and the amount locked up will be dictated by the yield market.

8.3. **Quick facts.** We list some basic facts about the CZZ token:

- Consensus: Proof of Work
- Mining algorithm: Bora Bora
- Block interval: 40s
- Max block size: 8m
- Min mining difficulty: 1 mh/s
- Initial block reward: 1000 CZZ
- Of the 1000 CZZ, 800 goes to the miner, 200 goes to the community reward pool (see liquidity mining section)
- Reward decay: Halves once every 1 million blocks
- Total issuance: 2 billion
- Premine: 0

The CZZ network will produce approximately 1 million blocks a year.

Here's the quantity of CZZ in total supply, including tokens going to the community pool

Year	CZZ supply (billions)
1	1.00
2	1.50
3	1.75
...	...
∞	2.00

9. Mining

9.1. **Asic resistance.** The Class ZZ community have invented our own hash algorithm Bora Bora. The main purpose of inventing a new hash for our PoW is to prevent a 51% attack from miners of other PoW projects. On the other hand, if Bora Bora is extremely ASIC resistant, we will be at risk for a 51% attack from GPU miners. Therefore, our view is that we want to a hash algorithm that will make ASIC production more expensive per hash than say Sha256d, but we do not preclude them from our ecosystem.

Previous attempts at coming up with ASIC resistant hash functions have generally been futile [insert citation]. Two mechanisms that people came up with are, introduction of

(1) Memory hard functions, and
(2) Bandwidth hard functions.

Their common observation is that the common Sha256d ASIC is just a hard coded hash calculator, incapable of any memory storage nor does it offer any data bandwidth. Hence, by introducing a function with one of these components, any architect based on the SHA256d ASIC will be deemed useless.

However, in the advent of lucrative financial incentives, it turned out that through a myriad of innovation by ASIC manufacturers have completely crushed these obstacles. The Bitmain E3 miner for example, placed a ring of DDR3 SDRAM chips around the actual ASIC. This enabled the table lookup step in EThash to be executed with significant efficiency over the GPU. Had Ethereum not gone PoS, the next generation of DDR chips will render Ethash's ASIC resistance completely useless.

We ask the question, is there a fundamental mechanism that we can leverage upon, to make our hash mechanism ASIC resistant over time? A straightforward approach is simply change your hash in regular block intervals. This was precisely what Monero did in order to fend off the onslaught of Bitmain X3. There are two downsides with Monero's approach,

(1) Rules of mining feels incomplete, with human intervention at regular intervals.
(2) Monero core team (or whoever with significant sway on the vote) can never prove that they have in fact pre-manufactured an ASIC to mine the new hash.

Another approach is to have a set of pre-determined hash functions and randomly recombine between them, and this was more or less what Dash and Raven

Algorithm 3: Ordinary mining algorithm

```
Build block header from: $\mathfrak{h}_{-1}, \mathsf{r}$
Initialize: $mined = false$
while $mined == false$ do
    $v = concat(\mathfrak{h}_{-1}, \mathsf{r}, \eta)$
    $h = \mathcal{H}(v)$
    $\eta = \eta + 1$
    if $h < 1/D$ then
        $mined = true$
        broadcast block
```

tried to do. However, Dash was recombining from a set of 11 different hashes, which makes it possible for ASIC manufacturers to simply exhaust all options - which was exactly how Bitmain D3 was built.

9.2. **Data-growth regime.** We came up with the following for our ASIC resistance,

(1) There exist a large pool of potential mining algorithms that is unfeasible to hard code each of them to an ASIC chip.
(2) The mining algorithm relies on a memory or bandwidth bottleneck that will be very expensive for ASIC production.
(3) The hurdle put forward for ASIC resistance (such as memory hardness or bandwidth hardness) must grow with respect to technological development.

In the following subsections, we describe in detail

(1) How to generate a large space of collision resistant hash functions
(2) How to generate new hash with increasing memory and bandwidth complication.

9.2.1. *Generating hash space.* Classically, mining algorithms would loop through the nonce η and repeatedly compute $h = \mathcal{H}(v(\eta))$, until $h < 1/D$. Let G be a large group (e.g. GL_{2048}), and $\rho_V : G \to \mathrm{End}(V, V)$ be a homomorphism from G to the endomorphism group of V, we call ρ the representation of G. That is, for every $g \in G$, $\rho_V(g)$ is a linear map from V to V. Since the hash function $\mathcal{H}$ can be applied to any element in V, we can compute instead, the modified hash function, $\mathcal{H}(\rho_V(g) * v(\eta))$.

Algorithm 4: Bora Bora mining algorithm

```
Running hash
Build block header from: $\mathfrak{h}_{-1}, \mathsf{r}$
Pick any $g \in GL_{2048}$
So $V = \mathbb{Z}_2^{2048}$ and $\rho_V(g) \in \mathsf{End}(\mathbb{Z}_2^{2048}, \mathbb{Z}_2^{2048})$
Initialize: $mined = false$
while $mined == false$ do
    $tmpv = concat(\mathfrak{h}_{-1}, \mathsf{r}, \eta)$
    $v = concat(tmpv, ..., tmpv)$
    (so $v \in V$)
    $h = \mathcal{H}(\rho(g_{16}) * ... * \rho(g_3) * (\rho(g_2) * (\rho(g_1) * v_1 + v_2) + v_3)... + v_{16})$
    (Note, $\rho_V(g) * v \in V$, and $\mathcal{H}$ can be applied to any element in $V$)
    $\eta = \eta + 1$
    if $h < 1/D$ then
        $mined = true$
        broadcast block
```

We can make the above algorithm more complicated by randomly generating 16 such group elements $g_1, ..., g_1$6 and 16 vectors $v_1, ..., v_{16}$ on 2^{2048}. Rather than looping through $\mathcal{H}(\rho_V(g) * v)$, we compute instead

$$\mathcal{H}(\rho(g_{16}) * ... * \rho(g_3) * (\rho(g_2) * (\rho(g_1) * v_1 + v_2) + v_3)... + v_{16}).$$

Computing this hash is both memory hard and bandwidth hard, as the above matrix multiplications must be repeatedly performed for every iteration of the hash computation.

Moreover, every time we change a group element g to g', we get a completely different hash algorithm, where $\mathcal{H}(\rho_V(g)*v(\eta))$ is replaced with $\mathcal{H}(\rho_V(g')*v(\eta))$. Since $\mathcal{H}$ is assumed to be collision resistant, we have a one-to-one correspondence between a hash algorithm and a group element. With this in mind, we can make the matrix arbitrarily big to catch up to technological innovation.

Leveraging on the above results, CZZ plan to double the dataset size once every 12 months. For ASICs to be competitive in computing Bora Bora, SRAM will be a key component. Periodically doubling the dataset size would imply that ASIC manufacturers are forced to use the most advanced SRAM, thereby rendering any

ASICs prohibitively expensive.

10. Digital signature extension

One limitation of the Te Waka protocol described in section 3 is the inability to interoperate with blockchains whose digital signature is not based on the elliptic curve Secp256k1. Prominent examples include, Libra (Curve25519), EOS (include Secp256r1), Monero (Ed25519). Therefore, it's in the interest of the Class ZZ community, for our network be able to interoperate with these other blockchains.

On the other hand, there is no reason why PoW miners on the Class ZZ network cannot verify more than one type of digital signature. Recall that to get from public key to the bitcoin address, it was simply a repeated composition of the functions $\mathsf{ripemd160}$ and $\mathsf{sha256d}$. We could create a new hash, by a repeated composition with a phase translation. Class ZZ addresses generated by points of other elliptic curves can be indexed by such transition, providing miners with enough cryptographic data to verify their relationship.

Specifically, we will use the following phase translation parameters. Let q be a large prime,

Curve	Phase translation
Secp256k1	0
Curve25519	$\mathsf{sha256d}(seed1)$
Secp256r1	$\mathsf{sha256d}(seed2)$
Ed25519	$\mathsf{sha256d}(seed3)$

The exact values of $seed1$ to $seed3$ is yet to be determined. This would allow the digital signatures of CZZ addresses to span over multiple elliptic curves.

Next, we briefly illustrate how the Te Waka protocol works over multiple elliptic curves. We would essentially inherit the same protocol rules as described in section 3. We only need to allocate an extra NUMS address for each elliptic curve. For example with EOS, in order to support both Secp256k1 and Secp256r1, we will need to allocate two NUMS addresses for the one blockchain.

11. Post-quantum cryptography

It is well known from the algorithm by Peter Shor[8] that both prime number factorization and discrete logarithm can be solved in polynomial time using a quantum computer. Therefore, cryptographic protocols like ECC or RSA, whose security rely on the difficulty in solving such problems, are no longer safe[7]. Other quantum algorithms such as [3], would offer significant computation advantage on the hash function.

On the other hand, we are also in process toward developing industrial standards for post-quantum cryptographic scheme [6]. It is expected that by 2030, post quantum encryption would replace RSA and ECC in the most basic layer of encryption protocols. Therefore, it is imperative for the blockchain community to start planning for such a transition.

In this section, we will briefly talk about the outstanding issues for post quantum encryption, and particularly, how these issue may impact the blockchain community.

11.1. Misconceptions of quantum supremacy.

People often equivocate quantum computers as simply “massively parallel” classical computers. Since a quantum bit can occupy both 0 and 1 simultaneously, a n-bit quantum computer can be in 2^n states at the same time, hence able to compute NP-complete problems extremely fast. Unfortunately this is a misconception, as measuring a quantum state would destroy all information of the quantum system that was not measured.

Success of the Shor’s algorithm was particularly “misleading” because only 1 solution was required in the final output, and the issue of quantum state annihilation was quietly suppressed. If the underlying problem had multiple solutions, each time we measure the computer’s quantum state, it would only output one candidate solution x, with probability proportional to the wave function amplitude a_x. In the special case of factoring, the one solution was all that you needed.

Let BQP be the class of problems solvable in polynomial time by a quantum computer. It has been shown in [2] that $\mathsf{NP} \not\subset \mathsf{BQP}$. Particularly, they showed that any quantum algorithm that searches an unordered database of N items for a single labeled item, must query the database $O(\sqrt{N})$ times. If we interpret the space of 2^n possible assignments to a Boolean formula φ as a database, and the satisfying assignments of φ as labeled items, then the result of [2] would imply that any quantum algorithm need at least $O(2^{N/2})$ steps to find a satisfying assignment, with high probability. Hence, there is no “brute force” quantum algorithm to solve NP-complete problems in polynomial time.

11.2. **Types of post quantum algorithms.** Post quantum cryptography is the study of crypto systems running on classical computers that are secure against a quantum adversary. NIST has already start a process to establish industry standards of such systems, and is currently reviewing the second round of submissions. It is not the intent of this project to conduct any original research in the field of post quantum cryptography. Rather, the Class ZZ community will be assessing which of the published post quantum algorithm is best suited for blockchain applications.

There are five different approaches, to date

- Lattice based
- Multivariate based
- Hash based
- Code based
- Supersingular isogeny

We won’t go through the technical details of each one, and we refer the reader to [6] for more information. What we will instead focus on are current outstanding issues and how their impact on blockchain applications.

A transaction would typically consist of

- Input: 1 address
- Output: several addresses
- Digital signature

If the size of each transaction is big, the number of transactions we can fit in a block would become severely limited. In some cases where public key size are close to 1 mb, you can only fit 1 transaction per bitcoin block, rendering bitcoin tps to the order of 10 minutes per transaction. Since the address length is correlated to the size of public key, and there is an inverse correlation between the public key and signature size, it is difficult to simultaneously get the two to be in reasonable size.

On the other hand, with the advances in communication technology such as 5G, it is possible that our network speed could be 100x faster than what’s currently available today. Perhaps this would allow us to make some compromises on the key / signature size, and simply settle for larger blocks in the future.

Supersingular isogenies is one example where both key and signature size are smaller than their peers, and it has advantages in providing forward secrecy. However it suffers from the drawback of long computation time. This issue may be overcome in the future by ever advancing specialized hardware such as [4].

11.3. **Class ZZ.** Our approach to post quantum encryption are as follows,

- 2020 - 2025: Observation phase. The Class ZZ community will be actively engaged with the post quantum research. A post quantum test net may be developed during this period.
- 2026 - 2028: Class ZZ will hard fork to a version with digital signature extension (see section 12) of a post quantum scheme. There will a smooth transition period of CZZ going from ECC based encryption to post quantum encryption.
- 2028 - 2030: Leveraging on Te Waka protocol, the Class ZZ network will provide post quantum address extension to traditional blockchains such as bitcoin. As the quantum threat become ever close to reality, we hope to provide the bridge for the entire blockchain community to transition to post quantum blockchain.

12. Road map and future directions

- September 2019: Main-net launched and genesis block mined
- Entire year of 2020: Research and coding...
- March 2021: Te Waka launched on test net, supporting Ethereum, HECO and BSC
- April 2021: Te Waka will launch on main net after the Maui fork, supporting additionally, Okchain and Solana
- April 2021: Insurance contract deployed on test net, and is expected to be deployed on the main net.
- May 2021: Te Waka will support additionally, Tron and Polkadot
- June 2021: Launch of sharding test net
- July 2021: Launch of sharding main net, supporting additionally, roll-up networks of Ethereum layer 2
- Sometime in 2021: Go to the moon!

References

[1] Anonymous. How to make a mint: The cryptography of anonymous electronic cash. *URL https://groups.csail.mit.edu/mac/classes/6.805/articles/money/nsamint/nsamint.htma*, 1996.

[2] B. G. V. U. Bennett C, Bernstein E. Strengths and weaknesses of quantum computing. In *https://arxiv.org/abs/quant-ph/9701001*.

[3] L. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th Ann ACM Symposium on Theory of Computing, pp 212 - 219*.

[4] M.-K. M. Koziel B, Azarderakhsh R. Fast hardware architectures for supersingular isogeny diffie-hellman key exchange on fpga. *https://eprint.iacr.org/2016/1044.pdf*, 2016.

[5] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *URL http://bitcoin.org/bitcoin.pdf*, 2008.

[6] NIST. Post quantum cryptography - round 2 submissions. In *https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions*.

[7] Z. C. Proos J. Shor's discrete logarithm quantum algorithm for elliptic curves. In *https://arxiv.org/abs/quant-ph/0301141*.

[8] P. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proc 35th IEEE Symposium on Foundations of Computer Science, pp 124 - 134*.