Belrium

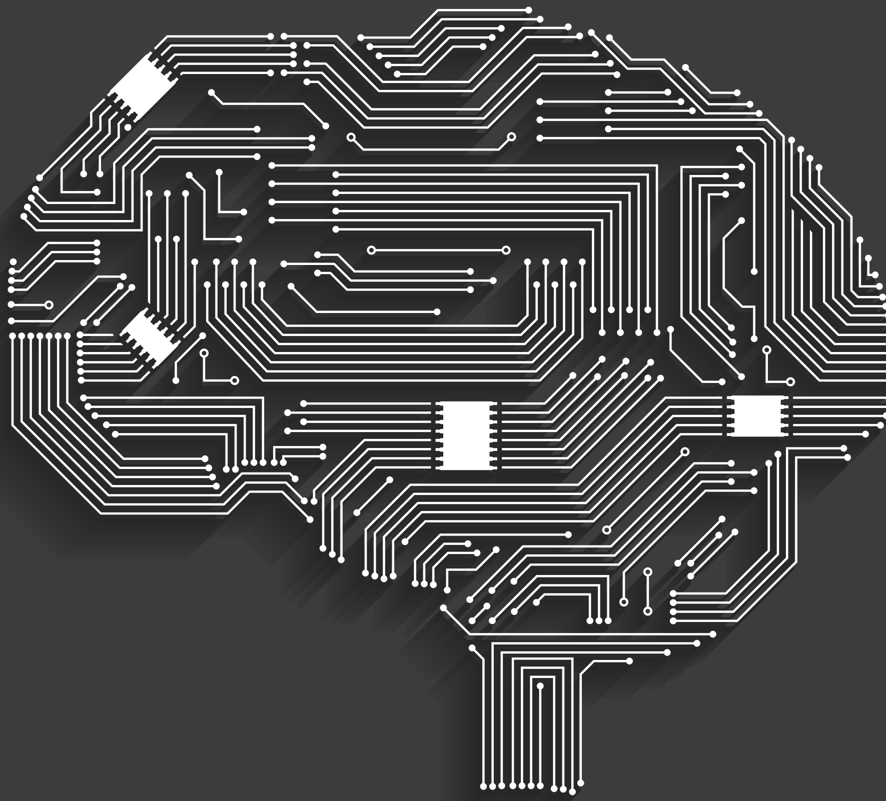
KYC Based Blockchain
Technical Whitepaper

Belrium – KYC Based Blockchain Whitepaper

Revision History

Date	Version	Description
06/26/2017	1.0	Whtepaper 2.0

Contents



Abstract

Know Your Customer (KYC) process is integral part of client on boarding for any institution involve in financial or value added services. The institution face many problem in client on boarding due to external and internal influential factors. As per research by a well known research organisation, institution loses 20 to 30% of business due to complex KYC process and regulations. The mushrooming of KYC companies are not much of help as they act only as data collection point. The task of investigation and due diligence is taken care by onboarding teams at institution, given the business risk and reputation.

KYC process are repetitive in nature, but as the institution doesn't have any interlinking centralized solution , it leads to duplication of information and administrative overhead. The administrative overheads leads to cost.

There have been initiatives by private entities like Society for Worldwide Interbank Financial Telecommunication (SWIFT) and government but all these initiatives are limited to specific area like financial services which is not much of help for other value added service providing institution.

BELRIUM - The KYC Based Blockchain*** is based on the fusion of two blockchain working independently

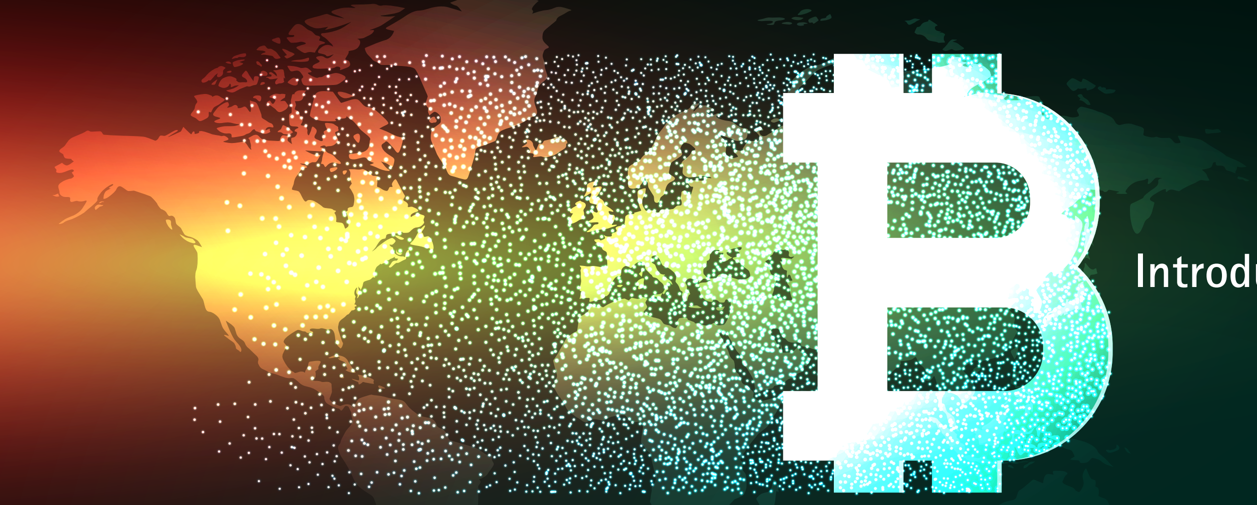
1. BELRIUM Blockchain - A public blockchain that anyone in the world can read, send transactions and expect to see them included if they are valid.

2. KYC Blockchain - A fully private blockchain where write permissions are kept centralized with the organisation. Read permissions will be public via REST based API.

Both the blockchain are based on Delegated Proof of stake(DPOS), can effectively overcome all the challenges faced by the institutions in KYC process.

BELRIUM is developed on NodeJs, a open source, cross-platform programing language, built on Chrome's JavaScript V8 engine for fast and scalable networking applications programing. Being an interface to the V8 JavaScript runtime, it enables event-driven programming to the web servers through super-fast JavaScript interpreter.

The KYC Blockchain will be developed on Hyperledger Fabric.Hyperledger Fabric is different for other blockchains. Rather than proof of work it uses membership service provider. Hyperledger Fabric also offers several pluggable options. Ledger data can be stored in multiple formats, consensus mechanisms can be switched in and out, and different MSPs are supported.



Introduction

Worldwide, After 9/11 KYC become very important process of client on boarding and diligence. There has been upsurge in initiatives by government and private sector to develop institution or governing body to formalize KYC process like SWIFT . The solutions has very limited vision and applicability.

Belrium - KYC Based Blockchain is a decentralized KYC exchange platform built on blockchain and smart contracts. The Belrium - KYC Based Blockchain places an emphasis on the institutions involved in KYC based process. The Belrium - KYC Based Blockchain platform is designed to disrupt and replace the traditional KYC process models by providing a transparent, focused solution for institution. It will provide the cost effective and time saving decentralized solution for KYC process.

Belrium - KYC Based Blockchain leverages blockchain and smart contracts, thus eliminating the complexity of existing KYC based system. The network is regulated by the authenticated agencies in private blockchain network, taking care of the most common issue faced by KYC based institutions i.e. lack of centralized KYC platform and KYC fraud.

BELRIUM - The KYC Based Blockchain based on the fusion of two blockchain working independently.

In the first part of document we will discuss about the public blockchain i.e. BELRIUM Blockchain and in second part we will discuss about the private blockchain i.e. KYC Blockchain

PART 1 – Belrium Blockchain

1. Belrium

The blockchain protocol can be decomposed into three distinct protocols:

- The network protocol discovers blocks and broadcasts transactions.
- The transaction protocol specifies what makes a transaction valid.
- The consensus protocol forms consensus around a unique chain.

Belrium has evolved from the the basic blockchain protocol to provide holistic solution of implementing KYC compliance, set and govern by 3rd party institution. In most of the cases the 3rd party is Government.

It implements a generic wrapper around the blockchain protocol. This wrapper interacts with KYC system after discovery and before broadcast of transection.

1.1 Mathematical Representation

A blockchain protocol is defined as monadic implementation of concurrent mutations of global state. It means that operators acting on this global state can be called as "blocks". The blocks acting on the genesis state forms a tree structure. A global, canonical, state is defined as the minimal leaf for a specified ordering. This suggests the following abstract representation:

- Let $(S, \leq)$ be a totally ordered, countable, set of possible states.
- Let$\varnothing \notin S$ represent a special, invalid, state.
- Let $B \subset SS \cup \{\ominus\}$ be the set of blocks.

The set of valid blocks is $B \cap S^S$

The total order on S is extended so that $\forall s \in S, \varnothing < s$. This order determines which leaf in the block tree is considered to be the canonical one. Blocks in B are seen as operators acting on the state. All blockchain like Bitcoin, Litecoin, Peercoin etc. can be fully determined by the tuple:

$$(S, \leq, \varnothing, B \subset SS \cup \{\ominus\})$$

The networking protocol is fundamentally same for all blockchain networks but the mining algorithm differs .

$$P = \{(S, \leq, \varnothing, B \subset S(S^*P) \cup \{\ominus\})\}(T)$$

1.2 The KYC wrapper

The former mathematical representation of block tells about how blocks are created and managed over the network. It doesn't talk about how transactions are verified and validated . The KYC wrapper works in conjugation with three factors

1. Transaction initiator
2. Transaction receiver
3. KYC verifier protocol

The Transaction initiator is the wallet user who is starting the transaction. The Transaction receiver is the end beneficiary of transaction. The KYC verifier protocol acts as a communication interface for between KYC and Belrium. It helps to verify that the transaction initiator and receiver are legitimate users of the system. If the initiator or receiver is not the legitimate users than it flush out the transactions from the blockchain.

1.2.1 Clock

A timestamp is carried by every blocks which is visible over the network. The value of timestamp is set at the time of generation. It provides extra level of security. Blocks that appear to come from the future are buffered if their timestamps are within a few minutes of the system time and rejected otherwise.

1.2.2 Chain selection algorithm

Belrium- The KYC based blockchain maintains a single chain of blocks rather than a full tree of blocks. This chain is only overwritten if the client becomes aware of a strictly better chain.

Maintaining a single chain is easy in comparison to maintaining a tree. Maintaining a tree would be more parsimonious in terms of network communications but would be susceptible to denial-of-service attacks where an attacker produces a large number of low-scoring but valid forks. Yet, it remains possible for a node to lie about the score of a given chain, a lie that the client may only uncover after having processed a potentially large number of blocks. However, such a node can be subsequently ignored. Fortunately, a protocol can have the property that low scoring chains exhibit a low rate of block creation. Thus, the client would only consider a few blocks of a "weak" fork before concluding that the announced score was a lie.

1.2.3 Security

Belrium uses cryptographic hashing in order to secure all aspects of the system. The system uses EdDSA as it provides a much faster mechanism for hashing.

1.2.3.1 CKVS Authnetication

Belrium - The KYC based blockchain is a private block chain were each and every miner are closely monitored. All miners are induced in the system are pre verified.

1.2.3.2 Key pair

A keypair is consists of a private key and a public key. A private key is a piece of information known only to the owner of the key. The public key is derived from the private key and can be used to validate that the private key belongs to the owner, but not provide access to the owner's private key. Elliptic curve cryptography is used to generate cryptographically secure key pairs.

1.2.3.3 Second pass phrase

Belrium offers an additional layer of security for the user. Using a specific class of transaction, the user can register a second pass phrase that is associated with the kp. This relationship requires that all subsequent transactions to be signed using the second pass phrase in order to be considered valid. The process of generating the second key pair is the same as the one for the main key pair.

1.3 Functional Representation

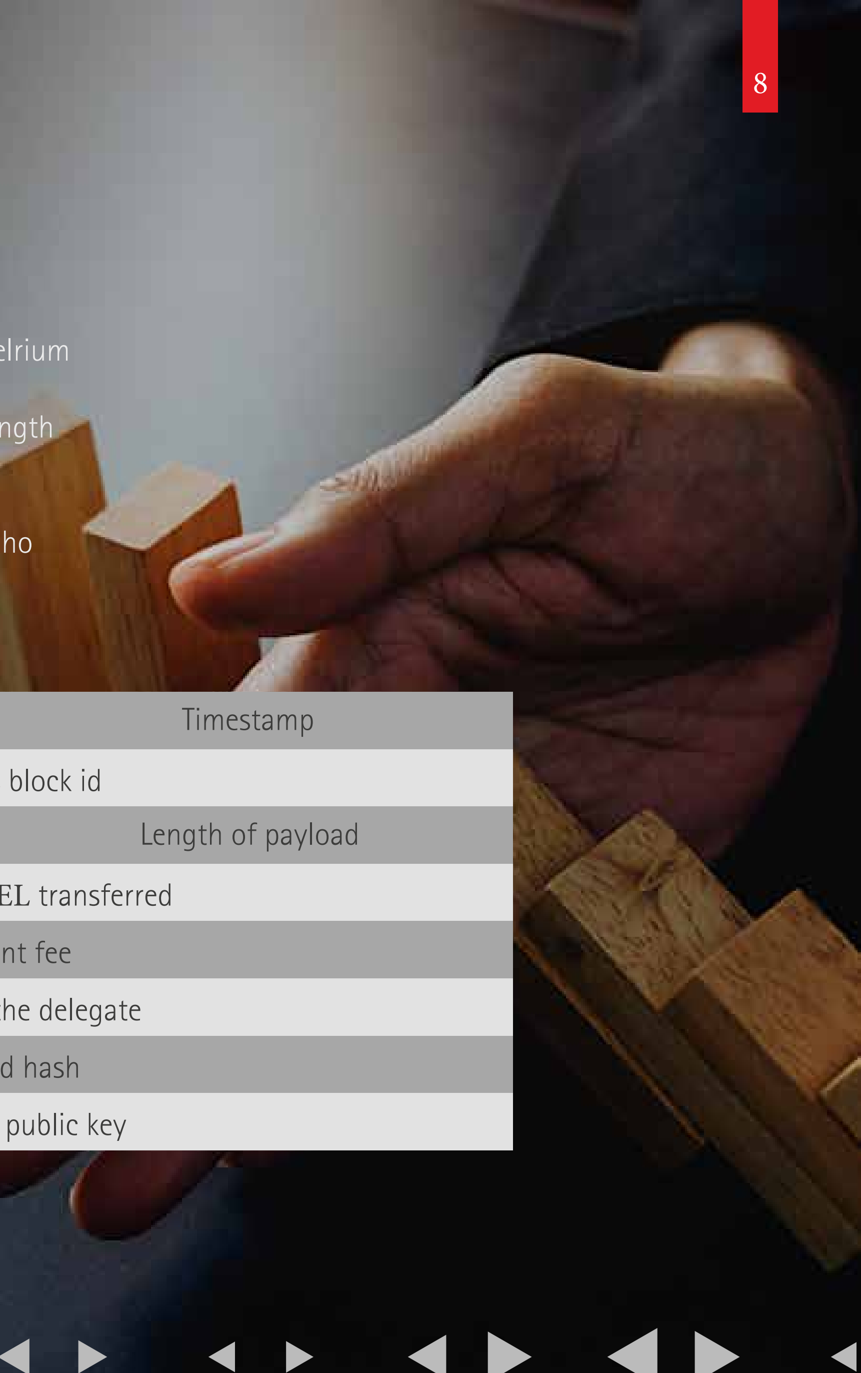
1.3.1 Block

A blockchain is composed of blocks, and a block is composed of a header and a list of confirmed transactions. When a delegate is assigned a slot and has a node running, that delegate generates the next block and confirms upto a pre decided number of transactions from the transaction pool. These confirmed transactions will be added the payload of the block and subsequently signed into that block.

1.3.1.1 Block header

The block header contains all the information about the block. The following fields compose the block header:

- A 32 bit integer identifying the version of the block
- A 32 bit epoch timestamp of when the block was created
- The 64 bit Id of the previous block
- A 32 bit integer corresponding to the number of transactions processed in the block
- A 64 bit integer corresponding to the total amount of Belrium transferred
- A 64 bit integer corresponding to the total Belrium charged as fee
- A 64 bit integer corresponding to the Belrium reward for the delegate
- A 32 bit integer corresponding to the length the payload
- The 256 bit hash of the payload
- The 256 bit public key of the delegate who generated the block

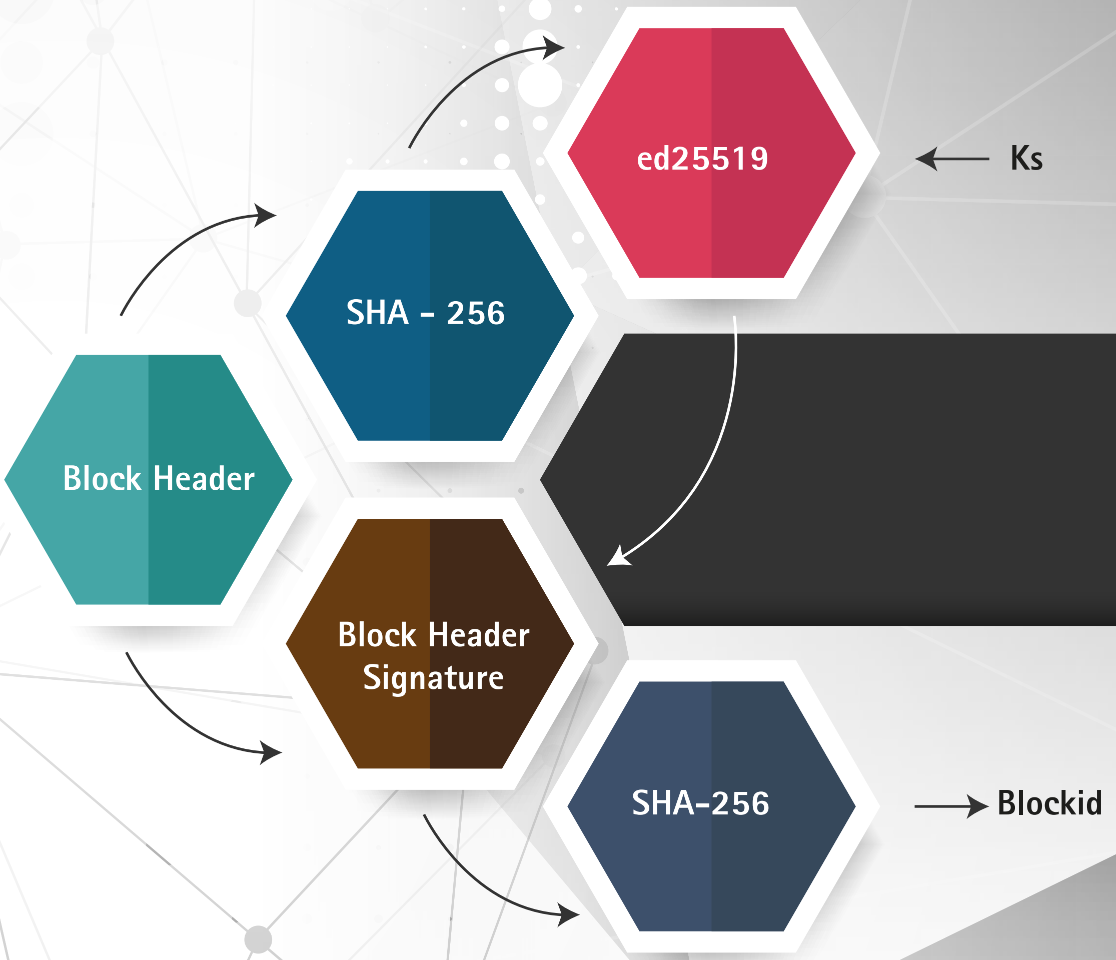


Version	Timestamp
Previous block id	
Number of transactions	Length of payload
Amount of BEL transferred	
Amount fee	
Reward of the delegate	
Payload hash	
Delegate's public key	

JSON Representation of Block

```
{
    "id": "15787022670460703397",
    "version": 0,
    "timestamp": 23039010,
    "height": 1574052,
    "previousBlock": "4576781903037947065",
    "numberOfTransactions": 0,
    "totalAmount": 0,
    "totalFee": 0,
    "reward": 500000000,
    "payloadLength": 0,
    "payloadHash": "e3b0c44298fc1c149afbf4c8996fb92427ae4",
    "generatorPublicKey": "c0ab189f5a4746725415b17f8092e",
    "blockSignature": "c6b2bcc960066be078efbfffed625f615",
    "totalForged": "500000000"
 }
```

The process for signing the block header is the same as the process for signing a transaction. A SHA-256 hash of the block header is generated, and signed using the key secret of the delegate. Once the block header has been signed, the system generates the blockId following the same logic as transactions. The completed block header is hashed using SHA-256 and the first 8 bytes of the hash are reversed and used as the blockId. A signed block generates its blockId using the following flow:



1.3.1.2 Block payload

The payload of the block is comprised of up to 25 unconfirmed transactions present on the system signing the block. The maximum number of available transactions up to this limit will be included, provided that the payload for a transaction doesn't exceed the max size for each transaction type. These max sizes are listed below:

Type	Name	Size(In Bytes)
0	SEND	220
1	SIGNATURE	149
2	DELEGATE	201
3	VOTE	2326
4	MULTI	1223

The max size of a block payload can then be determined as 58150 bytes with if every transaction is type 3 and contains the maximum number of assets. A data block is composed using the gathered unconfirmed transaction data blocks and signatures. The system then hashes the combined transactional data blocks to generate the block payload.

1.3.1.3 Block Generation

Block Generation occurs at every 15 seconds within the Belrium network using DPoS consensus. An active delegate is an account that has been given the right to generate blocks by a process of election from other stakeholders. Block generation requires 51% of peers to maintain broadhash consensus.Once broadhash consensus is established the node will generate a block.

1.3.2 Transactions

In BELRIUM, transactions are designated by type. These transaction types include:

Type	Name	Detail
0	SEND	Transmit funds to a specified Belrium address
1	SIGNATURE	Register a second secret
2	DELEGATE	Register a delegate
3	VOTE	Submit vote(s) for delegates
4	MULTI	Multisignature registration

1.3.2.1 Transaction Signing

Every transaction, regardless of the type, must be signed by the sender prior to being accepted by the network. The process of signing the transaction is identical for every transaction. First, a data block representing the transaction must be generated. Each data block contains a specific set of standardized information. Additional information contained in the data block will differs depending on the type of the transaction. The following fields must be present in all types of transactions:

1. A 8 bit integer identifying the type of the transaction
2. A 32 bit epoch timestamp of when the transaction has been created
3. The 256 bit public key of the issuer of the transaction
4. A 64 bit integer representing the amount of Belrium to be transferred
5. A 8 bit boolean identifying the valid sender
6. A 8 bit boolean identifying the valid receiver

The sender and receiver values are set from CKVS system. If any of the flag are marked as false then transaction will be not included in block.

1.3.3 Consensus

BELRIUM uses Delegated Proof of Stake(DPoS) as the consensus system. Delegates generate all of the blocks within the system and these delegates are chosen through a highly competitive election system driven by stakeholders. The number N (currently N = 101) of delegates are chosen to forge by all stakeholders. Each stake holder may vote for up to 101 delegates, and the weight of the vote depends on the amount of Belrium the stakeholder possess. A stakeholder can vote for a delegate using a vote transaction.
Consensus is a required aspect of any blockchain system. It serves a vital purpose for the system where there are many nodes and all nodes must agree on the integrity of the data. All nodes participating must agree on what transactional data is legitimate in order to move the blockchain forward.

1.3.3.1 Delegate

A delegate is a special type of account that has registered using a delegate registration transaction. These accounts have a special purpose within Belrium as they are allowed to generate blocks for the system provided that the delegate has been allocated enough stake by other users of the system. Any account may become a delegate, but only accounts with the required stake are allowed to generate blocks.

1.3.3.2 Delegate round

A round within the system is exactly N blocks in length (N is identical to the total forging delegates). During one round, each delegate will forge exactly one block. If an elected delegate cannot forge during a round another delegate will forge their block instead. At the beginning of each round, each delegate is assigned a slot indicating their position in the block generation process. Once a node with an enabled active delegate has forged a block, the node associated with the delegate includes up to 25 transactions into the block, signs it and broadcasts that block to the network. Once the block has reached the network, the next delegate will forge in the slot assigned to them.

1.3.4 Rewards

In Belrium, there are various incentives provided to make running a node appealing. The first of these is the block generation reward and the other reward is the accrual of fees for securing the network as an active delegate for the round in which that delegate participates.

1.3.4.1 Block Reward

Belrium rewards the block generator a fixed amount of tokens per block successfully generated and accepted by the system. In Belrium system, all active delegates that successfully participate are rewarded 5 Belrium coin for securing the network.

1.3.4.2 Transection Reward

A predetermined charge, defined by using smart contract is charged as transaction fee and awarded to the transaction processing delegate.

1.3.5 Peers Communication

Peers communication serves a vital function within the Belrium network. The peering mechanisms provide the required architecture to facilitate network consensus, block propagation and transaction propagation.

1.3.5.1 System headers

Within the Belrium network system headers are used to identify full nodes and provide a basic set of information about the software running on the system. During peers communications these headers are added to all messages sent between peers.

Version	Timestamp
Previous block id	
Number of transactions	Length of payload
Amount of BEL transferred	
Amount fee	
Reward of the delegate	
Payload hash	
Delegate's public key	

1.3.5.2 Broadhash Consensus

In the DPoS system, delegates are assigned slots based on timestamp and will attempt to forge a block when the system designates that delegate slot as ready. Broadhash consensus ensures that a majority of available peers agree that it is acceptable to forge. All peers with the same blocks will produce the same broadhash and propagate that information via the system headers.

1.3.5.3 Block Propagation

Blocks are made in a decentralized fashion and must be sent to all nodes found on the network in order to establish consensus. When a block is generated, it is broadcast to peers which broadcast that block to other peers. Without block propagation, the system would grind to a halt and the blockchain would cease to be functional.

1.3.5.4 Broadcast queue

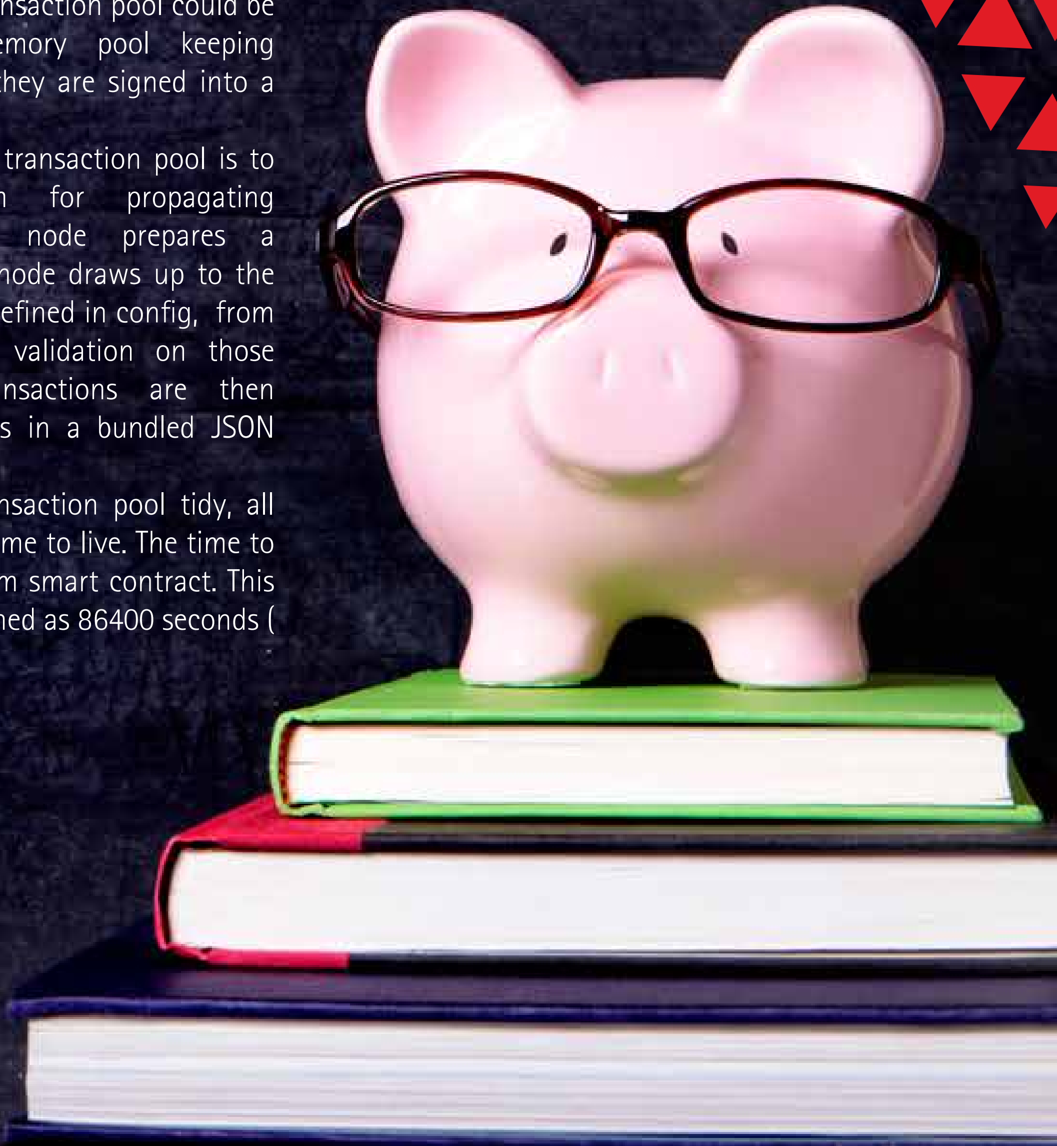
The broadcast queue works by grabbing upto N number of transactions (defined in Belrium config) from the transactions pool and aggregating them into a bundle. This bundle is then broadcast to the network on an interval, which is currently specified as every 5 seconds.

1.3.5.5 Transaction Pool

The transaction pool provides the Belrium network a very robust solution for preserving unconfirmed transactions that have overflowed into the next block.The transaction pool could be thought of as a memory pool keeping transactions ready until they are signed into a block.

The second usage of the transaction pool is to provide a mechanism for propagating transactions. When a node prepares a transaction bundle, that node draws up to the number of transactions defined in config, from the pool and performs validation on those transactions. These transactions are then broadcast to other nodes in a bundled JSON object.

In order to keep the transaction pool tidy, all transactions are given a time to live. The time to live can be controlled from smart contract. This default time to live is defined as 86400 seconds (24 Hour).



2. Belrium – Seed Protocol

As all blockchain technology starts with genesis block, Belrium - The KYC based block chain also starts with genesis block.

2.1 Economy

2.1.1 Coins

The Initial coins offering will be done through a crowd sale by Belfrics, which will later be replaced on a 1:1 ratio with Belrium coin. The Belrium coin will be referred as BEL. The BEL coins will have a total limit of 21 million, divisible up to 5 decimal places.

2.1.2 Mining and signing reward

We know that to successfully manage the decentralized currency system requires incentivization to participants. The active participants makes the system secure.

In seeding protocol the, the miners will be rewarded 5 BEL for each block mined. The system will also offer reward 0.001 BEL for signing a block. One block can have maximum 25 signed blocks. The number of signature per block may be increased.

Some percentage per signed block will be charged by system for KYC compliance.

2.1.3 Lost coins

The address which are not active for one year(as determined by timestamp) they will be marked as inactive. The address will lose their earning rights till they become active.

If a account is inactive for more than 5 years then the account will be fortified the amount will be put in to reward pool.

2.1.4 Amendment rule

Amendments are adopted over By monthly in first year . In later years , depending on frequency of amendments the adaptation period can be chane like Quarterly or half yearly ,

2.2 Conscious Protocol Delegated Proof of stake

2.2.1 Overview

The Belrium-KYC based Blockchain is based on DPOS algorithm, a blockchain based distributed computing platform. DPOS algorithm allows smart contracts i.e. distributed computer programs that can facilitate on-line smart contractual agreements in a cryptographically secure manner. DPOS algorithm is open-source package written in NodeJS and adopted by many institutions for Blockchain development.

The smart contracts enables the Belrium-KYC based Blockchain as a truly transparent and decentralized KYC serving exchange. Smart contracts are essentially computer programs that run on a distributed public ledger, therefore ensuring their result is always consistent, transparent and cannot be manipulated.

2.2.2 Clock

The DPOS protocol imposes fixed time delays between blocks. In principle, each block is mined by stakeholder after a fixed time. The Belrium has delay time of 15 second. The stakeholder receiving the highest priority may mine the block 15 second after the previous block. The stakeholder receiving the second highest priority may mine the block 30 second after the previous block, the third, 45 second and so on.

This guarantees that every active delegate gets a fair chance to create node. We can avoid the denial of service attack which can be done by tricking a node into verifying a very long chain claimed to have a very high score.

2.2.3 Generating the block

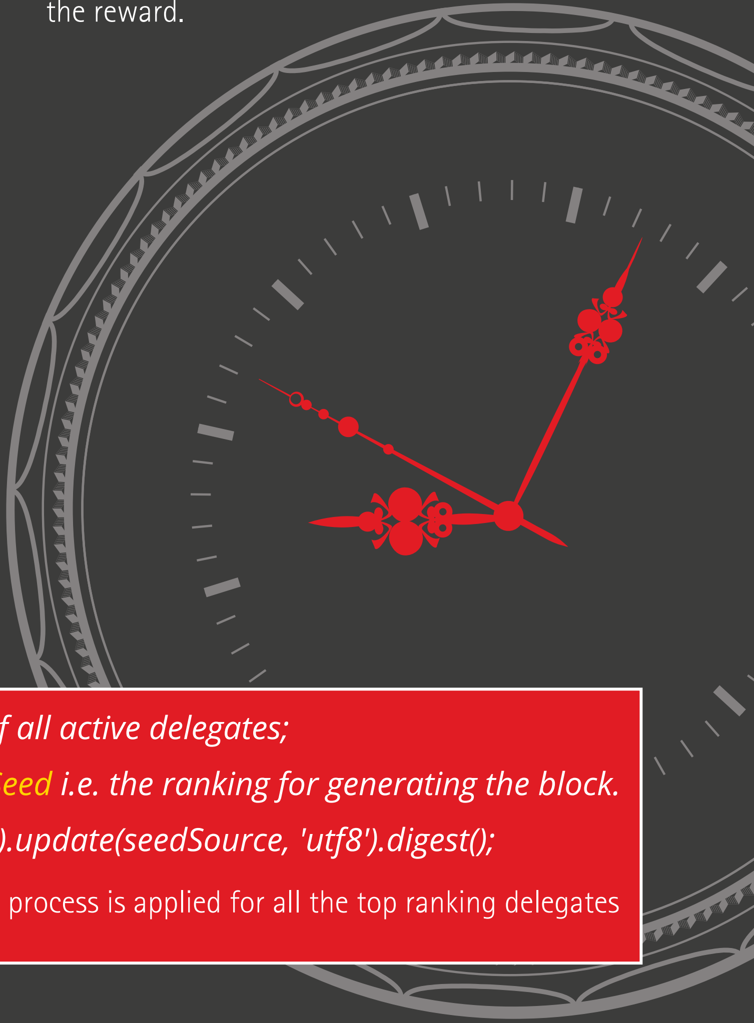
The blocks are generated by the miner on the round robin basis. On start of the every round , each delegate is assigned a slot depending on the sloat generating algorithm. Each node forged by active delegate can have maximum of 25 transaction.

2.2.4 Miner forging Rank

The miners forging rank is decided by the combination of functionality. First a seed Source is determined by

2.2.5 Mining blocks

Block Generation occurs every 15 seconds within the BELRIUM network using DPoS consensus. A delegate is an account that has been given the right to generate blocks by a process of election from other BELRIUM holders. If a delegate fails to generate the block then next delegate will get the chance to generate the pending node and claim the reward.



seedSource = Length of Block / Number of all active delegates;

The seedSource is used to get the currentSeed i.e. the ranking for generating the block.

currentSeed = crypto.createHash('sha256').update(seedSource, 'utf8').digest();

After every round of block generation the same process is applied for all the top ranking delegates

2.2.6 Signing block

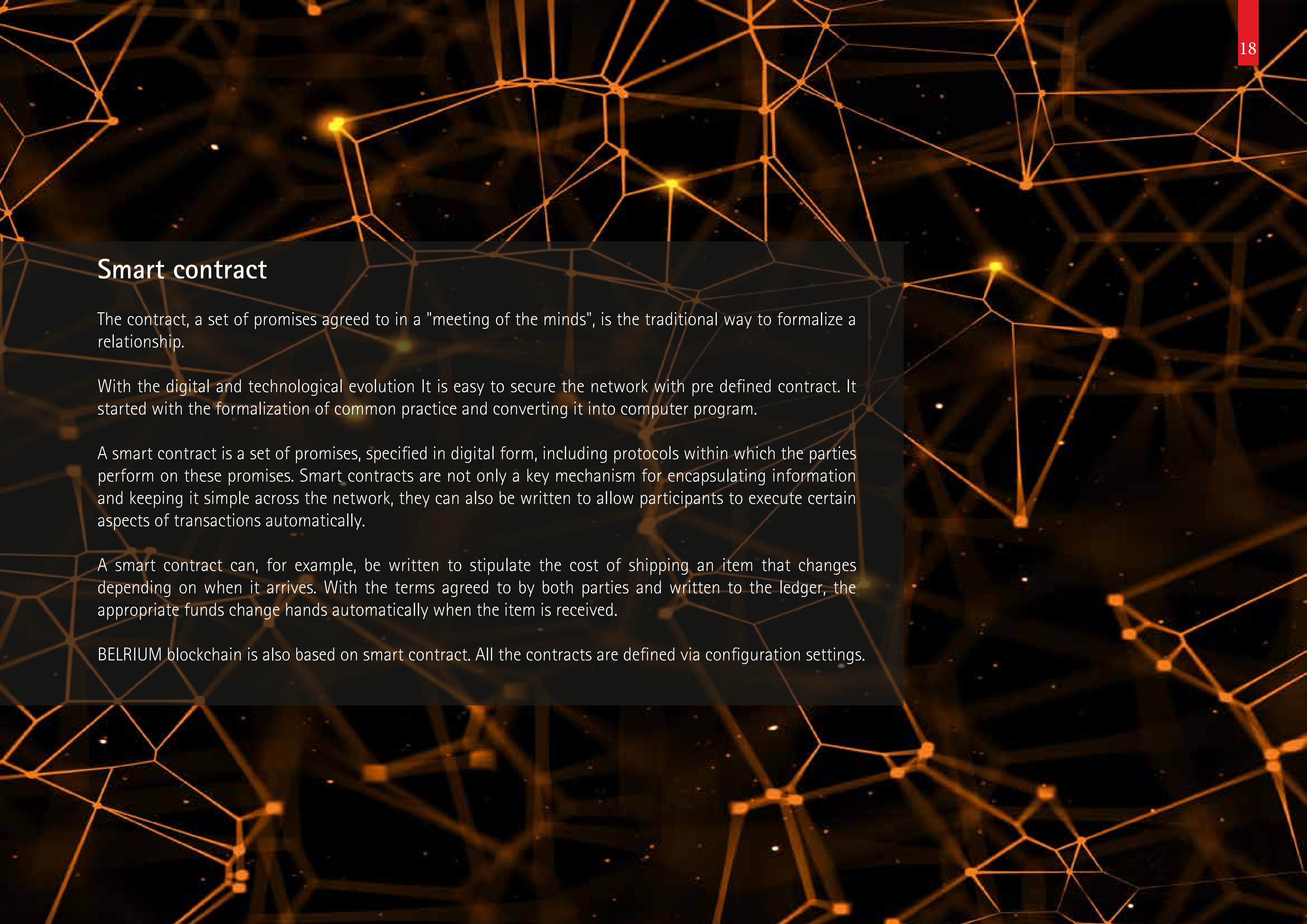
Active delegate generates the block as per their term. The verification is done by nodes as whole. If 51 % of the active delegate agree with the transaction and 49% don't then it will cause fork or stuck node.

2.2.7 Weight of the chain

TThe weight of a chain is determined by the number of signature in a node.

2.2.8 Denunciation

Denunciation is a process of avoiding the double attack. A miner attaches a denunciation in his block as a confirmation. Each block is signed by miner. The denunciations act as second signature.
Anyone participating miner can announce the malfeasance in the chain.



Smart contract

The contract, a set of promises agreed to in a "meeting of the minds", is the traditional way to formalize a relationship.

With the digital and technological evolution It is easy to secure the network with pre defined contract. It started with the formalization of common practice and converting it into computer program.

A smart contract is a set of promises, specified in digital form, including protocols within which the parties perform on these promises. Smart contracts are not only a key mechanism for encapsulating information and keeping it simple across the network, they can also be written to allow participants to execute certain aspects of transactions automatically.

A smart contract can, for example, be written to stipulate the cost of shipping an item that changes depending on when it arrives. With the terms agreed to by both parties and written to the ledger, the appropriate funds change hands automatically when the item is received.

BELRIUM blockchain is also based on smart contract. All the contracts are defined via configuration settings.

3.1 Active delegates

Every delegate is placed at a specific position on the delegate ranking list. The number of votes determines that position. The number of active delegates can be determined by configuration settings.

All delegates with a rank between 1 and N are active delegates. All other delegates with a rank over N (N+1- ∞) are classified as standby delegates.

In order to determine the delegate rank position, BELRIUM - The KYC based Blockchain uses a decentralized voting mechanism. Users can vote for any delegates registered on the network by paying a pre determined fee.

The default count of active delegate is 101.

3.2 Fee

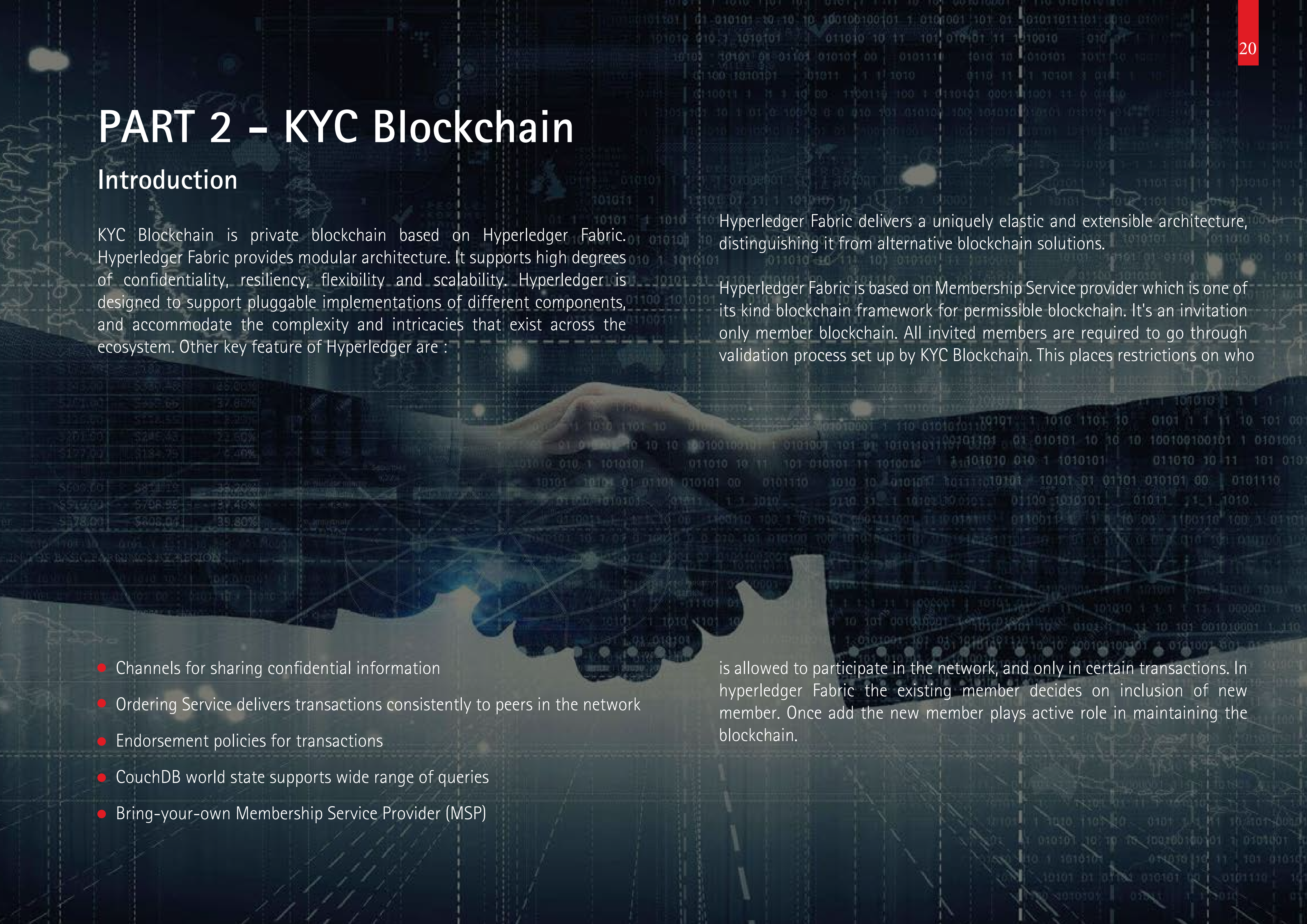
Depending on the height of chain and type transaction fees can be charged. Default value for lifetime is defined as below

Type	Name	Fee detail
0	SEND	10000000
1	SIGNATURE	500000000
2	DELEGATE	6000000000
3	VOTE	100000000
4	MULTI	500000000

3.3 Reward

In addition to transaction fee, delegates get forging rewards. The forging rewards occur at a fixed rate per block, and change over the course of the network's lifetime . It can be set as associative JSON array. The first value defines the height of blockchain and second parameter determines the reward for forging

```
rewards: [
        { height: 1,        reward: 0},
        { height: 10,       reward: 100000000},
        { height: 11,       reward: 30000000},
        { height: 12,       reward: 20000000},
        { height: 13,       reward: 100000000},
        { height: 640000,   reward: 110000000},
  ]
```



PART 2 - KYC Blockchain

Introduction

KYC Blockchain is private blockchain based on Hyperledger Fabric. Hyperledger Fabric provides modular architecture. It supports high degrees of confidentiality, resiliency, flexibility and scalability. Hyperledger is designed to support pluggable implementations of different components, and accommodate the complexity and intricacies that exist across the ecosystem. Other key feature of Hyperledger are :

- Channels for sharing confidential information
- Ordering Service delivers transactions consistently to peers in the network
- Endorsement policies for transactions
- CouchDB world state supports wide range of queries
- Bring-your-own Membership Service Provider (MSP)

Hyperledger Fabric delivers a uniquely elastic and extensible architecture, distinguishing it from alternative blockchain solutions.

Hyperledger Fabric is based on Membership Service provider which is one of its kind blockchain framework for permissible blockchain. It's an invitation only member blockchain. All invited members are required to go through validation process set up by KYC Blockchain. This places restrictions on who is allowed to participate in the network, and only in certain transactions. In hyperledger Fabric the existing member decides on inclusion of new member. Once add the new member plays active role in maintaining the blockchain.

Technical advantage of Hyperledger Fabric v1.0 over other blockchain

A. Permissioned Membership

Hyperledger Fabric is distinguished as a platform for permissioned networks, where all participants have known identities. The use of permissible network depends on the level of data security and protection regulations. Mostly in financial and healthcare industry, permissible network is used as it require high level of data security.

B. Performance, scalability, and levels of trust

Hyperledger Fabric is built on a modular architecture that separates transaction processing into three phases:

1. Distributed logic processing and agreement ("chain code")

2. Transaction ordering

3. Transaction validation and commitment.

The primary advantage of modular architecture is Fewer levels of trust and verification are required across node types, and network scalability and performance are optimized.
The working of transaction processing in Hyperledger Fabric can be summarised as below

1. Endorsing peer gets the proposal from applicant

2. As per the endorsement policies the number of endorser are decided to confirm the proposal. After deciding the number of endorser, the combination of endorsers are decided for signing the proposal. The chain code for proposal is executed in network peer with read/write set to simulate the proposal by endorser.

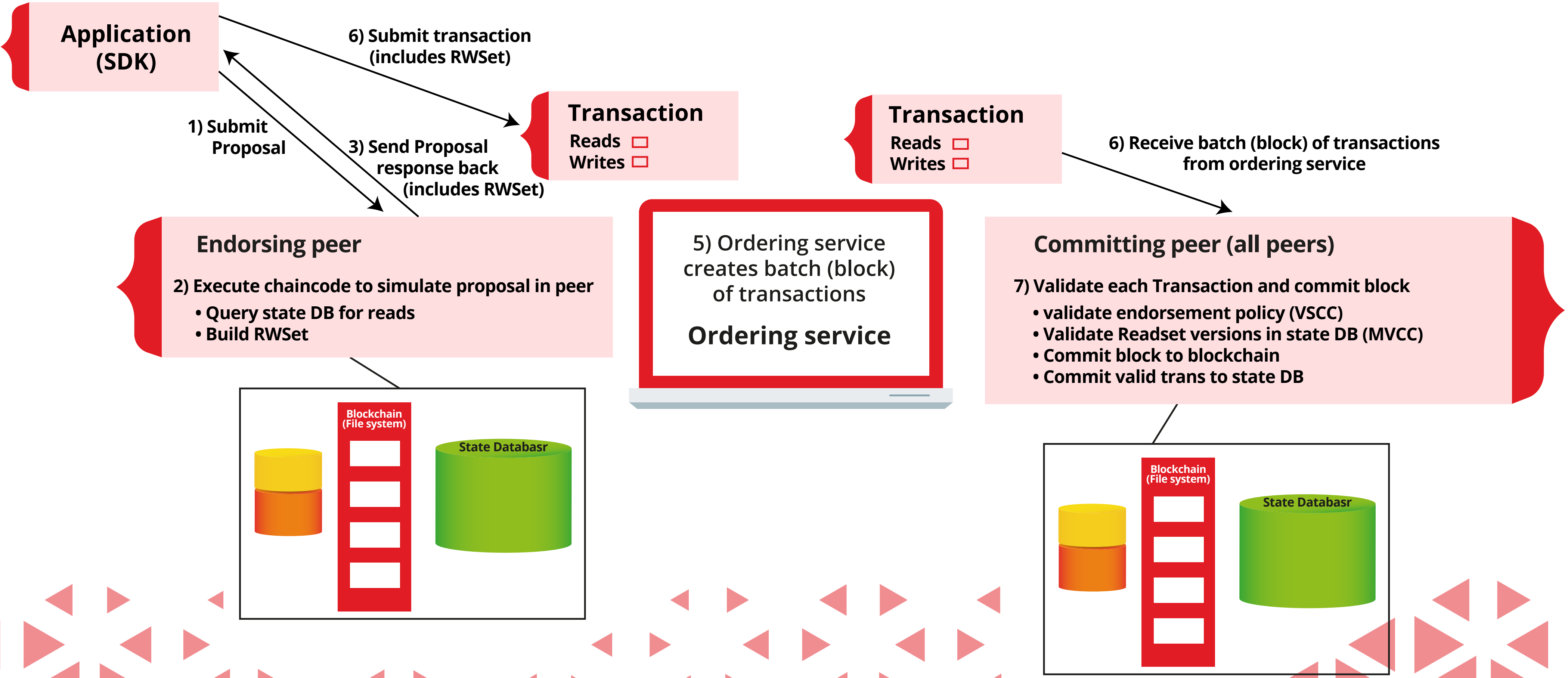
3. Then the endorsing peers send back the signed proposal responses (endorsements) to the application.

4. The application submits the transactions and signatures to the ordering service

5. creates a batch, or block, of transactions and delivers them to committing peers.

6. When a committing peer receives a batch of transactions, for each transaction it validates that the endorsement policy was met and checks in the read/write sets to detect conflicting transactions.

7. If both checks pass, the block is committed to the ledger, and the state updates for each transaction are reflected in the state database.


Application (SDK)
6) Submit transaction (includes RWSet)
1) Submit Proposal
3) Send Proposal response back (includes RWSet)
Transaction
Reads □
Writes □
Transaction
Reads □
Writes □
6) Receive batch (block) of transactions from ordering service
Endorsing peer
2) Execute chaincode to simulate proposal in peer
• Query state DB for reads
• Build RWSet
5) Ordering service creates batch (block) of transactions
Ordering service
Committing peer (all peers)
7) Validate each Transaction and commit block
• validate endorsement policy (VSCC)
• Validate Readset versions in state DB (MVCC)
• Commit block to blockchain
• Commit valid trans to state DB
Blockchain (File system)
State Databasr
Blockchain (File system)
State Databasr

C. Data on a need-to-know basis

Due to competitiveness, protection laws, and regulation on confidentiality of personal data, business need to maintain a certain level of privacy over data elements, which can be achieved through Channels supported in Hyperledger Fabric. Hyperledger Fabric, through channels allow data to go to only the parties that need to know.

Channels help provide a data-partitioning capability where only those that need to know the data will see the number of transactions and the data itself.

D. Rich queries over and immutable distributed ledger

The ledger is the sequenced record of state transitions for the blockchain application. Each transaction results in a set of asset key-value pairs that are committed to the ledger as creates, updates, or deletes. The immutable source of truth for v1.0 is appended into the file system of the peer, which also has LevelDB embedded.

LevelDB has, by default, a key value database and supports keyed queries, composite key queries, and key range queries. With optional support of a document database such as CouchDB, the content is JSON and fully queryable, where the data model is compatible with existing key/value programming model. As a result, the application changes are not required when modeling chaincode data as JSON when utilizing CouchDB.

The JSON format helps minimize the work required to produce simple reports and perform audit functions.

E. Modular architecture supporting plug-in components

The modularity of Hyperledger Fabric architecture enables network designers to plug in their preferred implementations for components, which is an advantage. One of the most requested areas for modularity is "bring your own identity" i.e. reuse of existing identity.

Other components of the architecture that can be easily plugged in include consensus or encryption, where some countries have their own encryption standards.

F. Protection of digital Keys and sensitive data

HSM (Hardware Security Module) support is vital for safeguarding and managing digital keys for strong authentication. Hyperledger Fabric provides modified and unmodified PKCS11 for key generation, which supports cases like identity management that need more protection. For scenarios dealing with identity management, HSM increases the protection of keys and sensitive data.

Conclusion

The need for a more secure and less time consuming KYC process is required. It will help the institution which are involved in KYC based regulatory process to reduced cost, reduced process overhead etc. The blockchain based Belrium-KYC system will have significant impact on market and it will certainly influence the KYC based process.

The infusion of Blockchain technologies and KYC process will revolutionize the industry. The technology behind the Belrium is powerful and offers limitless possibilities. At the same time, it is of the utmost importance to simplify the KYC process and to develop solution that have what it needs to excites the target audience. Therefore, An institution centric blockchain based KYC system which take care of all the overhead of institution who are involved in KYC based process is very much require .

In order to avoid the double minting of a block or the double signing of a block, a miner may include in his block a denunciation. This denunciation takes the form of two signatures. Each minting signature or block signature signs the height of the block, making the proof of malfeasance quite concise.

While we could allow anyone to denounce malfeasance, there is really no point to allow anyone else beyond the block miner. Indeed, a miner can simply copy any proof of malfeasance and pass it off as its own discovery.

Belrium