π Group

π White Paper (v2.0)

CATALOGUE

Abstract

This article introduces the principles and technical architecture of a brand new blockchain technology π.

This article surveys the advantages and disadvantages of the existing blockchains like BitCoin, Litecoin, Ripple and BitShares, then analyses consensus process and usage of these technologies. With combination of the application direction of π, we design the blockchain technology of this brand-new concept π.

π solves the network efficiency problems through the consensus mechanism of IPoS. The new assets are additionally issued and allocated through the original π incentive algorithm to promote the network activity and its healthy and stable development. π invented referendum mechanism to decide the subsequent issuance rate. Through this democratic way, every user of π has the right to decide where platform goes.

π provides rich smart contract interfaces and the users may flexibly create various types of smart contract applications on the blockchain to fulfil their different requirements.

Chapter 1 Background

1. Bitcoin

On November 1, 2008, a person claiming named Satoshi Nakamoto published a thesis titled *Bitcoin: A Peer-to-Peer Electronic Cash System* on the Internet, proposing how to construct a decentralized electronic transaction system with a peer-to-peer network.

On January 3, 2009, he achieved the first Bitcoin reference client and dug up the first block in the Bitcoin history. The first 50 Bitcoins were generated.

From then on, Bitcoin is becoming a flourishing ecological system step by step. Although it went through various crises and its market value fluctuates significantly, the thought of Satoshi Nakamoto's is well known by people all over the world via Bitcoin as a medium. People start to explore the infinite possibility of Bitcoin and the blockchain technology behind it.

As a pioneer of the blockchain technology, Bitcoin has the following characteristics:

(1) Recording the data in an open distributed database as blockchain. The data are stored in the blocks which connected each other in chronological order as the chains. All blocks have an immutable timestamp and the data generated there cannot be tampered.

(2) Ensuring the data security as well as participant's privacy in a manner

of asymmetric encryption.

(3) Issuing the new coin in a manner of PoW (Proof of Work) as the motive power of network development. PoW enhances robustness of the network which gets stronger when more people are involved in the mining.

(4) The transaction parties could safely transact without participation of a centralized mint or authority. This makes Bitcoin a real decentralized currency, which is completely out of control by governments or commercial structures.

(5) No risk of double spending.

The PoW mining design greatly enhances people's enthusiasm to participate in Bitcoin and more and more calculation power is invested, which greatly ensure the network reliability.

However, along with expansion of the network scale, Bitcoin mining has undergone several stages of CPU mining, GPU mining, FPGA mining, ASIC mining, large-scale cluster mining. The computational power of the mining from the initial few tens of M Hash per second, developed to the present T Hash per second level. Within just a few years, the computational power has increased millions of times.

With the progress of mining technology and the increase of mining computational power, it is very difficult for the ordinary miners to mine Bitcoin by themselves, this facilitates the miners to work together to form a pool for mining. Such spontaneous behavior is in breach of the design of Satoshi Nakamoto, because once a pool masters more than 50% of the calculation

power, the foundation of the Bitcoin will be fluctuated.

In addition, another side effect brought about by the mining competition is higher and higher operation cost of the network and the maintenance cost apportioned to each Bitcoin is also getting higher, which may result in the reduction of Bitcoin network participation, and thus making the network scale shrink.

At present, it will take at least 10min to generate a block for the Bitcoin network, too slow for instant payment. In addition, the handling capacity of the Bitcoin network is also limited. At present, it only reaches the handling speed of unit digit per second (TPS), which cannot meet the requirements at the occasion where frequent transactions with a small amount are conducted.

2. Other Blockchains

The emergence of the Bitcoin also stimulates that of a series of derivative technologies. Based on the basic principles of the Bitcoin, such technologies add their own commercial or technical innovation.

2.1 Litecoin

Essentially, Litecoin is built on the same principles of Bitcoin, Litecoin has the following characteristics:

(1) Litecoin accelerates the block generation speed, it changes the

generation speed 4 times faster from 10 minutes per block to 2.5 minutes per block. A faster block generation speed means faster transaction process speed.

(2) The expected total output of the Litecoin is 84 million, which is just 4 times as Bitcoin's 21 million.

(3) Litecoin uses crypt as its PoW algorithm. Compared with Bitcoin's Hash algorithm, the importing of GPU mining has no advantages beyond CPU mining (This was before ASIC mining machine's invention).

Just as Bitcoin is called the gold of digital currency, Litecoin's goal is to become the silver of the digital currency. It produces faster, its total output is higher and it's more dividable, all these features help it to fulfil this goal.

However, essentially, Litecoin does not change any core idea of Bitcoin. Although it increases total output, accelerates the block generation speed, it's still the same as Bitcoin, the handling efficiency is the same magnitude as Bitcoin, which still cannot meet the highly frequent transaction requirements in the existing commercial society.

2.2 Ripple

Ripple's vision is to build the world's first open payment network. It is a distributed P2P clearing network.

Ripple has the following characteristic:

(1) It supports transfer of multiple currencies, in addition to Ripple's own native currency XRP, it also supports fiat currency (such as dollars, yen, yuan, etc.), a variety of digital currency (Bitcoin, Litecoin, etc.) transfer. It achieves a variety of digital assets transfers through gateway trust.

(2) It supports the automatic exchange of trans currency, through the role of market maker, Ripple helps users to pay any type of currency to get any other type of currency, which achieves all currencies circulation over the whole network.

(3) Fast transaction confirmation speed, Ripple’s transaction confirmation process can be completed within a few seconds. Ripple introduced a "Consensus" mechanism, only special nodes participate the vote process, this helps Ripple to verify and confirm the transaction in very short time.

(4) The client does not need to download the blockchain. The ordinary node can drop blocks that has been verified, only to retain the most recent verification of the ledger and a link to the previous ledger, and thus synchronization and download the total ledger are much easier.

(5) No mining, Ripple uses consensus mechanism, only the validator

nodes are involved in the block generation process. Validator nodes have a very strict join mechanism. By ensuring the security of the validator nodes, guarantees the integrity of the entire network. This reduces the waste of resources generated by the peer-to-peer mining competition.

(6) The total amount of native currency is limited. Initially, Ripple has issued all 100 billion XRP and is committed to no additional issuance. Users spend a certain amount of XRP on each transaction (the amount is quite low, less than 1 cent) as a transaction fee. The transaction fee will not be given to anyone, but permanently disappear from the system.

In Ripple, everyone can be their own bank. They can issue credits, and become credit channel (for example, A wants to borrow money from B, they do not know each other, but they both know C, then C can be their credit channel, C borrows money from B, and then lend it to A, this achieves the goal - A borrow money from B).

Ripple is based on acquaintance and trust lines. This design requires a person who wants to transfer or lend money has to have friends on the network, otherwise, he cannot establish a trust line with other users.

This gateway trust model is very similar to the mode of the existing bank and exchange. We can say it's just an expression on P2P network of these institutions. This is in breach of the principles of decentralization and unnecessary trust declared by Bitcoin.

Only a few validator nodes participating in the block generation process

helps Ripple accelerate its transaction process speed. But it's validator joining mode is born doubtful. If some of its founder's or later joined validator nodes are malicious, or compromised by a malicious attacker, it's very likely that more than half of the nodes are committed fraud, putting the entire network in danger.

2.3 BitShares

Bitshares is an open source distributed trading system that supports virtual currency, legal currency, precious metals and other valuable assets. The system provides a solution of a decentralized exchange, makes everyone having the ability to become an exchange.

Bitshares' consensus mechanism is DPoS (Delegated Proof-of-Stake). 101 delegates are elected by all shareholders to generate blocks. We can see them as 101 mining pools, they have equal rights. BTS shareholders can vote for other delegates (pools), if these delegates fail to provide stable service or trying to do evil.

Because there is no mining and no blocks generation by multiple nodes, BitShares' processing speed has been greatly improved. Its official statement says it has achieved a processing capacity as VISA and MasterCard's processing capacity sum up.

Bitshares has following characteristics:

(1) Price-Stable Cryptocurrencies – SmartCoins, provides the freedom of

cryptocurrency with the stability of the dollar. A SmartCoin is a cryptocurrency whose value is pegged to that of another asset, such as the US Dollar or gold. SmartCoins always have 100% or more of their value backed by the BitShares core currency, BTS, to which they can be converted at any time at an exchange rate set by a trustworthy price feed.

(2) The Decentralized Asset Exchange - BitShares provides a high-performance decentralized exchange, with all the features you would expect in a trading platform. It can handle the trading volume of the NASDAQ, while settling orders in real time.

(3) Industrial Performance and Expansibility. High performance blockchain technology is necessary for cryptocurrencies and smart contract platforms to provide a viable alternative to existing financial platforms. BitShares is designed from the ground up to process more transactions every second than VISA and MasterCard combined. With Delegated Proof of Stake, the BitShares network can confirm transactions in an average of just 1 second.

(4) Delegated Proof-of-Stake Mechanism. Delegated Proof of Stake (DPOS) is the fastest, most efficient, most decentralized, and most flexible consensus model available. DPOS leverages the power for stakeholders to resolve consensus issues in a fair and democratic way. Deterministic selection of block producers allows transactions to be confirmed in an average of just 1 second.

(5) Referral Rewards Program. BitShares has an advanced referral

program built in its software. Financial networks derive their value primarily from their network effect: more people on the same network increases the value of that network for everyone. BitShares capitalizes on this by rewarding those who sign up new users, and does so in a fully transparent and automated way.

(6) Dynamic Account Permissions. Every account can be controlled by any weighted combination of other accounts and private keys. This creates a hierarchical structure that reflects how permissions are organized in real life, and makes multi-user control over funds easier than ever. Multi-user control is the single biggest contributor to security, and, when used properly, it can virtually eliminate the risk of theft due to hacking.

However, Bitshares' concept is very complicated, which slows down people's recognition process for it. The manner of BitShares to issue all BTSs in the beginning makes people doubt the fairness to distribute its original assets.

Chapter 2 π

1. Introduction of New Blockchain- π

Through introduction above, we see that various blockchain technologies have their respective advantages and disadvantages. By referring to such existing technologies, we want to achieve a safe and efficient blockchain, which is easily expanded and promoted, and named π by us.

π is one of the most famous constants in mathematics. As π, it has unlimited sense of beauty and charm; as a special infinite non-repeating irrational number, it is full of mystery, greatly stimulating the exploration desire of people. The reason that we take it as the name is to hope that our technology will be perfect and charming like it.

2. Positioning of π

Positioning of π is to achieve a safe and reliable common blockchain platform with light weight and low cost, which is easily used and expanded. Its possible application fields include:

- Decentralized application
- Digital assets management
- Smart contract platform

We issue an original digital asset π as an original driving force of all activities on this platform. To participate in the π network activities in different manners will consume or obtain certain π. The existence of π is to stimulate the user to participate in the construction of the π network and get appropriate income from it. Along with development of π network, all π holders will be benefited continuously.

3. Consensus Mechanism of π

In the previous article, we studied Bitcoin and Litecoin, which are based on PoW mechanism. The advantage lies that they are totally decentralized, while their disadvantages are that the block generation cycle is too long and the process capability is low. We studied Ripple, whose block generation is based on validator voting consensus. It increases block generation speed and transaction process speed, but these are achieved at the expense of the absolute meaning of decentralize.

At last, we studied Bitshares' DPoS (Delegated Proof-of-Stake) mechanism. By electing block generation node, all users are involved in the block generation process. With only a few delegates generate blocks, computation and communication costs are cut down, which helps getting ideal block generation speed and transaction process speed.

Through research on the existing consensus mechanism above, we

created a consensus mechanism called IPoS (Improved Proof of Stake) which are based on PoW (Proof of Work) with a lot of improvements done by us.

IPOS has the following features:

1. Blocks are generated by only a few nodes elected by all users. These nodes usually hold a lot π by themselves, and are widely accepted by the network. They can get more benefits by honest work beyond fraud work. With only a few nodes to generate block, block generation speed and transaction process capability are greatly increased. Block generation speed of π is set as 3 seconds per block, for processing small amount payments in real time.

2. Using construction capital to do incremental issuance (we'll explain construction capital in details in the following passages). Its core idea is to encourage uses to hold π, to express their recognition of π, helping the development of π. Construction capital also brings user income.

3. Through the acceleration mechanism of construction capital, we introduce social elements into π, we encourage users to promote π to others, and get more income from promotion process.

4. Increasing Expansibility through smart contract. π executes smart contracts during consensus process, smart contract helps users to build different types of applications to fulfil their requests.

4. Performance and Expandability

Benefited from the IPoS mechanism, the mining concept in π is completely different from PoW of Bitcoin. The block node mined from π is elected by the algorithm, rather than the competition of many nodes.

π generates a block in 3 seconds, this is much faster when comparing with Bitcoin's 10 minutes and Litecoin's 2.5 minutes. When comparing with Ripple and Bitshares, which produce block in about 10 seconds, it still has advantage.

Therefore, the transaction handling capacity of π is enhanced by several orders of magnitudes comparing with that of the Bitcoin. We expect that the handling capacity of π will at least reach the magnitude of VISA and MasterCard in the future or even higher.

As for expansibility, π is totally decentralized where everyone can run their own π node to join the network. By adding new nodes, we can improve the responsiveness of π, so we can say that π has infinite expansibility.

5. Security and Privacy

Like Bitcoin, the whole π system is based on the cryptology system through mathematical theoretical demonstration. Therefore, we have good grounds to believe that π could at least reach the security degree of that of Bitcoin.

All transactions on π network are anonymous, so there's no need to worry about leaking personal information.

For the block data, we provide the encryption option to ensure that only transaction parties may know about the specific transaction contents and the nodes on the network only verify the transaction legality, for which it is unnecessary to know about the specific contents, radically meeting the user's requirements for privacy.

6. Issuance Algorithm

While π is getting wider application, the demand for it gets higher. Moreover, according to the law of economic development, the economic aggregate grows continuously. Therefore, we design a reasonable issuance algorithm to adapt to such economic aggregate and change in application mass. A reasonable issuance algorithm should not only adapt to the economic development law, but also spur the system promotion and attract

more people to actively use the π network and participate in its construction process.

At the beginning of creation, we will issue 10 million π as initial capital. Then, we will issue more according to certain rules during the process to generate new blocks in the future.

In order to elaborate the specific issuance rules, we first introduce several concepts:

6.1 Construction Capital

A stable holding of π means the recognition of its network. More people holding π stably means that it is recognized more pervasively and the construction capital is a tool designed for the user to recognize the network.

The construction capital means that the user locks π held by him/ her for a certain period (π in the locking period cannot be transacted) to express his/ her recognition and confidence for the network. The system will provide the user with certain issuance rewards. In principle, the more π are locked, the longer the period is, the more income will be earned.

The following conditions shall be defined for describing the construction capital:

1. Locked amount V: It is used for describing user's actual locking amount. Meanwhile, we introduce an effective amount constant V_{min}. Only when the π amount locked by the construction capital V be greater than the constant V_{min},

it will be regarded to be effective. For the establishment of this constant, more considerations are given to the achievement of the decentralized blockchain. If the user is allowed to establish the construction capital with too small amount, there may be lots of construction capital accounts with small amounts in the system. Although they may get a very little of issuance income during issuance, they consume lots of computing and network resources. Therefore, for health and efficiency, we introduce this constant V_{min}, which shall meet the conditions when it is expressed with a formula as follows: $V \geq Vmin$

Issuance period T: It is used for setting the income distribution period acceptable to the user. Likewise, for the system efficiency, the user's issuance income is not obtained in each block, but distributed according to the issuance period T designated by the user. Namely, a part of issuance income is distributed every complete period with T blocks. Because the issuance computing per time consumes the system resources, the fixed handling charge F with a small amount will be charged every time the issuance income is distributed. Theoretically, the longer the issuance period selected by the user, the less to be charged, which encourages the user to set a longer issuance period to reduce the system resource consumption and handling charge expenditure. We set a minimum issuance period constant T_{min} for restricting the user to set too short issuance period, with its formula as follows: $T \geq Tmin.$

Locking duration L: L is used for marking the locking duration of the

construction capital, which is a positive integer greater than or equal to 1 and valuing the locked issuance period quantity. If the number of block is used for expressing the locking duration, the formula is as follows: **L*T.**

6.2 Acceleration Mechanism

The acceleration mechanism is a mechanism encouraging the user to promote the π network to others.

When the construction capital was established, the user could help others to accelerate with the construction capital, increasing the yield efficiency of the counterparty. Such mechanism forms a flexible social system where the user may develop new users through actively promoting the π network and get acceleration from the new users for additional income.

Suppose there were n accounts accelerating account A, the construction capital amount of each account is V_i(C = 1,2,3 … n), the period of the construction capital is T_i (i = 1,2,3 … n), and the locking duration is L_i (i = 1,2,3 … n). We use S to represent the acceleration index of account A, with its definition as follows:

$$S = \sum_{i=0}^{n} Vi * Ti * Li$$

We assume that the construction capital amount of account A is V and its issuance period T, the acceleration effect is that when the acceleration index S of user A reaches V*T*L, the user immediately obtains the income of one

period and S deducts V*T*L and L reduces one period. Namely, the construction capital of A will generate income one period in advance and the unlocking period is also reduced by one period.

The acceleration effect has a ceiling value which the maximum acceleration proportionality coefficient is set as A_{max}. We suppose L for user's predetermined output period sum, then the maximum accelerated advanced output period is L* A_{max}. Even The acceleration period counted from the acceleration index is greater than the value, L* A_{max} is the maximum period to be accelerated.

6.3 Issuance Interval

Our issuance algorithm is based on the construction capital. From the moment when the user's construction capital takes effect, the user gets the issuance income every T^{th} block(s).

The generation speed of π network block is set as 3s per block. The current value of T_{min} is 201,600, i.e. 7 days.

If the value of T set by the user is 201,600, the user gets part of issuance income every 7^{th} day.

6.4 Issuance Rate and Referendum Mechanism

6.4.1 Issuance Rate

The issuance rate d is used for marking the issuance speed of the system,

whose meaning is the yield generated by the construction capital in one year.

In the initial stage, the system performs issuance according to the preset issuance rate which is adjusted every third month. The initial issuance rate will be reduced step by step according to the pre-set value.

When the issuance rate is reduced to a certain rate (at present, the time point of the first referendum is the 4th period and the income rate for 50 weeks will be reduced to 90%), the system will initiate the referendum mechanism to decide the subsequent issuance rate.

The system performs issuance according to the issuance rates shown in Table below.

Adjusting	Issuance rate d (%)	Income for 50
2017.03.01	104.2857143	100
2017.06.01	99.07142857	95
2017.09.01	93.85714286	90
2019.12.01	Launching referendum to decide the issuance rate	
2020.03.01	Referendum tells the issuance rate	
2020.06.01	Referendum tells the issuance rate	
2020.09.01	Referendum tells the issuance rate	
2020.12.01	Referendum tells the issuance rate	
2021.03.01	Referendum tells the issuance rate	
.....		

6.4.2 Referendum

By December 1st, 2017, the income rate for 50 weeks has reduced to 90% and the system has entered into a stage of referendum to determine the subsequent issuance rate. All π holders and construction capital accounts

were participated in the vote to determine the subsequent issuance rate.

The referendum system provided the three options:

(1) Increasing the issuance rate

(2) Decreasing the issuance rate

(3) Maintaining current issuance rate

The referendum employs the mechanism that one vote for each π and any account holding π or construction capital may participate in the vote. Accounts can vote one of the three options, more holding of π or construction capital is endowed with more powerful vote. The process of voting is entirely fair and transparent.

There is a voting period for the referendum. It usually starts a week before next issuance period. For example, Dec. 1st, 2017 was the first referendum period, the system launched the referendum on Nov. 25th, 2017. Users could vote and check the poll during the period.

The referendum ends at the last block before the starting of the next period. The weighted average for all votes will be calculated as the issuance rate for the next period.

The referendum of issuance rate has its caps and collars that the final issuance rate must be ranging from 0% to 100%. In case that the income rate is reduced to 0%, the "decreasing" option will be unavailable; on the other hand, if the income rate is increased to 100%, there will not be "increasing" option.

6.5 Calculation Method

We assume that the construction capital amount of user A is V and the issuance period is T, the income D that A gets out of one issuance (either by meeting the period T or acceleration index reaching the threshold of output in advance) is calculated with the following formula:

$$D = \frac{V * T * d}{10512000}$$

where the 10,512,000 refers to the total block quantity during a year, i.e.365 days (one block per 3s).

6.6 Node Income

As a distributed network, π welcomes all users to operate their own nodes to join in. The operation of π node not only gets better access effects and enhances the robustness of the π network, but also brings about certain income for the users themselves. Every day, there are some π with an equivalence of 15% of the incremental issuance being delivered to the accounts running π nodes.

In the initial construction period, the service version upgrades so frequent that the π network must be operated in a closed environment, where only the official nodes are running and their income would be used as marketing budget. When the timing is right, other nodes would be opened for joining and any individual could obtain income fairly through operating their nodes. In the next chapter, we will explain how the nodes get accessed to

π network.

7. Getting Accessed to π Network

As an open blockchain network, π allows all users to get accessed to the network equally. In this chapter, we will explain how the nodes get accessed to π network.

7.1 Installment and Operation

7.1.1 Docker

Docker is the simplest tool for operating "Pi". The Docker mirror compiled with the latest code has been offered on Docker Hub, allowing a direct initiation of Pi node through Docker commands.

The detailed steps are as follows (Here we take initiating a service node outside packaging blocks as an example. There are some differences between consensus nodes, we will make extra illustration in the following passages.):

1. Environment Preparation

Item	Specification
OS	ubuntu x64 16.04
CPU	8 cores
Storage	32G

Hard disk	500G (SSD recommended)
Docker	17.06 or later

2. Docker Pull

```
docker pull wallex64/pi
```

3. Data Preparation

```
# Suppose data disk at /opt, create data catalogue
mkdir -p /opt/data
cd /opt/data
# Download genesis data
wget
https://raw.githubusercontent.com/pidiscovery/pi/master/genesis.json
```

4. Node Initiation

```
  # Initiating service node via Docker
docker run \
     -d \
     --name witness \     # Container
     -v /opt/data:/data \ # Data catalogue mapping
     -p 8010:8010 \       # Service port mapping
     -p 40010:40010 \     # p2p port mapping, close the mapping if
p2p not offered to public.
     -e "RPC_ENDPOINT=0.0.0.0:8010" \  # service address and port
configuration in container
     -e "P2P_ENDPOINT=0.0.0.0:40010" \ # p2p address and port
configuration in container
     -e  "SEED_NODES=
peer1.piex.pro:40010#peer2.piex.pro:40010" \ # connected peer
nodes
     -e "WITNESS_ARGS=--disable-witness-plugin" \ # witness
plug-in not running
     wallex64/pi
```

The exclusive IP and ports in SEED_NODES are the defaults. The IP can be connected to the machine only after it is verified and authorized to the white list.

The parameter list of WITNESS_ARGS contains the following information

for enabling or dismissing corresponding functions:

Parameter	Function	Description
--disable-witness-plugin	Not running as witness	It is not running as consensus nodes, and not running blocks to generate logic. Using this parameter on service node would improve its running efficiency.
--disable-account-history-plugin	Not recording account history.	It doesn’t record account history. When public inquiry of account info is not available, this parameter can be enabled as it saves internal storage.
--disable-market-history-plugin	Not recording exchange history	It doesn’t record exchange history. When public inquiry of exchange info is not available, this parameter can be enabled as it saves internal storage.
--disable-incentive-history-plugin	Not recording Issuance history	It doesn’t record the history of construction capital issuance. When public inquiry of issuance history is not available, this parameter can be enabled as it saves internal storage.
--disable-transaction-record-plugin	Not recording transaction history	It doesn’t record the transaction history. When public inquiry of transaction history is not available, this parameter can be enabled as it saves internal storage.

5. Running Status Checkout

After the nodes are enabled, it takes a long time to load history block

data from the network. But we can confirm the newest data by checking logs:

```
# Entering log
cd /opt/data/logs
# Checking log
tail -f efault.log
```

If the output log is similar with the following and with a timestamp showing the current time, the data has caught up with the main nodes:

```
2018-04-09T09:19:54 th_a:invoke
handle_block        handle_block ] Got block: #17726374 time:
2018-04-09T09:19:54 latency: 388 ms from: pic07  irreversible:
17726365 (-9)        application.cpp:501
  2018-04-09T09:19:57 th_a:invoke
handle_block        handle_block ] Got block: #17726375 time:
2018-04-09T09:19:57 latency: 351 ms from: pic05  irreversible:
17726367 (-8)        application.cpp:501
  2018-04-09T09:20:00 th_a:invoke
handle_block        handle_block ] Got block: #17726376 time:
2018-04-09T09:20:00 latency: 616 ms from: pic09  irreversible:
17726368 (-8)        application.cpp:501
  2018-04-09T09:20:03 th_a:invoke
handle_block        handle_block ] Got block: #17726377 time:
2018-04-09T09:20:03 latency: 339 ms from: pic  irreversible:
17726369 (-8)          application.cpp:501
  2018-04-09T09:20:06 th_a:invoke
handle_block        handle_block ] Got block: #17726378 time:
2018-04-09T09:20:06 latency: 358 ms from: pic02  irreversible:
17726370 (-8)        application.cpp:501
  2018-04-09T09:20:09 th_a:invoke
handle_block        handle_block ] Got block: #17726379 time:
2018-04-09T09:20:09 latency: 368 ms from: pic03  irreversible:
17726371 (-8)        application.cpp:501
  2018-04-09T09:20:12 th_a:invoke
handle_block        handle_block ] Got block: #17726380 time:
2018-04-09T09:20:12 latency: 339 ms from: pic  irreversible:
17726372 (-8)          application.cpp:501
  2018-04-09T09:20:15 th_a:invoke
handle_block        handle_block ] Got block: #17726381 time:
2018-04-09T09:20:15 latency: 382 ms from: pic06  irreversible:
17726372 (-9)        application.cpp:501
  2018-04-09T09:20:18 th_a:invoke
```

```
handle_block        handle_block ] Got block: #17726382 time:
2018-04-09T09:20:18 latency: 372 ms from: pic09  irreversible:
17726373 (-9)        application.cpp:501
  2018-04-09T09:20:21 th_a:invoke
handle_block        handle_block ] Got block: #17726383 time:
2018-04-09T09:20:21 latency: 355 ms from: pic04  irreversible:
17726373 (-10)          application.cpp:501
```

7.1.2 Code

1. Compiling environment

Item	Specification
OS	Ubuntu 16.04 x64
CPU	4 cores +
Storage	8G+

2. Installing dependencies

```
  sudo apt-get update
  sudo apt-get upgrade -y
  sudo apt-get install -y git build-essential libboost-all-dev
libssl-dev cmake autoconf
```

3. Downloading codes and dependencies

```
cd /path/of/pi/src
  # Cloning the newest codes from github.
  git clone https://github.com/pidiscovery/pi.git
  cd pi
  # Downloading sub-dependencies
  git submodule update --init --recursive
```

4. Compiling and installing

```
cd /path/of/pi/src/pi
  mkdir build
  cd build
  cmake ..
  make
  make install
```

5. Data preparation

```
    mkdir /path/of/pi/data
    cd /path/of/pi/data
    # Downloading genesis data
    wget
  https://raw.githubusercontent.com/pidiscovery/pi/master/genesis.
  json
```

Prepare config.ini file. Following is the example of a configuration file:

```
# rpc service listening address
  rpc-endpoint = 0.0.0.0:8010
  # p2p listening address
  p2p-endpoint = 0.0.0.0:40010
  # peer address can be connected only after the peer pairs connecting
to open ports of the machine.
  seed-node = peer1.piex.pro:40010
  seed-node = peer2.piex.pro:40010
  # genesis file path
  genesis-json = /path/of/pi/data/genesis.json
  # Running synchronized with main chain are connected to TV, the
following two configurations is maintained as false.
  enable-stale-production = false
  required-participation = false
  # Deploying the time interval of candlestick charts in a unit of
second.
  bucket-size = [15,60,300,3600,86400]
  # The size of history data
  history-per-size = 1000
```

6. Running

```
  witness -d /path/of/pi/data
```

7.2 Nodes Participation

7.2.1 Service Node

The service nodes do not participate in the consensus generating blocks. However, it contains all block data to provide comprehensive inquiry service and transaction submission service. The service node offers RPC services via

WebSocket, which can extend the service capacity by increasing node quantities.

It is recommended to run service nodes with Docker as its inquiring scope can be controlled by parameters.

7.2.2 Consensus Node

The consensus node refers to that participating in the generation of consensus. As DPoS consensus is adopted in Pi, its nodes need to prepare the following for participating in block generation:

1. Preparing an account

2. Upgrading the account to membership

3. Upgrading the account to witness (consensus node account)

4. Attracting votes to make it top 11 in the poll

5. Binding the account with node

6. Running the node

The following passage is instructing the steps in details.

0. Running the CLI wallet

This is the preparation for the following steps. The CLI wallet can be used via code compiling or docker. We will introduce a simple method here.

First of all, we should run a service node via docker and enable a CLI wallet through the following commands:

```
# Entering docker container
docker exec -it witness bash
# Connecting service via CLI wallet
cli_wallet --server-rpc-endpoint=ws://127.0.0.1:8010 \
```

```
--chain-id=ae471be89b3509bf7474710dda6bf35d893387bae70402b54b616d
72b83bc5a4
```

After connected, the output is as following:

```
  Logging RPC to file: logs/rpc/rpc.log
  1019503ms
th_a       main.cpp:120                  main                ]
key_to_wif( committee_private_key ):
5KCBDTcyDqzsqehcb52tW5nU6pXife6V2rX9Yf7c3saYSzbDZ5W
  1019503ms
th_a       main.cpp:124                  main                ]
nathan_pub_key:
PIC6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
  1019503ms
th_a       main.cpp:125                  main                ]
key_to_wif( nathan_private_key ):
5KQwrPbwdL6PhXujxW37FSSQZ1JiwsST4cqQzDeyXtP79zkvFD3
  Starting a new wallet with chain ID
ae471be89b3509bf7474710dda6bf35d893387bae70402b54b616d72b83bc5a4
(from CLI)
  1019504ms
th_a       main.cpp:172                  main                ]
wdata.ws_server: ws://127.0.0.1:8010
  1019506ms
th_a       main.cpp:177                  main                ]
wdata.ws_user:  wdata.ws_password:
  Please use the set_password method to initialize a new wallet before
continuing
  new >>>
```

Initialize the wallet and set a password (attention to the strength), then unlock the wallet.

```
New >>> set_password your_password
  null
  locked >>> unlock your_password
  null
```

The next steps are carried out after the preparation. If not specifically stated, the next steps are all completed in CLI wallet dialogue.

1. Preparing an account

Prepare a pair of secret key. The secret keys must be kept well as its loss would cause the corresponding assets loss.

```
unlocked >>> suggest_brain_key
  {
    "brain_priv_key": "BINUKAU COITURE BURSTER CITYISH THONGY SAIMY
PACKAGE BRISS OLLAMH DESCALE RETTORY ADORNER LOATHLY POSITOR CADDOW
DUBBY",
    "wif_priv_key":
"5Hq7yLvUwTX1xVn7GtbNsuq9uBkZLbLWXSfH566es2xnHdqBBRU",
    "pub_key":
"PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj"
  }
```

Send sufficient PIC to the address of "PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj" and activate the account which is guaranteed to use for transaction charges in the following steps. Recommendations are practicing on online wallet.

Import the new account into the wallet

```
unlocked >>> import_key
nhwa1vdg1ti7qvmthr7zeployl63qd6tnzzk232q4c5m7iu6ubu
5Hq7yLvUwTX1xVn7GtbNsuq9uBkZLbLWXSfH566es2xnHdqBBRU
  true
```

Check the account balance via the following command.

```
unlocked >>> list_account_balances
nhwa1vdg1ti7qvmthr7zeployl63qd6tnzzk232q4c5m7iu6ubu
100 PIC
```

2. Upgrading the account to membership

The accounts are common ones by default which become node accounts only after upgrading to membership account. We upgrade the accounts through the following commands:

```
unlocked >>> upgrade_account
```

```
nhwa1vdg1ti7qvmthr7zeployl63qd6tnzzk232q4c5m7iu6ubu true
  {
    "ref_block_num": 33557,
    "ref_block_prefix": 362954262,
    "expiration": "2018-04-09T10:55:42",
    "operations": [[
        8,{
          "fee": {
            "amount": 10000000,
            "asset_id": "1.3.0"
          },
          "account_to_upgrade": "1.2.30288",
          "upgrade_to_lifetime_member": true,
          "extensions": []
        }
      ]
    ],
    "extensions": [],
    "signatures": [
      "1f04e8c647e24ba72aa4aae7b8832e419b26b52c3b720cda959f1569d
319a86f8146256a102382353109026ea04e4a7fa15bb5e5942113829f9ed7b39
30913f657"
    ]
  }
```

The upgrading needs a minimum of 100PIC service charges.

3. Upgrading the account to witness (consensus node account)

Upgrade the account to witness (consensus node account) via the following commands:

```
  unlocked >>> create_witness1
nhwa1vdg1ti7qvmthr7zeployl63qd6tnzzk232q4c5m7iu6ubu
PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj "some_url"
true
  {
    "ref_block_num": 33777,
    "ref_block_prefix": 729205876,
    "expiration": "2018-04-09T11:06:42",
    "operations": [[
        20,{
```

```
          "fee": {
            "amount": 10000000,
            "asset_id": "1.3.0"
          },
          "witness_account": "1.2.30288",
          "url": "some_url",
          "block_signing_key":
"PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj"
        }
      ]
    ],
    "extensions": [],
    "signatures": [
      "1f61431415e716fc0c0a9b01207daa9d828129d4eb167a8a305697027
b14701d7153b158e6f00de4706f4985510f22b73cd04116dd3e669011003ee9c
fce0c9fe3"
    ]
  }
```


The operation will be charged with a minimum of 100 PIC.

Check details of the new witness via the following commands:


```
  unlocked >>> get_witness 1.2.30288
  {
    "id": "1.6.12",
    "witness_account": "1.2.30288",
    "last_aslot": 0,
    "signing_key":
"PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj",
    "vote_id": "1:22",
    "total_votes": 0,
    "url": "some_url",
    "total_missed": 0,
    "last_confirmed_block_num": 0
  }
```


Among them, the id field shows the consensus account's ID while the total vote reveals the current poll.

4. Attracting votes to make it top 11 in poll

The new account is of no ballot. When the consensus account has

attracted sufficient votes to list at the Top 11, it is eligible for generating to a signature block.

We vote ourselves with our own account:

```
  vote_for_witness
nhwa1vdg1ti7qvmthr7zeploy163qd6tnzzk232q4c5m7iu6ubu
nhwa1vdg1ti7qvmthr7zeploy163qd6tnzzk232q4c5m7iu6ubu true true
  {
    "ref_block_num": 33947,
    "ref_block_prefix": 2988566179,
    "expiration": "2018-04-09T11:15:12",
    "operations": [[
        6,{
          "fee": {
            "amount": 105,
            "asset_id": "1.3.0"
          },
          "account": "1.2.30288",
          "new_options": {
            "memo_key":
"PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj",
            "voting_account": "1.2.5",
            "num_witness": 0,
            "num_committee": 0,
            "votes": [
              "1:22"
            ],
            "extensions": []
          },
          "extensions": {}
        }
      ]
    ],
    "extensions": [],
    "signatures": [
      "20228e21acd087519cdd3470e2cc557b9750d70d40ec213d22c465453
6ca72bdc81ec66aa4322d371e08fbb6453663bd0749c01db60890f77c68b8efc
1a46d21a5"
    ]
  }
```

vote_for_witness

nhwa1vdg1ti7qvmthr7zeployl63qd6tnzzk232q4c5m7iu6ubu

nhwa1vdg1ti7qvmthr7zeployl63qd6tnzzk232q4c5m7iu6ubu true true

In the above command, the four parameters represent voter account, candidate account, vote (true)/ cancel vote (false) and broadcasting (true).

Then check again the consensus account information:

```
  unlocked >>> get_witness
nhwa1vdg1ti7qvmthr7zeployl63qd6tnzzk232q4c5m7iu6ubu
  {
    "id": "1.6.12",
    "witness_account": "1.2.30288",
    "last_aslot": 0,
    "signing_key":
"PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj",
    "vote_id": "1:22",
    "total_votes": 0,
    "url": "some_url",
    "total_missed": 0,
    "last_confirmed_block_num": 0
  }
```

We found that the poll didn't change. The reason lies that Pi sets a 2-hour-period of vote-counting which updates the poll every 3rd hours.

```
  unlocked >>> get_witness
nhwa1vdg1ti7qvmthr7zeployl63qd6tnzzk232q4c5m7iu6ubu
  {
    "id": "1.6.12",
    "witness_account": "1.2.30288",
    "last_aslot": 0,
    "signing_key":
"PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj",
    "vote_id": "1:22",
    "total_votes": 11000000,
    "url": "some_url",
    "total_missed": 0,
    "last_confirmed_block_num": 0
  }
```

The votes for the account are still very small so that more votes are

needed from other users. The account starts to generate block when its poll is listed on top 11 in the system.

Users could vote via the voting option (under development) in cold wallet, while vip users could also vote in the CLI wallet mentioned before.

Enclosed: Illustration of vote-counting

The poll of witness account will be recounted in every vote-counting period. The total poll is the sum of votes and PIC balance in that account. The top 11 witness getting the most votes is entitled to generating a block.

Every account can vote to several witnesses and the PIC balance of that account will be counted into the poll. At present, each account could support a maximum of 101 witnesses, which might be altered in the future.

The network holds around 10 seconds for every 2 hours as maintenance period which is also the vote-counting period. That is, the votes of a witness updates every 2 hours. The vote-counting period might be altered later.

5. Binding the account with node

Suppose the account had sufficient votes to generate a block, what we need to do is binding the secret key of that account with a node which will be run to generate a block.

For running with self-compiling codes, we could make it by adding two configurations in the configuration file.

```
  # refer to the id fields in "get witness"
  witness-id = "1.6.12"
  # the generated secret key pairs on registration
  private-key =
["PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj",
"5Hq7yLvUwTX1xVn7GtbNsuq9uBkZLbLWXSfH566es2xnHdqBBRU"]
```

The configurations will be read when the node is running. And the signature of the secret key will be verified on generating a block. As the secret key pair is kept on server, the security of the server must be guaranteed in case of property loss.

The running of consensus node with Docker is essentially like running it with codes. The only difference is that the files are not directly edited and configured with Docker, but transmitting secret key pair via parameter by Docker. How they are transmitted will be explained in the next passage.

6. Running the node

The running method for self-compiling code is the same as mentioned above, only did the configuration file is modified.

The preparations for running consensus nodes with Docker are the same as mentioned above. But different parameters are needed on running:

```
  docker run \
      -d --name witness \
      -v /opt/data:/data \
      -p 8010:8010 \
      -p 40010:40010 \
      -e "RPC_ENDPOINT=0.0.0.0:8010" \
      -e "P2P_ENDPOINT=0.0.0.0:40010" \
      -e
"SEED_NODES=52.221.188.236:40010#13.229.67.149:40010#52.77.249.1
28:40010#54.169.77.194:40010#54.255.181.71:40010#54.255.223.90:4
```

```
0010#:52.77.219.12240010#54.169.201.229:40010#52.221.194.247:400
10#54.251.135.96:40010#52.221.201.240:40010#" \
     -e "WITNESS_IDS=1.6.12" \ # see the id fields in "get witness"
     -e
"SIGN_KEYS=PIC6TEeWqG7xBSqyz5TFaubU48SBed8GuhkzYFjGmdSULtDXq5vCj
,5Hq7yLvUwTX1xVn7GtbNsuq9uBkZLbLWXSfH566es2xnHdqBBRU" \ # secret
key pair on registration
     -e "WITNESS_ARGS=--disable-account-history-plugin
--disable-market-history-plugin
--disable-incentive-history-plugin
--disable-transaction-record-plugin" \
     wallex64/pi
```

In the above command, "--disable-witness-plugin" is no longer in the "WITNESS_ARGS" which is an imperative plug-in for running consensus nodes to generate a block.

And we have increased nearly all disable options. For security thinking, a consensus node should not be disclosed. So we disabled the plug-in by default.

It is recommended to run consensus node with Docker and ensure its security and stability by closing RPC interface.

7.3 Node Income

7.3.1 Service node

The service node is of no income. But running service nodes improves the response speed and stability of the application. We recommend running its own service nodes on self-built applications, as it will improve the using experience as well as adjust services according to its own scale.

7.3.2 Consensus node

Every time a block is generated on the consensus node, a specific amount of PIC award is obtained. The awards are accumulated in the bonus pool which can be extracted via a special transaction after some time.

8. Smart Contract

π provides rich smart contract interfaces for users to create various types of smart contract applications on the blockchain. For example:

1. The user creates a smart contract for the locked construction capital. When the contraction capital expires, the new contraction capital is established automatically, keeping the user from suffering issuance loss due to forgetting the renewal.

2. The user could set a smart contract. When receiving the income from incremental issuance, it automatically transfers the income to other designated users. It helps the merchants construct the feedback system sharing their income to the valuable customers to enhance the service quality.

With π's rich interfaces, we will develop its infinite potential together through our visions.

Chapter 3 Conclusion

Through analyzing and summarizing the current application situation of the existing blockchain and combining its application characteristics, this white paper proposes a new blockchain platform π.

π uses a brand-new IPoS consensus mechanism, greatly optimizing the energy consumption and efficiency of PoW in Bitcoin and giving consideration to fairness at the same time.

The innovative π issuance incentive algorithm distributes the income fairly while ensures the efficiency, contributing to a healthy development of the network.

Enclosed: Discovery Application

Discovery is a social and gaming application established on π, which achieves some of its functions via the smart contract.

Discovery has concepts of treasure hiding and mining. Treasure hiding corresponds to locking the construction capital, while treasure mining corresponds to getting the incremental issuance income. The yield of treasure mining is estimated according to the amount of the construction capital. On this basis and with the combination of digital asset spending, π Pay has been developed.

Using the acceleration mechanism of π and through the game rules set by Discovery, the users could invite more other users to enhance the treasure mining speed of the main account.

There is only an account address on the corresponding π blockchain in Discovery. Through executing the same issuance mechanism and algorithm as π, Discovery achieves to manage π assets and data of internal users and provide API interface the same as π. With a set of API, the third-party developers could connect their own applications with π blockchain and Discovery, and put equally on the internally generated address from π blockchain and Discovery.