White Paper of Codablecash

Tomohiro Iizuka
https://www.codablecash.org

Index

1. Introduction....4
    1.1. Continuous Integration Framework of Smart Contract....4
        1.1.1. Relational Database on Smart Contract....5
        1.1.2. Stateless Object Oriented programming language....5
        1.1.3. Upgrade Smart Contract....5
    1.2. Scalable Ledger Network....6
        1.2.1. Speedy Blockchain Consensus with Pre-calculated Hash....6
        1.2.2. Network sharding for Scalable Blockchain....7
2. Blockchain P2P Network....8
    2.1. Network Topology....8
    2.2. Blockchain and Shard Network....8
    2.3. Node Identifier and Miner Nodes....8
3. Blockchain & Ledger....9
    3.1. Entire Architecture....9
    3.2. Flash Chain....9
    3.3. Main Chain....9
4. Blockchain Consensus Algorithm and Mining....10
    4.1. Miner Identifier....10
        4.1.1. Generate Miner Identifier....10
        4.1.2. Abolishment of Miner Identifier....10
    4.2. Network Transfer Speed....11
    4.3. Flash-Chain Consensus....11
5. Blockchain Shard Network....12
    5.1. Split Network Request....12
        5.1.1. Required Condition to Split Network....12
        5.1.2. Vote and Splitting Process....12
6. Type of Blockchain Transactions....13
    6.1. Transfer Coin Transaction....13
    6.2. Smart Contract Transactions....13
        6.2.1. Register Smart Contract Transaction....13
        6.2.2. Smart Contract Context Creation Transaction....13
        6.2.3. Creating Smart Contract Instance Transaction....13
        6.2.4. Upgrade Smart Contract Instance Transaction....13
    6.3. Miner Identifier Transaction....13
        6.3.1. Register Miner ID Transaction....13
        6.3.2. Ban Miner ID Transaction....13

7. SmartContract....................14
7.1. Entire Architecture for Continuous Integration....................14
7.2. Object Oriented Language....................14
7.2.1. Main Instance....................14
7.2.2. Off-Chain and On-Chain Method....................14
7.2.3. Inter Smart Contract Communication....................14
7.3. Relational Database....................14
7.3.1. Data Definition Language....................14
7.3.2. Data Manipulation Language....................14
7.3.3. Data Query Language....................14
8. Detail of Smart Contract Life Cycle....................15

1. Introduction

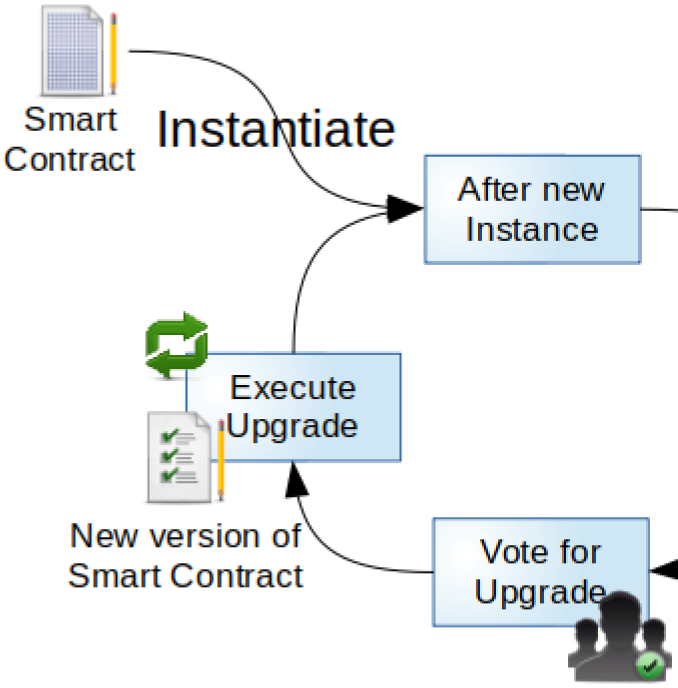
Codablecash is a blockchain platform to make Web3.0 based DAO and DAPP(Decentralized Application). It supports continuous integration of Smart Contract program.

In addition to the Smart Contract functionality, it has blockchain ledger which can support both scalability and enough short latency of transaction.

Codablecash has following functional features to implement those requirement for blockchain.

1.1. Continuous Integration Framework of Smart Contract

The goal of Codablecash is to implement Smart Contract Life Cycle on scalable blockchain network.

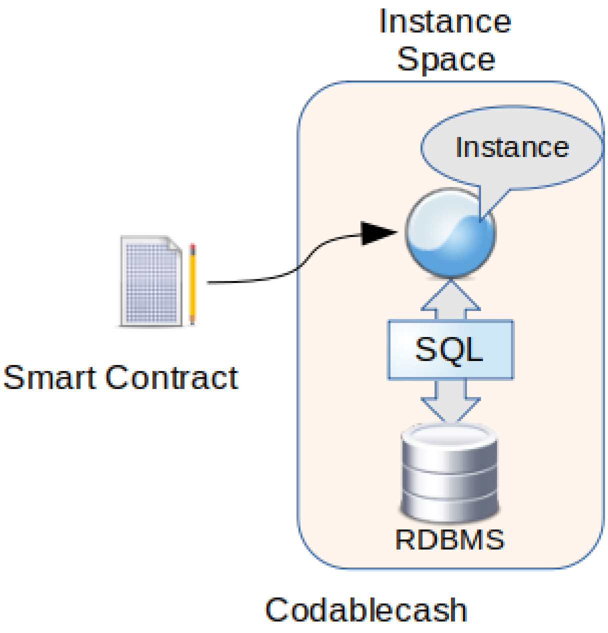


Smart Contract is program working on blockchain. As other programs need to upgrade, it also need it. When upgrade is necessary, it has to make consensus by some method like voting. The method to make consensus can be written in Smart Contract program.

After make sure the consensus, the instance can be upgraded into new version of Smart Contract program.

1.1.1. Relational Database on Smart Contract

We can use Relational Database Management System(RDBMS) on Smart Contract instances.



When a Smart Contract Instance is created from Smart Contract program, an Instance Space is created and the instance is created in it. An RDBMS instance is also created in it.

1.1.2. Stateless Object Oriented programming language

Smart Contract program can be written in Object Oriented Programming language similar to Java language.

The language can implement the Main Instance of the Smart Contract. The instance is stateless, therefore it can’t store persistent data, because the data and algorithm have to be divided to execute smooth upgrade of the Smart Contract Program.

1.1.3. Upgrade Smart Contract

On upgrading Smart Contract program for the instance, following check is necessary.

- The new Smart Contract program is compatible with current version
- The new Database Schema is compatible with current version

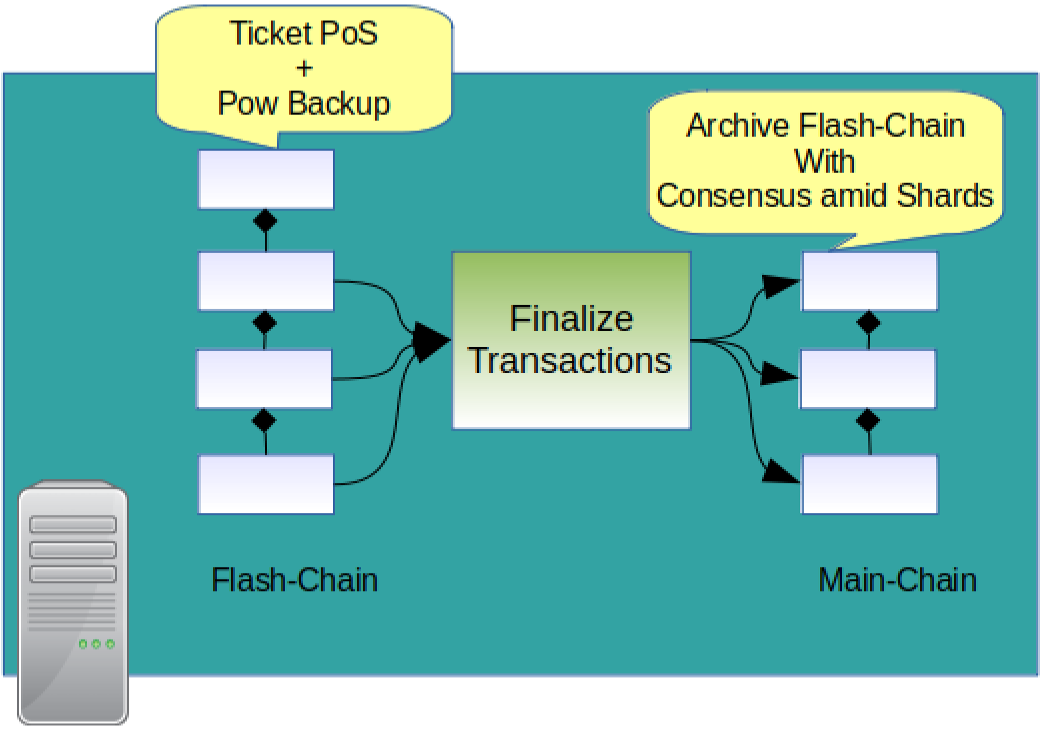
The algorithm of the program and data management are dissociated with database schema, we can check compatibility easily, and upgrade using ALTER commands of SQL language.

1.2. Scalable Ledger Network

Both scalability and response of transaction are essential factor on using blockchain ledger.

1.2.1. Speedy Blockchain Consensus with Pre-calculated Hash

Blockchain ledger of Codablecash consists of two blockchains, which are Flash-Chain, and Main-Chain.



Flash-Chain is front blockchain to put transaction from memory pool. Then the system do following task.

1. Validate transactions
2. Decide order to execute transactions

The consensus algorithm of Flash-Chain is Ticket PoS and PoW. Basically it uses Ticket PoS, which is like DPoS algorithm, and it fails, it uses PoW to generate a new block, instead of Ticket PoS, at the failed height's block.

When the Flash-Chain's block is finalized, then finalized blocks are archived into Main-Chain. A Main-Chain's block contains several Flash-Chain's block.

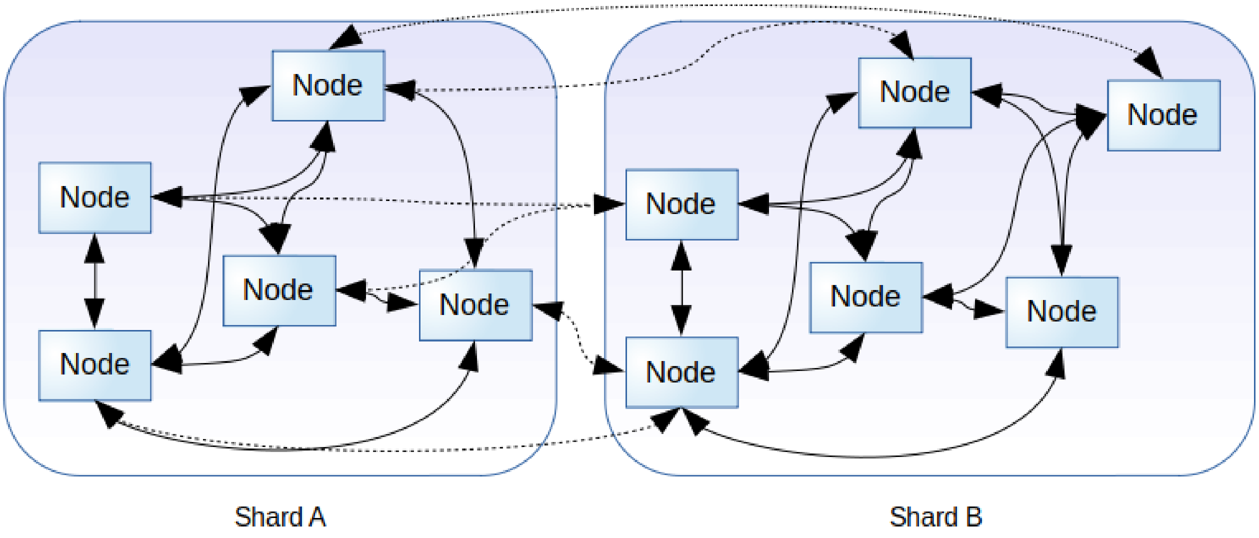
The Main-Chain's header contains information which is necessary for inter shard network communication. The shard network is used to make the blockchain scalable.

1.2.2. Network sharding for Scalable Blockchain

Codablecash uses network sharding method to scale transaction processing speed.

2. Blockchain P2P Network

Network of Codablecash is like below.



2.1. Network Topology

The network has several partitions of p2p network for each shard. And the node in a shard has connections to nodes in other shards.

Therefore, the node can transfer data to following ranges.

- Broadcast blocks to nodes in same shard
- Broadcast block header to all nodes in entire network

2.2. Blockchain and Shard Network

Each shard network has own blockchain. Therefore if there are two shards, two blockchains are there, and they works simultaneously.

2.3. Node Identifier and Miner Nodes

The nodes has own node identifier. The node identifier is unique identifier which the node can generate by them selves.

In addition to that, Miner Identifier exists. The Miner Identifier have to be registered in the blockchain. And the node which has Miner Identifier can be the Miner node.

3. Blockchain & Ledger

Codablecash

3.1. Entire Architecture

3.2. Flash Chain

The Flash-Chain consists of two types of blocks, those are PoS Block, and PoW Block.

3.3. Main Chain

4. Blockchain Consensus Algorithm and Mining

Miners can generate blocks with Consensus Algorithm rule. Miners mainly mines blocks of Flash-Chain. Main-Chain is generated on finalizing Flash-Chain.

4.1. Miner Identifier

In order to do mining Codablecash, a Miner Identifier is necessary.

4.1.1. Generate Miner Identifier

In order to get it, we have to send transaction to generate miner identifier.

The transaction contains nonce of hash, and the difficulty of nonce have to be greater than one which is calculated for each height of block.

The difficulty depends on number of PoW blocks, included in last 2000 blocks. That’s because if existence of PoW blocks means shortage of PoS blocks, which is generated by miners with ID.

If there are no PoW blocks, new miner identifier is not available.

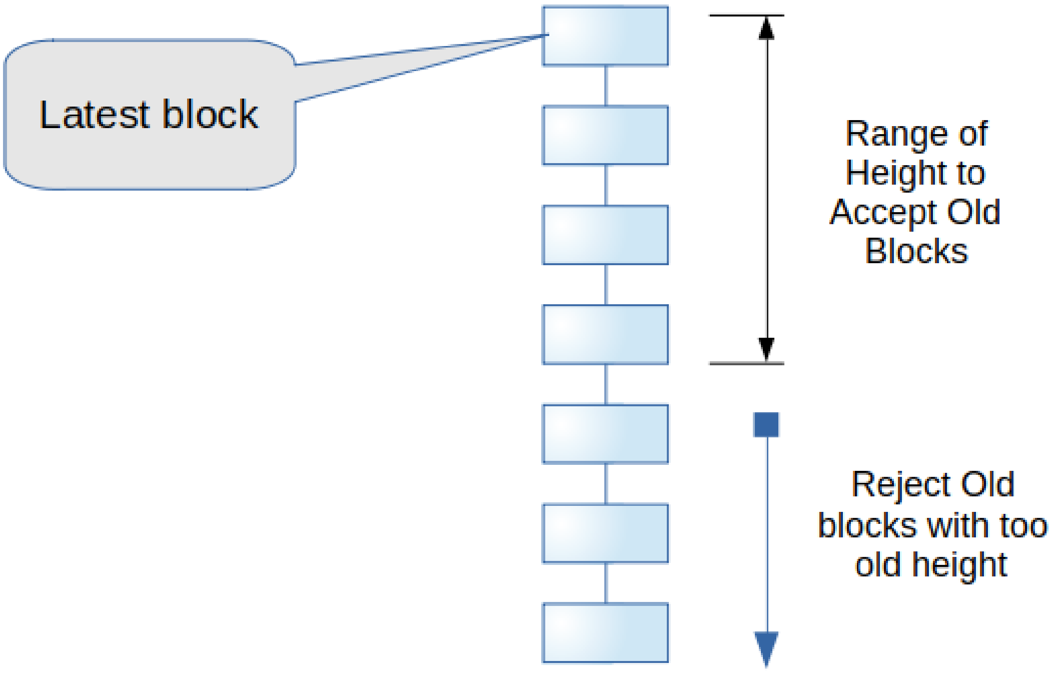
4.1.2. Abolishment of Miner Identifier

Miner Identifier is abolished by following reason.

- Miner Identifier is not used for proper period.
- Miner execute malicious software on mining, and generate wrong block.

4.2. Network Transfer Speed

Codablecash network assumes a block generated by miners can be transferred to all nodes in the network within proper time. Therefore a node rejects too old blocks.


Latest block
Range of Height to Accept Old Blocks
Reject Old blocks with too old height

An old block have to reach all nodes before the height of newest node is not greater than 4.

4.3. Flash-Chain Consensus

5. Blockchain Shard Network

5.1. Split Network Request

5.1.1. Required Condition to Split Network

5.1.2. Vote and Splitting Process

6. Type of Blockchain Transactions

6.1. Transfer Coin Transaction

6.2. Smart Contract Transactions

6.2.1. Register Smart Contract Transaction

6.2.2. Smart Contract Context Creation Transaction

6.2.3. Creating Smart Contract Instance Transaction

6.2.4. Upgrade Smart Contract Instance Transaction

6.3. Miner Identifier Transaction

6.3.1. Register Miner ID Transaction

6.3.2. Ban Miner ID Transaction

7. SmartContract

7.1. Entire Architecture for Continuous Integration

7.2. Object Oriented Language

7.2.1. Main Instance

7.2.2. Off-Chain and On-Chain Method

7.2.3. Inter Smart Contract Communication

7.3. Relational Database

7.3.1. Data Definition Language

7.3.2. Data Manipulation Language

7.3.3. Data Query Language

8. Detail of Smart Contract Life Cycle