

# FrigateChain Technical White Paper v1

## Abstract

FrigateChain is a new blockchain architecture focusing to further innovate horizontal structure of decentralized applications. FrigateChain creates a master network node which serves as a platform where independent networks can be created. FrigateChain software provides millions of transaction per second using grid computing technology, flexibility due to total customization by creating a platform where child mainnet can attach to FrigateChain, and security and total transparency by implementing open block centralization.



# Table of Contents

---

<b>Background</b> .....	3
<b>Parallel Computation</b> .....	3
– Limitations	
– Workaround	
<b>Grid Computing</b> .....	4
<b>Independent Network</b> .....	5
– Transaction Process	
– Verification and Transaction Data	
– Data Structure	
<b>Independent Server</b> .....	7
– Developing Tools	
<b>Action &amp; Handler</b> .....	8
– Standard Format	
– Custom Format	
– Action Format Proposal	
– Handler	
<b>Transparency</b> .....	9
– Open Block Centralization	
– Watchdog	
<b>Workers</b> .....	10
– Efficiency	
– Block Reward	
<b>Conclusion</b> .....	11



# Background

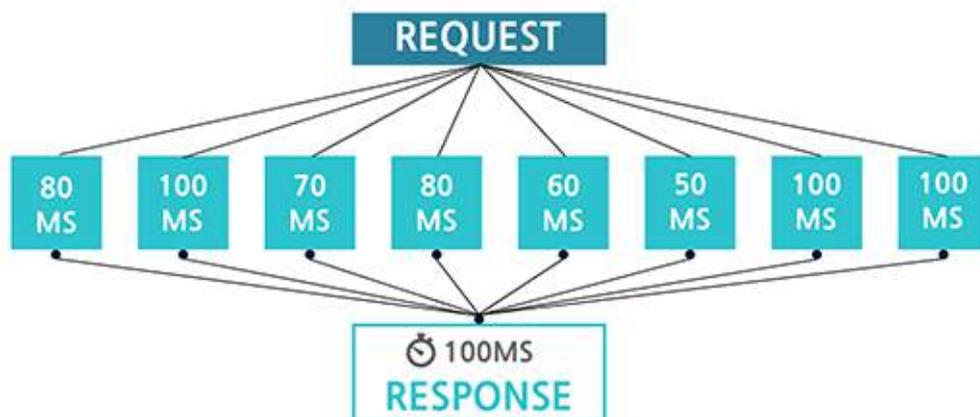
---

Blockchain technology was first introduced with the launch of Bitcoin in 2008 by Satoshi Nakamoto. The blockchain technology is now popular due to decentralization which gave users some controls over the company. While blockchain is a groundbreaking technology which many entrepreneurs try to create decentralized applications for their ventures, there are some issues that blockchain technology has not been resolved yet. Some of the major issues are decentralization applications' complete reliance on blockchain network and slow transaction speed due to the rise in popularity and limited computational capacity.

## Parallel Computation

---

There are other blockchain architects that use parallel computation. Parallel computation assigned set amount of work to each individual participants and the task is only complete when every participant involved in the task finishes their own job.



### Limitations

---

The downside of parallel computation method is that the slowest person will be the bottleneck of the task and there is a change where anyone participant can refuse to work thus making the whole task ineffective. Another downside of the parallel computing is that there is a fundamental limit to how small a task can be broken down into. This characteristic reinforces the bottleneck problem since the slowest participant must be given a specific amount of data to be processed.



## Workaround

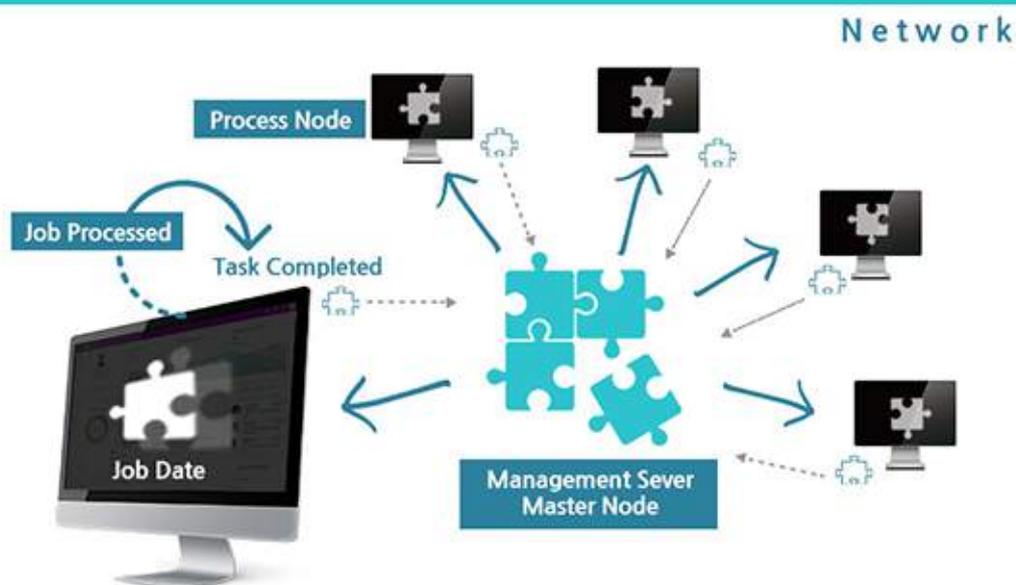
---

FrigateChain addresses issues caused by parallel computing by utilizing grid computing method. In the case of grid computing, a group of participant works together to do a task. The main difference is that in parallel computing, participants are giving certain chunks to process opposed to in grid computing, a group of participants is working on the whole task together. In grid computing, all the participants come together and create a supercomputer that will solve a task.

# Grid Computing

---

FrigateChain introduces a new way of consensus mechanism utilizing grid computing. When there is a transaction request, it will be sent to FrigateChain whether the request was produced in the child mainnet or main network. All the transactions are processed in FrigateChain by workers. FrigateChain will break down block data into smaller chunks and distribute to a group of workers to be mined simultaneously.

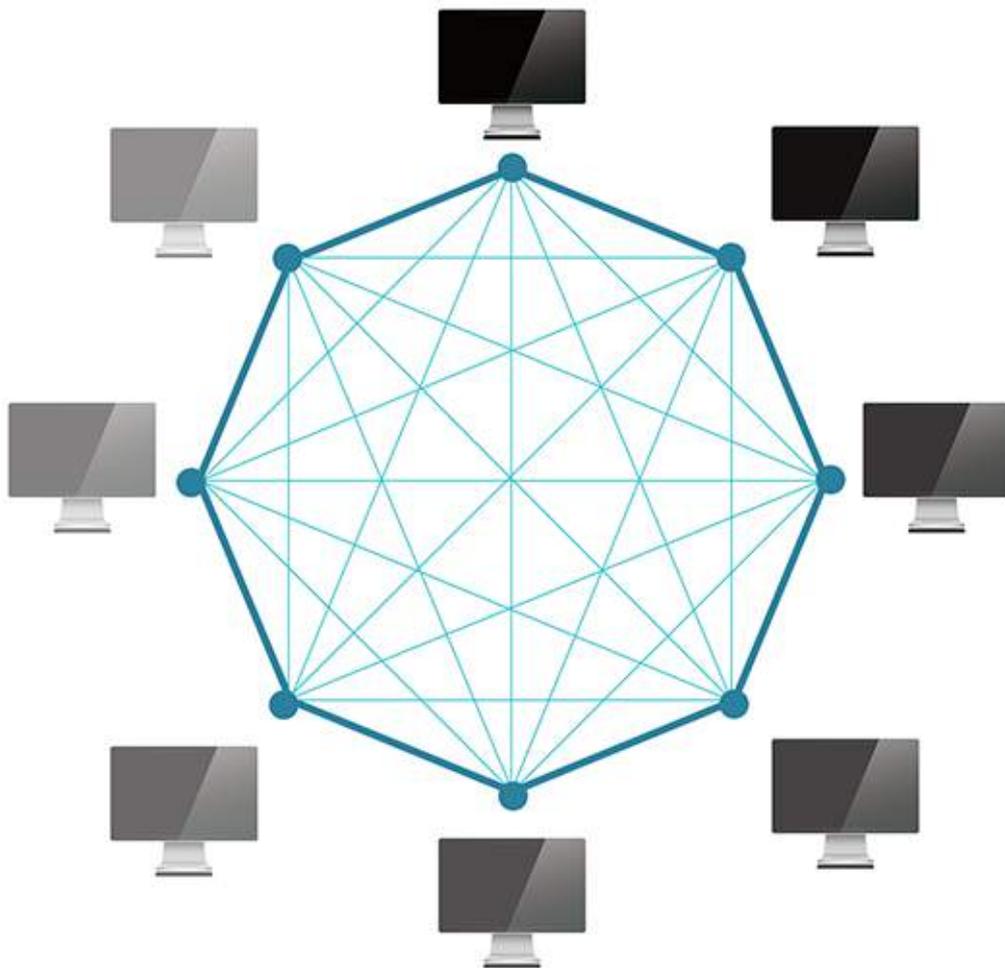


# Independent Network

---

Every network that is attached to FrigateChain has its' own blockchain network. Due to individualized blockchain network, entrepreneurs can now have full control of their network and customize blockchain to fit their needs. By doing so, each network only contains data related to its field, resulting in smaller block information which will drastically reduce the search time.

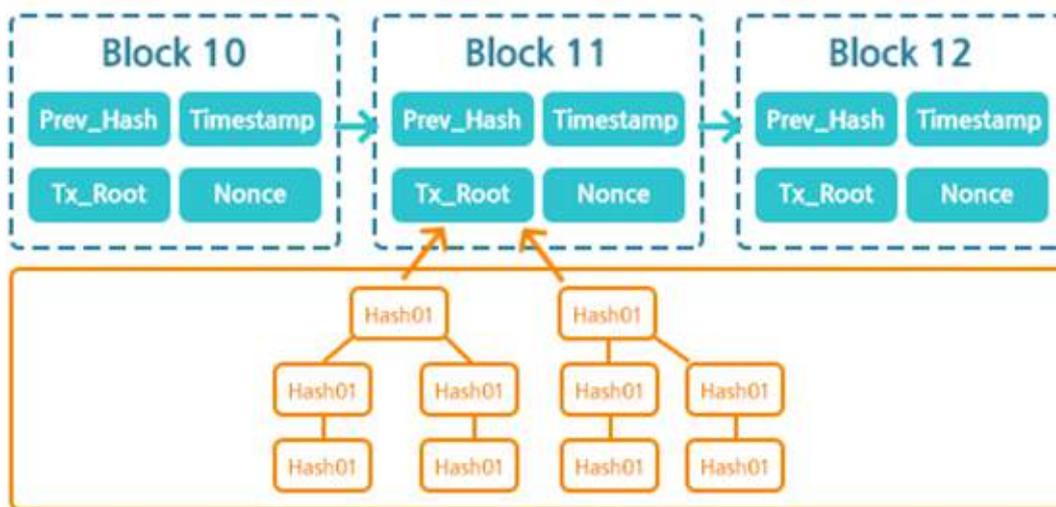
While FrigateChain promotes the individuality of each network, to ensure that all the networks under FrigateChain can interact without a problem, there will be some set of standards that child mainnet will have to follow. One obvious example would be the coin standard.



## Transaction Process

---

In order for transactions to be processed, all transactions will go to FrigateChain to be processed. Once FrigateChain receives a request for the transaction confirmations, it will assign workers to process and confirm all the requests. Once the transactions are processed, it will save a hash of that transaction to its blockchain and send back information to respective child mainnet for child mainnet to include in their respective blockchain. All transaction processing and block productions are done in FrigateChain.



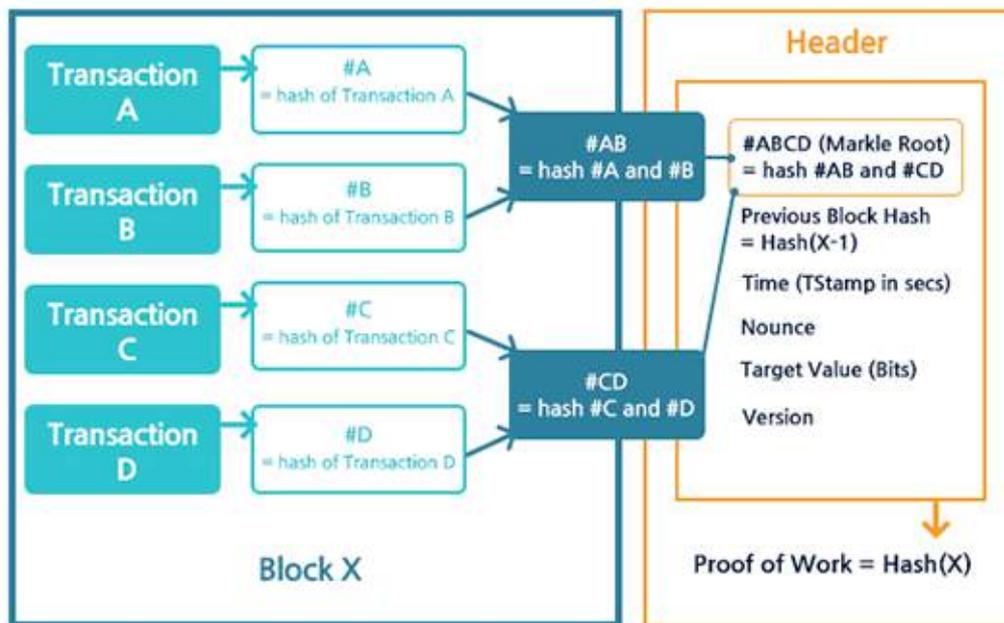
## Verification and Transaction Data

---

Every transaction contains a header and contents. The header will contain the hash of previous transactions to ensure the security and contents will include actions. When FrigateChain receives a transaction from child mainnet to be processed, FrigateChain will first check the header to see whether hash in the header matches with the previous listing in the FrigateChain block. Once FrigateChain verifies that the transaction request is valid, it will then process the data and save the hash of transactions into its own block before sending to child mainnet. Essentially, FrigateChain's blockchain act as logs for all transactions performed in child FrigateChain to cross reference for any irregularities.

Some openings for data modification are present due to the nature of data transferring between FrigateChain and child mainnet. Countermeasures for these opening will be addressed later on with the introduction of Watchdogs.





## Data Structure

Due to the nature of having an independent network, the roles of blockchain network in FrigateChain differs from ones in child mainnets. In the FrigateChain, blocks only hold information regarding child mainnets' transaction hashes. This is used for confirmations purposes. On the other hand, each child mainnets contain all the transactions and actions recorded similar to traditional blockchain information.

## Independent Server

FrigateChain provides tools to create mainnets but does not provide hardware. This means that any developers who want to create mainnet via FrigateChain would have to provide their own server to run their nodes and store block information.

## Developing Tools

While developers have to provide their own server, they will be provided with a server installer to ensure easy and fast integration to FrigateChain. Along with server installer, FrigateChain will provide block explorer and action handler. Server Installer and SDK are compatible on multiple operating systems including Windows, Linux, and macOS.

# Action & Handler

---

In order to communicate within or between blockchain, one can send structured actions to each other and set up a script to handle when an action is received. In FrigateChain, there are 2 types of format that action can take.

## Standard Format

---

In order to ensure smooth communication between child mainnet and FrigateChain, a set standard will be given to developers to structure their actions so that it will be recognized universally. There will be predetermined protocol names that will be used for common purposes.

## Custom Format

---

Aside from predetermined protocols, developers can create any other names for the protocol and how to handle each protocol. This way, developers can freely add features to their mainnet and create actions and handlers to fit their needs.

## Action Format Proposal

---

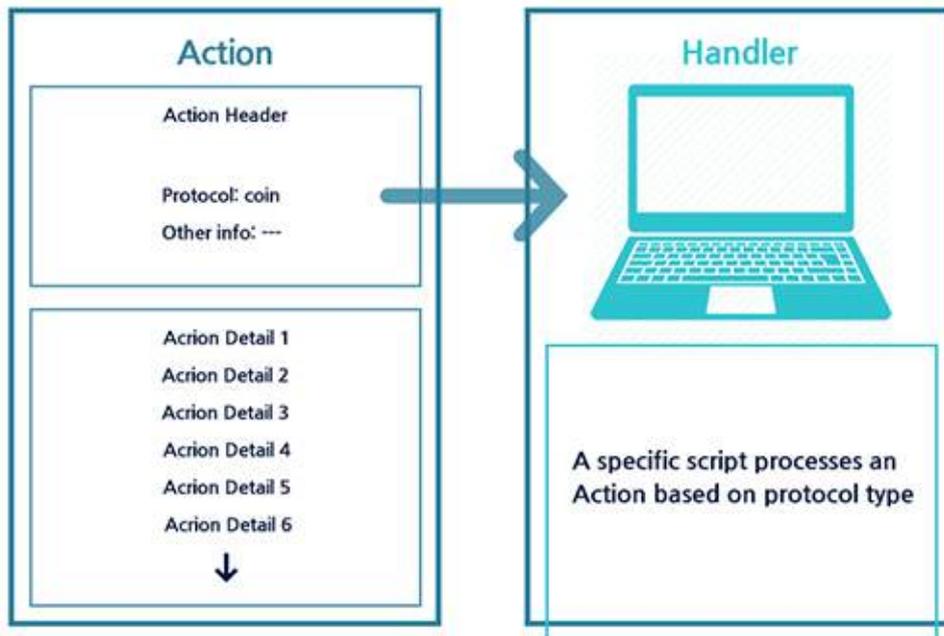
FrigateChain will always be attentive to improve the system where it can. To strive for the best, FrigateChain incorporates a community-driven proposal system where both FrigateChain and community can propose a custom format and turn it into a standard format after careful scrutiny and adjustments. Anyone in the community will be able to share and suggest ideas to improve the existing proposal or they can even create a new proposal. Based on the popularity of the proposal and the necessity of the proposed solution, it will be implemented accordingly.

## Handler

---

Handlers are scripts that will run when receiving an action. When an action is received, the handler will look at the header to determine which scripts to run based on the protocol.





## Transparency

Due to the centralized nature of FrigateChain, monitoring is required to ensure the integrity of the blockchain. To address this issue, FrigateChain block information will be public for anyone to view and cross reference with any child mainnets' blockchains.

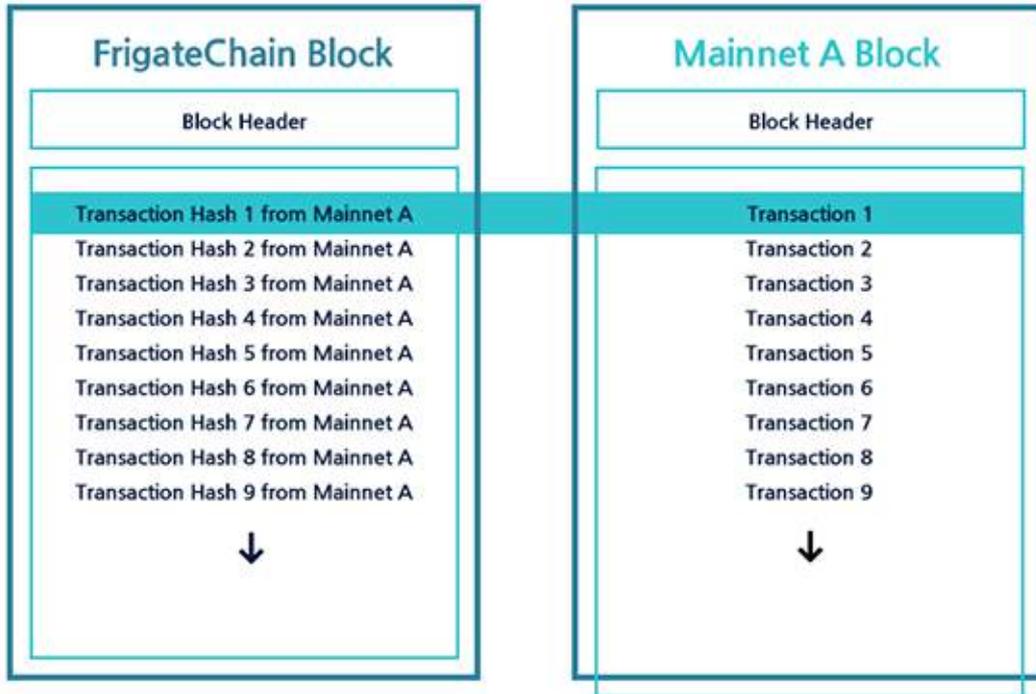
### Open Block Centralization

Unlike the traditional centralization model where the database is not visible to the public, FrigateChain will open up all blockchain information so that the public can verify the integrity of the system. This means that anyone who sets up a node and has access to all blockchain information can monitor all activities (referred to as Watchdog) in both FrigateChain and child mainnet.

### Watchdog

The role of watchdog is to ensure that there aren't any irregularities in both WiiX.NET and Child Mainnet. Since WiiX.NET stores all transaction hash of all Child Mainnet, anyone who has access to both blocks will cross confirm the legitimacy of the blocks.





# Workers

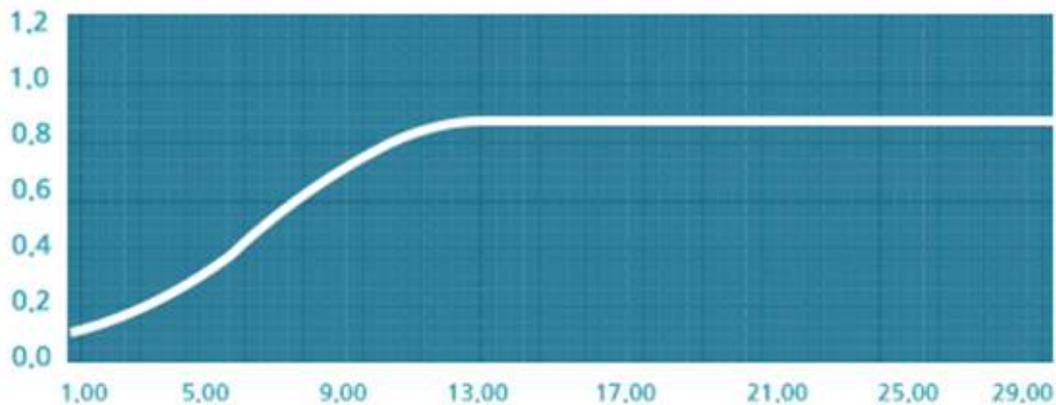
Workers refer to a collective entity that executes grid programming. Each worker will provide their CPU power creating a supercomputer that processes transactions and creates blocks.



## Efficiency

---

Due to the law of diminishing returns, there exists a point where assigning more workers to a task would only increase the efficiency marginally. To ensure most efficiency, FrigateChain divides workers into groups and assign them to process transactions.



## Block Reward

---

Compensation to the workers will be given entirely from transaction fees. The workers will be rewarded when a block is produced and their reward will be directly proportional to the amount of work, CPU/GPU power, provided on creating that block.

# Conclusion

---

FrigateChain is a platform that enables other developers to easily create mainnet and provides many features such as fast transactions that predecessors were struggling with.

