BitShares 2.0: General Documentation

Fabian Schuh
BitShares Europe, BitShares.eu
BitShares Europe, BitShares.eu
BitShares Europe, BitShares.eu
Erlangen, Germany
fabian@bitshares.eu

Daniel Larimer
Cryptonomex, Cryptonomex.com
Blacksburg (VA), USA
dan@cryptonomex.com*

***Abstract*—BitShares 2.0 is an industrial-grade decentralized platform built for high-performance financial smart contracts. The decentralized exchange that allows for trading of arbitrary pairs without counterparty risk facilitates only one out of many available features. Market-pegged assets, such as the bitUSD, are crypto tokens that come with all the advantages of traditional cryptocurrencies like bitcoin but trade for at least the value of their underlying asset, e.g. $1. Furthermore, BitShares represents the first decentralized autonomous company that lets its shareholders decide on its future direction and products. This paper gives a brief overview over the whole BitShares platform, recapitulates known blockchain technologies and redefines *state-of-the-art*.**

1 Introduction

BitShares is a technology supported by next generation entrepreneurs, investors, and developers with a common interest in finding free market solutions by leveraging the power of globally decentralized consensus and decision making. Consensus technology has the power to do for economics what the internet did for information. It can harness the combined power of all humanity to coordinate the discovery and aggregation of real-time knowledge, previously unobtainable. This knowledge can be used to more effectively coordinate the allocation of resources toward their most productive and valuable use.

Bitcoin is the first fully autonomous system to utilize distributed consensus technology to create a more efficient and reliable global payment network. The core innovation of Bitcoin is the Blockchain, a cryptographically secured public ledger of all accounts on the Bitcoin network that facilitates the transfer of value from one individual directly to another. For the first time in history, financial transactions over the internet no longer require a middle man to act as a trustworthy, confidential fiduciary.

BitShares looks to extend the innovation of the blockchain to more industries that rely upon the internet to provide their services. Whether its banking, stock exchanges [1], lotteries [2], voting [3], music [4], auctions or many others, a digital public ledger allows for the creation of *distributed autonomous companies* (or DACs) that provide better quality services at a fraction of the cost incurred by their more traditional, centralized counterparts. The advent of DACs ushers in a new paradigm in organizational structure in which companies can run without any human management and under the control of an incorruptible set of business rules. These rules are encoded in publicly auditable open source software distributed across the computers of the companies' shareholders, who effortlessly secure the company from arbitrary control.

BitShares does for business what bitcoin did for money by utilizing distributed consensus technology to create companies that are inherently global, transparent, trustworthy, efficient and most importantly profitable. Why and how BitShares achieves a decentralized but profitable business is described in more detail in a distinct paper [5].

BitShares has went through many changes and has done its best to stay on top of blockchain technology. Towards the end of 2014 some of the DACs were merged and the X was dropped from "BitShares X" to become simply BitShares (BTS).

The next step in the evolution of BitShares was named *Bitshares 2.0*, and incorporates all of the feedback and lessons learned from the BitShares stakeholders, partners, developers, marketers, and other community leaders throughout a full year of research and development.

With the former BitShares 1.0, the core development team has closely controlled the development and direction of BitShares. With BitShares reaching maturity at version 2.0, the team is ready to remove the training wheels, and let the direction of all future development be decided completely by stakeholder vote.

By utilizing a new worker voting system that will be included in BitShares 2.0, the development will continue in whatever direction is approved by its stakeholders. With this new structure, BitShares will be more robust, and sustainable while being agile, flexible and adaptive to overcome unforeseen hurdles of the future.

This paper is intended as an introduction to BitShares 2.0 and presents the basic concepts of the peer-to-peer nature, the distributed public ledger in form of a blockchain, and give a brief overview of the decentral consensus mechanism applied to reach blockchain state consensus. We further discuss the basic blockchain tokens (BTS), its distribution and usage in BitShares. We also describe the wallet and operations with the network as well as outline the functionalities of BitShares accounts.

*This work was supported by Cryptonomex and honorable members of the bitsharestalk.org community.

2 Architecture of BitShares

Before describing how BitShares can be used to secure financial freedom, we first discuss the technical specifications briefly. Among these is the public ledger (also referred to as *the blockchain*), the peer-to-peer network, the distributed consensus finding mechanism and the system parameters available to BitShares.

2.1 Public Ledger

As in other crypto-currencies, the public ledger of BitShares is built and stored in a linked series of blocks, known as a blockchain.

The ledger provides a permanent record of transactions that have taken place, and also establishes an order in which transactions have occurred. Hence, every *content* of the blockchain can be assigned an permanent and unique identifier in form of a scalar number.

Every full node in the BitShares network stores a full copy of this blockchain and can verify its validity and the evaluate new blocks.

Every block contains

- a reference to the previous block,
- a timestamp,
- a hash of a secret,
- the secret of the previous hash,
- a set of transactions, and
- a signature by the block producing authority

As will be discussed in section 2.3, the consensus mechanism allows for synchronous block production with constant block confirmation times, e.g., one block every $5\,\mathrm{s}$.

Since the blocks mainly embrace customer transactions but has to perform time intensive tasks, or execute rare events from time to time, some actions such as reenumeration of blockchain-based votes and rare events such as newly registered block producers (witnesses) are carried out more rarely but still on a frequent so called *maintenance interval*.

The following parameters are associated with the blockchain operations and are subject to shareholder consensus:

`block-interval`
: time in seconds between blocks

`maintenance-interval`
: time in seconds for a maintenance interval to evaluate more time-consuming tasks

`maximum-transaction-size`
: maximum size of a transaction in bytes

`maximum-block-size`
: maximum size of a block in bytes

`maximum-witness-count`
: number of witnesses allowed to actively produce blocks

`maximum-committee-count`
: maximum amount of committee members to take into account for parameter definition.

2.2 Low Latency Peer-to-Peer Network

The peer-to-peer network distributes the full blockchain database across the world. It consists of public and private nodes as well as seed nodes that are used for initial connection to the peer-to-peer network. Anybody may connect to any known node and download the current global and unique state (i.e. the blockchain).

Once a node is in sync with the peer-to-peer network it received and applies newly created blocks, and assists new network nodes by further distribution of the blockchain. Additionally, new blocks are broadcast to all connected nodes.

Furthermore, network nodes receive transaction from participants and forward them to the rest of the network until they reach the witness that is in charge of constructing the next block. Hence, new transaction broadcasts do not necessarily need to reach all nodes.

Building a *low-latency* network requires P2P nodes that have low-latency connections and a protocol designed to minimize latency. For the purpose of this document we will assume that two nodes are located on opposite sides of the globe with a ping round-trip time of $250\,\mathrm{ms}$.

In the Bitcoin network architecture, transactions and blocks were broadcast in a following manner: inventory messages notify peers of transactions and blocks, then peers fetch the transaction or block from one peer. After validating the item a node will broadcast an inventory message to its peers.

Under this model it will take approximately $0.75\,\mathrm{s}$ for a peer to communicate a transaction or block to another peer even if their size was 0 and there was no processing overhead. This level of performance is unacceptable for a network attempting to produce one block *every second*.

This prior protocol also sent every transaction twice: initial broadcast, and again as part of a block.

To minimize latency each node needs to immediately broadcast the data it receives to its peers after validating it. Given the average transaction size is less than $100\,\mathrm{bytes}$, it is almost as efficient to send the transaction as it is to send the notice (assuming a $20\,\mathrm{byte}$ transaction id).

Hence, each node implements the following optimized protocol:

```
function ONRECEIVETRANSACTION(from_peer, transaction)
 if ISKNOWN(transaction.id()) then
  return
 end ifMARKKNOWN(transaction.id())
 if ! VALIDATE(transaction) then
  return
 end if
 for peer : peers do
  if peer != from_peer then
   SEND(peer, transaction )
  end if
 end for
end function
function ONRECEIVEBLOCK( from_peer, block_summary )
 if ISKNOWN( block_summary ) then
  return
 end if
 full_block = RECONSTRUCTFULLBLCOK( from_peer,
block_summary )
 if !full_block then
```

```
    DISCONNECT(from_peer)
  end ifMARKKNOWN( block_summary )
  if !PUSHBLOCK( full_block ) then
    DISCONNECT(from_peer)
  end if
  for peer : peers do
    if peer != from_peer then
      SEND(peer, block_summary )
    end if
  end for
end function
function ONCONNECT( new_peer, new_peer_head_block_num )
  if peers.size ≥ max_peers then
    SEND( new_peer, peers )
    DISCONNECT( new_peer )
    return
  end if
  while new_peer_head_block_num < our_head_block_num do
    SENDFULLBLOCK( new_peer, ++new_peer_head_block_num )
  end while
  new_peer.synced = true
  for peer : peers do
    SEND( peer, new_peer )
  end for
end function
function ONRECEIVEPEERS( from_peer, peers )
  ADDTOPOTENTIALPEERS( peers )
end function
function ONUPDATECONNECTIONSTIMER
  if peers.size < desired_peers then
    CONNECT( random_potential_peer )
  end if
end function
function ONFULLBLOCK( from_peer, full_block )
  if !PUSHBLOCK( full_block ) then
    DISCONNECT(from_peer)
  end if
end function
function ONSTARTUP
  INIT_POTENTIAL_PEERS(config)
  START(onUpdateConnectionsTimer)
end function
```

2.3 Distributed Consensus Mechanism

Consensus is the mechanism by which a subset of people decide upon unitary rational action. The process of consensus decision-making allows for all participants to consent upon a resolution of action even if not the favored course of action for each individual participant. Bitcoin was the first system to integrate a fully decentralized consensus method with the modern technology of the internet and peer-to-peer networks in order to more efficiently facilitate the transfer of value through electronic communication. The proof-of-work structure that secures and maintains the Bitcoin network is one manner of organizing individuals who do not necessarily trust one another to act in the best interest of all participants of the network.

It is of importance to distinguish a democratic voting process in which every citizen of a community has one and only one vote from a distributed consensus mechanism in crypto-currencies hand over voting power either in relation to hashing power (e.g. proof-of-work) or on a per stake basis (e.g. proof-of-stake). In both cases, those that invest in the required infrastructure to increase their voting percentage (i.e. by buying mining hardware or stake) act as shareholder in a distributed community.

The BitShares community employs *Delegated Proof-of-Stake* (DPOS) in order to find efficient solutions to distributed consensus decision making. DPOS attempts to solve the problems of both Bitcoin's traditional proof-of-work system, and the proof-of-stake system of Peercoin and NXT by implementing a layer of technological democracy to offset the negative effects of centralization. For historical reasons, the technology is still called *delegated* proof-of-stake even though what have been delegates in BitShares 1.0 are now so called *witnesses*.

In DPOS set of N witnesses (formerly known as *delegates*) sign the blocks and are voted on by those using the network with every transaction that gets made. By using a decentralized voting process, DPOS is by design more democratic than comparable systems. Rather than eliminating the need for trust all together, DPOS has safeguards in place the ensure that those trusted with signing blocks on behalf of the network are doing so correctly and without bias. A more detailed description about the distributed consensus mechanism as well as a discussion how blockchain forking is prevented during attacks is given in a separate paper [6].

Additionally, each block signed must have a verification that the block before it was signed by a trusted node. DPOS eliminates the need to wait until a certain number of untrusted nodes have verified a transaction before it can be confirmed.

This reduced need for confirmation produces an increase in speed of transaction times. By intentionally placing trust with the most trustworthy of potential block signers, as decided by the network, no artificial encumbrance need be imposed to slow down the block signing process. DPOS allows for many more transactions to be included in a block than either proof of work or proof of stake systems.

In a delegated proof-of-stake system, centralization still occurs, but it is controlled. Unlike other methods of securing crypto-currency networks, every client in a DPOS system has the ability to decide who is trusted rather than trust concentrating in the hands of those with the most resources. DPOS allows the network to reap some of the major advantages of centralization, while still maintaining some calculated measure of decentralization. Furthermore, once a witness has reached approval by shareholders, surpasses the threshold of the most N active witnesses, and, hence, is elected to actively participate in the block production procedure, its power is *equivalent* to all other active witnesses. This system is enforced by a fair election process where anyone could potentially become a delegated representative (witness) of the majority of users.

Please note that DPOS has a recommended $1-2$ block confirmation versus bitcoin's 6 block recommendation. DPOS is much more resistant against forks for the following reasons:

- When a fork is produced it is very likely that all witnesses have seen and processed your transaction and thus no alternative transactions can be broadcast and the next witness is almost certain to include your transaction. All witnesses are much more trusted than miners.

- The probability of a fork after a block has been produced is very low (¡ 0.01%) where as Bitcoin has 25 orphans in the last 22 days (about 1 per day in Dec 3,2014) which translates into 0.7% of blocks are orphaned.
- On normal operations, DPOS achieves a 100% witness participation rate and when we are less than that it is more often because a witness went offline and didn't produce a block than because they produced a fork.
- In BitShares 1.0 forks have almost always been resolved within 30 seconds.

Assuming a 10 second block interval, Bitshares is mathematically over 70x less likely to orphan after 1 block than Bitcoin after 1 block (10 minutes). After 3 blocks (30 seconds) any random orphan will have been resolved and the probability of alternative chains is much lower than the 0.000001% of Bitcoin. By the time Bitcoin gets to .7% orphan probability, BitShares has 60 blocks which would have a probability of being orphaned of less than 10^{-120}.

3 Crypto Token

In the BitShares network the base token is called *a BitShare* and carries the abbreviation `BTS`. It is dividable into 10^5 sub-units which are denoted as follows (proposal) :

$$\begin{aligned} 1\,\text{mises} &= 0.000\,01\,\text{BTS} \\ 1\,\text{xennon} &= 0.0001\,\text{BTS} \\ 1\,\text{oxyd} &= 0.001\,\text{BTS} \\ 1\,\text{graphn} &= 0.01\,\text{BTS} \\ 1\,\text{epox} &= 0.1\,\text{BTS} \end{aligned}$$

In general, all properties of Bitcoin also apply to BTS, namely, they have value, can be transfered on the blockchain and are secured by an Elliptic Curve Digital Signature Algorithm (ECDSA) on the curve `secp256k1`.

In contrast to most crypto-currencies, BitShares does not claim to be a currency but rather an *equity* in a decentral autonomous company (DAC). As a result, the market valuation of BitShares is free floating and may be as volatile as any other equity (e.g. of traditional companies).

Nonetheless, BTS tokens can be used as *collateral* in financial smart contracts [7] such as market pegged assets and thus back every existing smartcoin such as the bitUSD.

The following subsection recapitulate the initial distribution and supply of BTS.

3.1 Distribution of BTS

BitShares has set an example of a *social agreement* by establishing its own *sharedropping* standards. The idea behind sharedropping is that any future chain will always benefit by choosing to align itself with the ones who worked hard at making the technology possible.

The base tokens of BitShares *2.0* will be distributed on a 1:1 basis fully honoring the BTS tokens in the BitShares *1.0* network. For the sake of completeness, the following paragraphs will describe the initial distribution of BTS tokens in the aforementioned BitShares 1.0 network from PTS and AGS.

3.1.1 Bitshares PTS

The original grandfather prototype, formerly called proto-shares (PTS), BitShares PTS was a simple minable crypto-currency (similar to Bitcoin) that was created to allow people to advertise their interest in receiving free token samples in future DACs. PTS functions as a high-tech *mailing list* for distributing free sample bitshares from many developers of decentral autonomous companies (DACs). The only people who tended to own PTS tokens were those who understand DACs, so DAC developers prefer to target them with free samples rather than *air dropping* their samples onto a much less interested general population.

The industry recommendation was that when a DAC is launched, at least 10% of the DAC's total tokens are given proportionally to holders of PTS. This was not a contract or a guarantee; it was a *social consensus* of those in the DAC community about what percentage of a new DAC's tokens should be distributed to those who have supported the BitShares industry by owning its PTS tokens.

The BitShares DAC honored this social consensus and even sharedropped 47% of its ever existing supply onto BitShares PTS holders.

3.1.2 Bitshares AGS

The original grandmother prototype formerly called angel-shares (AGS) in reference to the patron *angels* who once funded the performing arts. That's why AGS are *not liquid*. (No one can trade the proof that you were the once willing to donate to this cause.)

The donations have been recorded in the public blockchain of bitcoin which now acts as a *book of honorable donors*. The bitcoin address used as donation address was

`1ANGELwQwWxMmbdaSWhWLqBEtPTkWb8uDc`

Note, that the donation period for AGS lasting 200 days has ended already and that donations to this address never resulted nor will result in any obligations whatsoever.

Since the social consensus includes AGS, the industry recommendation again is to give at least 10% to holders of AGS. Similar to BitShares PTS, the 47% of the total BTS supply have been sharedropped onto members of the AGS mailing-list proportionally.

3.2 Bitshares Genesis Distribution

We see that the seed allocation (initial distribution) of BitShares, which took place over a 1 year period, from November 2013 to November 2014, was achieved by sharedropping 47% to BitShares PTS and another 47% to BitShares AGS. This way, the full, fairness was defined by equal opportunity and in the case of BTS we have distributed *fairly* by CPU mining of PTS while, alternatively, everyone had an additional equal opportunity by contribute to AGS.

Having attracted two different groups of investors with a mined crypto token via PTS and a donation based book of donors via AGS, everyone had a chance to participate and be rewarded with stake in the genesis block of BitShares 1.0. This genesis block solely consisted of AGS and PTS holders on a 50%/50%

ratio such that the BTS tokens initially issued by this genesis can be considered *well distributed*.

The other 6% are set aside to secure the future of BitShares and funds its development and operational costs. In practice, they are put into the so called *reserves pool* that no one has control over except the BitShares protocol. In contrast to many other crypto-currencies, every shareholder has a say as to how these funds are spend (see section 5.2).

4 Business Units

Let us discuss the organization structure of the BitShares network when interpreted as a company. Some of these entities are associated with a cost for the business and need to be accounted for in profit calculations.

4.1 BitShares Witnesses

In BitShares, the witnesses' job is to collect transactions, bundle them into a block, sign the block and broadcast it to the network. They essentially are the block producers for the underlying consensus mechanism (see section 2.3).

For each successfully constructed block, a witness is payed in shares that are taken from the limited reserves pool at a rate that is defined by the shareholders by means of approval voting.

4.2 BitShares Committee

Since Bitcoin struggled to reach a consensus about the size of their blocks, the people in the cryptocurrency space realized that the governance of a DAC should not be ignored. Hence, BitShares offers a tools to reach on-chain consensus about business management decisions.

The BitShares blockchain has a set of parameters available that are subject of shareholder approval. Shareholders can define their preferred set of parameters and thereby create a so called *committee member* or alternatively vote for an existing committee member. The BitShares committee consists of C *active* committee members.

For each business parameter the protocol will calculate the difference between up- and down-votes ($v_{\text{pro}} - v_{\text{con}}$) for each active committee member and then take the median of the top C active members:

// Derive active C committee members
for i : active committee members **do**
  member weight: $w[i] \leftarrow v_{\text{pro}} - v_{\text{con}}$
end for
members $\leftarrow$ SORT(w)
active $\leftarrow$ members$[0 \rightarrow C]$
// For each Parameter: derive median of active members
for parameter : parameters **do**
  $p \leftarrow$ GETPARAMETERS(active, parameter)
  $x = \text{sort}(p[i])$

$$\tilde{p} = \begin{cases} x[\frac{C+1}{2}] & C \text{ odd} \\ \frac{1}{2}\left(x[\frac{C}{2}] + x[\frac{C}{2}+1]\right) & C \text{ even.} \end{cases}$$

  parameter $\leftarrow \tilde{p}$
end for

Since, C is a parameter as any other, the shareholders decide for the size of the committee.

The BitShares ecosystem has a set of parameters available that are subject of shareholder approval. Initially, BitShares has the following blockchain parameters:

fee structure:
fess that have to be paid by costumers for individual transactions

block interval:
i.e. block interval, max size of block/transaction

expiration parameters:
i.e. maximum expiration interval

witness parameters:
i.e. maximum amount of witnesses (block producers)

committee parameters:
i.e. maximum amount of committee members

witness pay:
payment for each witnesses per signed block

worker budget:
available budget available for budget items (e.g. development)

Please note that the given set of parameters serves as an example and that the network's parameters are subject to change over time.

Additionally to defining the parameters any active witness can propose a protocol or business upgrade (i.e. hard fork) which can be voted on (or against) by shareholders. When the total votes for the hard fork are greater than the median witness weight w then the hard fork takes effect.

4.3 BitShares Budget Items

Thanks to the funds stored in the reserve pool, BitShares can offer to not only pay for its own development and protocol improvement but also support and encourage growth of an ecosystem.

In order to be get paid by BitShares, a proposal containing

- date of work begin
- date of work end
- daily pay (denoted in BTS)
- name
- internet address

has to be publish on the blockchain and approved by shareholders.

A blockchain parameter (defined by shareholders through the committee) defines the daily available budget. No more than that can be paid at any time to all so called *workers* combined.

The daily budget is distributed as illustrated in fig. 1: (1) The available budget is taken out of reserves pool. (2) The budget items are sorted according to their approval rate ($v_{\text{pro}} - v_{\text{con}}$) in a descending order. (3) Starting at the worker with the highest approval rate, the requested daily pay is payed until the daily

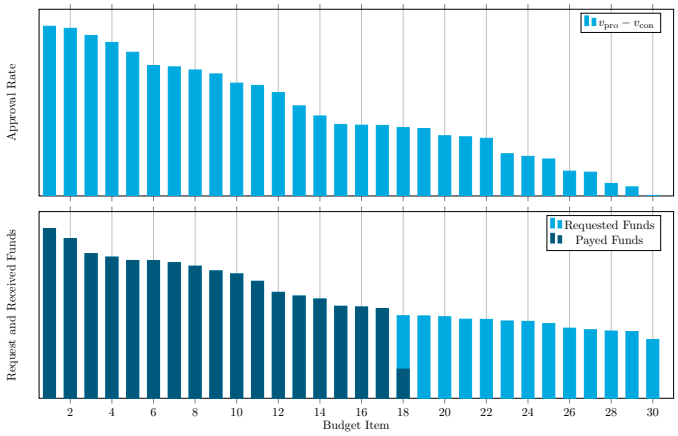

Figure 1: Illustration of budget item payments.

budget is depleted. (4) The worker with the least approval rate that was paid may receive less than the requested pay

Hence, in order to be successfully funded by the BitShares ecosystem, the shareholder approval for your budget item needs to be highly ranked.

Since the payments for workers from the non-liquid reserve pool result in an increased supply of BTS, these payments are vesting over a period of time defined by shareholders.

5 BitShares: A profitable DAC

BitShares is a decentralized autonomous company, and as such offers products to to earn their shareholders a profit. As we have seen in the previous section, it also offers a way to pay for expense, such as development and administration but earns a profit by burnning (i.e. reducing supply). Of course, the company can only be profitable if the income exceeds the expenses. Thus, we will now discuss both in detail.

5.1 The products

The products offered by the BitShares DAC are:

* Flexible Identity Management * Fast International Payments * Stable SmartCoins * Customizable Assets * Instant Trading

Costumers: * Global Accessibility * Anonymity * Security * Trustless: Counterparty-Risk Free * Flexible Control * Low Delay * Decentralization

5.2 Revenue and Expenses

Revenue streams are essentially caused by fees that have to be payed when using the DACs products, such as market pegged assets, user-issued assets, or the decentralized exchange [7]. These fees are variable, can be changed by shareholder approval and include (a) transfers, (b) order operations, (c) account operations, (d) asset operations, (e) witness creations (f) proposal operations (g) withdraw permission operations (h) committee member operations, (i) worker creation, and more.

In contrast to bitcoin, where newly created coins in each block are distributed solely among countless miners that immensely overpay for the network security [8], the BitShares ecosystem achieves a better security at lower costs by means of an adjustable number of approved and trusted witnesses in DPOS. Additionally, the BitShares ecosystem has the capability to pay for its own development through budget items. Both, the payment for witnesses, as well as the budget items are required to be approved by the shareholders.

As an example, an entrepreneur may approach the shareholders and offer to launch a business in the BitShares space that would greatly benefit the ecosystem. If he succeeds and convinces the shareholders to vote for and not against his plan, he could get an initial funding by the DAC.

Another use-case would be the improvement of the blockchain's protocol. A developer could propose a change or extension of the existing software implementation and be payed by the DAC to do so (after shareholder approval). Hence, as long as the average shareholders acts rational, the BitShares blockchain can be seen as a self-funded but profitable business

5.3 Fee Schedule and Cash Flow

5.4 Growth Considerations

6 Technical Specifications

We will now describe the use of objects to implement different entities on the blockchain.

6.1 Operations

Similar to most crypto-currencies, there is a set of predefined operations that can be performed on the blockchain. In contrast to Bitcoin, which uses a technique called *script* to describe operations that shall be performed in a programmatic way using *OP* codes, the BitShares network has a predefined (but extensible) set of operations that a user may perform.

All operations end up on the blockchain eventually. Once they are validated and confirmed by a witness by being included into a block, they are *executed* and update the state of the blockchain accordingly.

The release version of BitShares 2.0 comes with (a) transfer ops, (b) trading order ops, (c) account ops, (d) asset ops, (e) witness ops, (f) committee ops, (g) worker ops, and (h) vesting ops, However, since BitShares allows for shareholder approved, live protocol upgrades, the set of operations can be extended and modified.

On the blockchain level, each operation is assigned an individual *id* with a custom set of parameters for performance and latency reasons.

6.2 Transactions

Having defined *operations*, we can now put these into a *list of operations* and construct a *transaction*.

```
{
    "ref_block_num": ...,
    "ref_block_prefix": ...,
    "expiration": [...],
```

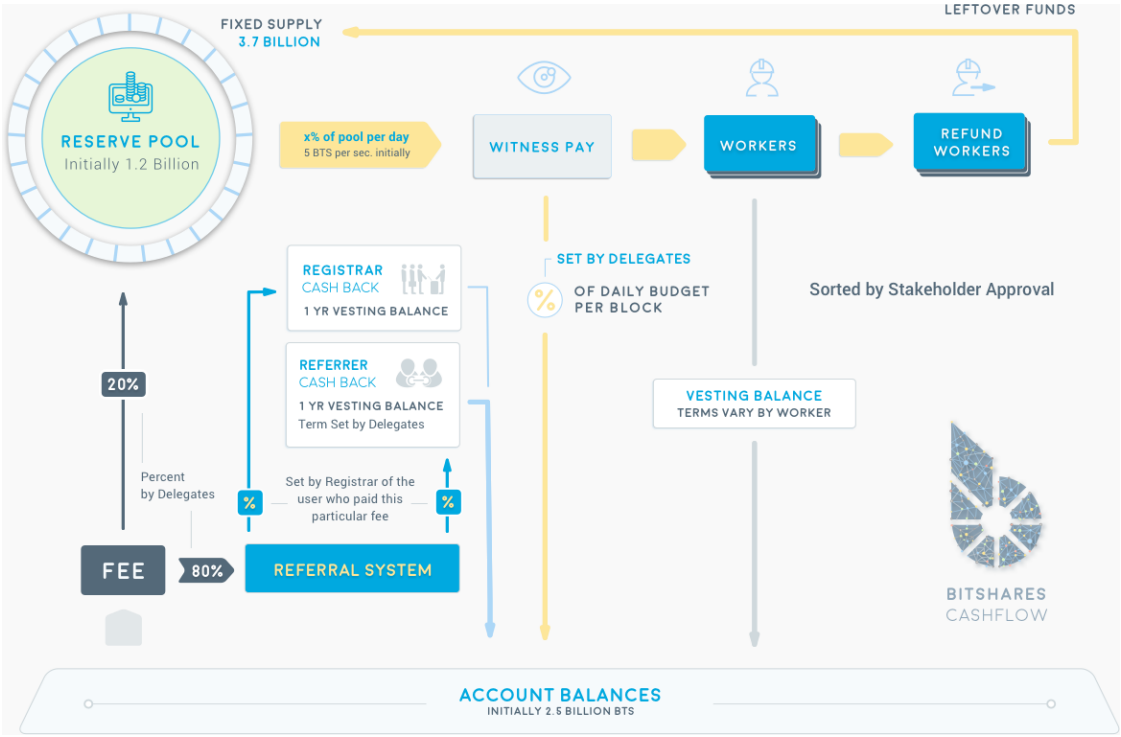

Figure 2: Cash flow of BitShares 2.0

```
    "extensions": [...],
    "operations": [...],
    "signatures": [...],
}
```

In addition to its operations, a transaction also consists of (a) an expiration date, (b) a reference block number, (c) a reference block prefix, (d) a set of extensions, and (e) a set of signatures to authorize each operation.

Each node (including witnesses) verifies that all requires signatures to perform the given operations are present and valid prior to propagating the transactions to the rest of the network and hence to the witness node constructing the next block. If the transaction is included into a block it is considered *finally valid* or *executed*.

6.3 Objects

Speaking abstractly, the BitShares blockchain does not distinguish between assets, account, or operations. These terms do only exist outside the blockchain. Instead, the blockchain deals with *contextual objects* that are associated with a given set of features, permissions, etc.

Hence, on the BitShares blockchains there are no addresses similar to Bitcoin, but objects identified by a unique *id*, a *type* and a *space* in the form:

```
space.type.id
```

The reserved *space* are

```
enum reserved_spaces {
   relative_protocol_ids = 0,
   protocol_ids          = 1,
   implementation_ids    = 2
};
```

As an example, we have the following objects:

1.2.15 15th blockchain account

1.6.105 105th blockchain witness

1.14.7 7th blockchain worker proposal

2.1.0 wallet dynamic global properties

2.3.8 8th asset

A programmatic description of all fields can be found in the sources.

6.4 Named Account

BitShares makes use of read-able account names that have to be registered together with a public key in the blockchain prior to its usage. On registration, a new account is associated with an individual and unique identifier that will be used internally to identify an account. In this case, the blockchain acts as a name–to–public-key resolver similar to the traditional domain name service (DNS). These named accounts enable users to easily remember and communicate their account information.

In BitShares, an account name can be transfered by updating the public key that is associated with it.

Furthermore, BitShares is the first smart contract platform with built-in support for *recurring payments* and subscription payments. This feature allows users to authorize third parties to

make withdrawals from their accounts within certain limits. This is a convenient way to *set it and forget it* for monthly bills and subscriptions.

BitShares also designs permissions around people, rather than around cryptography, making it easy to use. 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. Hence, BitShares does technically not have multi-*signature* accounts, but has multi-*account* permissions.

In the following we introduce the above mentioned features that come with named account in the BitShares network in more detail.

6.4.1 Transferability

Every BitShares account is assigned a globally unique name that can be selected by its creator. There are many potential uses for account names beyond simply being an alias to a set of *dynamic account permissions*. They can be used as user logins or mapped to domain names. These names are transferable, which means that they are valuable in their own right.

The BitShares blockchain defines a simple algorithm to determine the fee it charges to reserve a new account name. Names that contain a number, are longer than 8 characters, or contain no vowels are essentially free. Otherwise the name is priced according to its length. Delegates can propose a different fee for each length which gives BitShares the power to adapt to market demands.

As mentioned already, accounts can be transferred by updating the permissions used to control the account. However, the semantics of transferring an account are slightly different in the context of a web-of-trust. Users need a means to update their keys for security purposes while maintaining their standing in the web-of-trust. In other words, users must be given a way to explicitly transfer an account name to a new user while breaking any liability for how the account is used in the future.

When users transfer an account name to another user, they use a special transaction that clears all of the links in the web-of-trust. Both the buyer and the seller are protected by this fact, because simply updating the key that controls a named account does not signify a legal change in ownership.

6.4.2 Dynamic Permissions

The ability to require multiple digital signatures for sensitive operations on the blockchain is integral to the security of the platform. While a single secret key may be compromised, multiple keys distributed over multiple locations add redundant protections, which result in a far more secure experience.

Competing blockchain systems suffer from the following shortcomings:

- The M-of-N model cannot sufficiently reflect the management hierarchies of many real-life organizations.
- Equal weighting of M keys is not sufficient to express asymmetric ownership over an account.
- Coordination and signing must be done completely out-of-band.
- Keys cannot be changed without coordinating with all other parties.
- Signatures cannot be retracted while waiting on other parties.

Multi-signature technology is all about permission management, and permissions should be defined in terms of people or organizations rather than keys. Consider an example company that is run by 3 individuals: Alice, Bob, and Carol. Alice and Bob each own 40% of the company and Carol owns 20%. This company requires 2 of the 3 principles to approve all expenses. You could define the company in terms of keys assigned to Alice, Bob, and Carol, but what if Alice wants to protect her own identity with a multi-signature check? Alice opts to use a service provider that performs 2-factor authentication on every action Alice makes. This protects both Alice and the company and the company does not need to change its permission structure to accommodate Alice's choice of 2-factor authentication provider.

In BitShares, we introduce a new approach to permissions based upon accounts which are assigned globally unique IDs.

Under this system, it is possible to define an account that has no keys itself, but instead depends solely upon the approval of other accounts. Those other accounts can, in turn, depend upon the approval of other accounts. This process forms a *hierarchy* of accounts that must grant permission. Each account can change its own permissions independently of any accounts above it in the hierarchy, which is what makes the permissions *dynamic*.

Each account defines its permissions as a set of keys and/or other account IDs that are each assigned weights by the account holder. If the combined weight of keys and/or accounts exceeds a threshold defined by the account, then permission is granted.

The second solution is to include the partially signed transaction in the consensus state and allow accounts to publish transactions that add or remove their approval of the transaction. This simplifies the signing coordination problem, enables people to change their mind before the threshold is reached, and applies the transaction immediately upon receipt of the final approval.

The process for executing a transaction that requires multi-signature authority is as follows:

1. Someone proposes a transaction and approves it with their account.

2. Other account holders broadcast transactions, adding their "Yes" or "No" to the proposal.

3. When the proposed transaction has the approval of all accounts, it is confirmed.

Every account is assigned *two* authorities: *owner* and *active*.

- An authority is a set of keys and/or accounts, each of which is assigned a weight.
- Each authority has a weight threshold that must be crossed before an action requiring that authority may be performed.

- The owner authority is designed for cold-storage, and its primary role is to update the active authority or to change the owner authority.
- The active authority is meant to be a hot key and can perform any action except changing the owner authority.
- The motivating use case is a 2-factor authentication provider as a co-signer on the active authority, but not on the owner authority.

With this approach, a user can remain confident that their account will always be in their control, and yet that control can be kept in cold storage where no one can hack it. This means that a company account can require the approval of its board of directors and each board member may in turn require 2 factor authentication.

Anyone can rotate keys frequently without having to disturb the permissions on the accounts of its users.

One of the challenges that has made multi-signature approaches difficult to use in the past is that the act of gathering the required signatures was entirely manual, or required specialized infrastructure. Once a transaction is signed, there is no ability to retract your signature, so the last party to sign gains a slight advantage over the other parties. With deeper hierarchies, gathering signatures becomes even more complex.

To simplify this process, a blockchain should manage the signature gathering process by tracking the state of partially approved proposed transactions. Under this process, each account can add (or remove) their permission to a transaction atomically, without having to rely upon an outside system to circulate the transaction. This becomes especially critical for hierarchies that are arbitrarily deep.

In order to keep things computationally bounded, an individual transaction will only traverse down two layers in a hierarchy. If more than two layers of hierarchy are present, then an account will have to propose (create one transaction) to approve a proposal (the other transaction). When the first proposal (transaction) is approved, permission is then added to the second proposal (transaction).

Under this approach, each individual pays a single transaction fee each time they approve an action, and every action involves at most 1 signature verification by the network. This process allows arbitrarily deep hierarchies to be formed without exposing the permission system to vulnerability of unbounded computation.

In theory, accounts can form a hierarchy that is arbitrarily deep, and evaluating that hierarchy can take an arbitrary amount of time. In practice, it is unlikely that a single transaction will have signatures more than 2 levels deep, which keeps them computation bounded. Anything that requires more than 2 levels is likely to involve many people, and would not be signed all at once. Instead, it would use the built-in proposed transaction infrastructure, which tracks partially approved transactions.

- With this approach, a board member can propose that his company approve a transaction.
- This can be extended logically to propose, and account propose, to approve a transaction.
- This process would collect transaction fees as all of the layers in the hierarchy gradually add their permissions, and at no time requires an unbounded calculation.

It is possible for two accounts to require each other to approve a transaction.

Imagine account X is created that requires A and Y to approve. Imagine account Y is created that requires B and X to approve. The graph looks like this:

$$A \rightarrow X \leftrightarrow Y \tag{1}$$

$$B \rightarrow Y \leftrightarrow X \tag{2}$$

A proposes that X spend 1 BTS and waits for approval from Y. B proposes that Y approve the proposal from A and waits for approval from X.

There is no way to resolve this problem with a single approval from any party due to the following reasons:

1. Neither account can act without the other and thus nothing can be accomplished.

2. Cycles don't have to be direct as in this case, they can involve arbitrarily long sequences and thus be non-obvious.

3. If users create an approval cycle in the active authority then the owner authority can be used to break the cycle; however, if they construct a cycle in the owner authority and the active authority then the accounts involved in the cycle would be locked out.

4. In practice client software can detect cycles and prevent them from being formed.

Dynamic hierarchical threshold multi-signature permissions provides people and organizations with a more natural way to express ownership and control policies. This approach makes the system easier to use, and ultimately more secure, than existing solutions.

The Ripple wiki has a documented, but unimplemented, proposal for a similar Multisign feature [9] that was discovered independently.

6.4.3 Recurring & Scheduled Payments

Recurring Payments are implemented as a set of withdrawal permissions. Each account can grant any number of withdrawal permissions to other accounts. A withdrawal permission includes following properties:

1. Start Date

2. End Date

3. Withdrawal Limit per Period

4. Period Length (i.e. 1 month)

Any asset type can be used in the withdrawal limit.

After a user grants the withdrawal permissions, the authorized account is allowed to make one transfer per period of an amount up to the limit. If there is insufficient funds then the withdrawal will fail. Withdrawal permissions are designed to be a convenience for merchants and users, as they do not represent a commitment to pay.

It is up to each merchant to initiate each withdrawal. The BitShares platform does not automatically authorize the transfer of funds unless sufficient signing authority has been reached.

For security purposes, many banks place daily withdrawal limits on user accounts. In the event that an account is compromised, a thief is limited in the amount of damage that they can do. Withdrawal permissions enable users to protect their BitShares funds in the same manner. To do so, a user creates two accounts: savings and checking.

The savings account has keys kept offline where they are unlikely to be compromised. Before placing the keys in cold storage, the savings account authorizes the checking account to make a daily withdrawal of up to $1000, for example.

The checking account can then pull money out of savings at up to this limit, per day, and then use those funds as needed. This gives the user confidence that their losses would be limited if their account is compromised.

As stated above, the withdrawal permission system does not automatically make payments. However, BitShares has another feature which enables scheduled payments: proposed transactions. At any time, a user can propose a transaction to execute at a specific date and time in the future. If the transaction has sufficient authorization (i.e. is properly signed by authorities) at the specified time, then it will automatically be executed.

A merchant can use this feature, combined with withdrawal permissions, to implement automatic payments after a one-time setup fee. In practice, it may be cheaper for merchants to maintain their own scheduler to automate billing, since the blockchain charges a fee to propose a transaction separately from the transaction's own fees.

6.4.4 Costumer Privacy

BitShares 2.0 makes use of confidential transactions (by means of blind signatures [10]) in combination with stealth addresses to protect customers privacy while keeping scalability and performance.

In [10], a scheme was proposed that allows generating a blind signature compatible with the existing Bitcoin protocol. There, the client requests a set of parameters from a 3rd partiy (e.g. a *signing server*) and synthesizes a public key to use in a transaction. To redeem the funds, the client transforms the hash of the transaction (*blinds*), sends to the 3rd party to sign and then transforms the signature (*unblinds*) to arrive at a valid ECDSA signature. The signed transaction is published revealing the synthetic public key and the unblinded signature. Hence, the 3rd party cannot learn about its participation neither from the public key nor from the signature.

The novelty of the scheme is that unlike the original Chaum blind signature scheme, this approach does not allow anyone to prove that the signing party signed a particular message, but instead provides a much stronger privacy: the resulting signature, public key and the message are all completely unlinkable to the signing party.

For BitShares, it allows to have *someone* else sign something without them being able to prove they were the one who signed it which is useful for multi-signature/two-factor on confidential transactions.

In combination with stealth addresses, we can now derive a *one-time public key* that can be signed by *someone* knowing that this *someone* cannot use any knowledge of the one-time public key to link back to us.

7 Discussion

In general, BitShares has similarities and differences to most known crypto-currencies. As many others, BitShares is based on a blockchain that stores and propagates transactions, i.e. user operations. Since, with DPOS, computational resources are used solely for the purpose of transaction propagation and confirmation, rather than wasteful computational work, the block production interval has been reduced to a few seconds. Eventually, this improves the over-all profitability of the DAC.

Additionally, we make use of *named* accounts that can be registered on the blockchain. Users no longer need to send money to an alphanumeric string that can be copied incorrectly. Rather, funds can be sent as easily as sending an email, and in the same fashion. Name registration allows for the identification of who transactions are are originating with with no need to manually create a contact account for a given address. Transactions may contain a memo field that allow users to describe the nature of the transaction or broadcast secure messages about the price of the current transaction fee. Since BitShares 2.0 implements *confidential transaction*, there is no longer a need for *mixing* or *master nodes*. Transactions can be more private iin BitShares than in Bitcoin, for example, with no additional work needed from the user.

BitShares is a 100% proof-of-stake system. This means it is a lot more efficient (cost per security) than proof-of-work and therefore does not have to dilute stakeholders/coinholders (there is a 10% yearly dilution of Bitcoin-holders as per 2015 with Bitcoin and lowering this dilution would mean to lower the security). Hence, the cost of securing the BitShares network is merely a fraction of all transaction fees accumulated by the network.

The job of the block producers is simple: include as many valid transactions in your given block as possible and sign a single block. These Block producers compete for the most approval in order to be allowed to produce blocks. Shareholder votes are proportionate to the relative number of shares they own. The BitShares DAC is *completely* shareholder run. Now people can be hired by the blockchain. Where coins like Bitcoin dilute to pay for network security, BitShares takes these fees and directs them towards continual improvement of the network and community. This helps insure BitShares will stay competitive in its feature set. More details about the consensus scheme of BitShares can be found in a separated whitepaper [].

Recalling the initial distribution of BTS, it seem convincing to assume that most alternative distributions are way more unfair and some disproportionately favor their respective core developers. Since BitShares is a self-funded DAC, it can *pay* for its future development autonomously by dilution, if shareholders reach an on-blockchain consensus by approval voting.

8 Conclusion

The properties and features mentioned in this paper make clear that the BitShares DAC is well-prepared for its own features. It was shown that, due to on-blockchain voting, a decentralized development and funding can be achieved. The consensus mechanism DPOS reaches a trade-off between efficiency and required trust while keeping a better decentralization that mostly every other blockchain consensus scheme.

References

[1] Daniel Larimer, "Creating a Fiat/Bitcoin Exchange without Fiat Deposits." https://bitcointalk.org/index.php?topic=223747.0.
[2] "DACPlay," http://dacPLAY.org.
[3] Adam Ernest, "Follow My Vote," http://followmyvote.com.
[4] Cédric Cobban, "Follow My Vote," http://peertracks.com.
[5] "BitShares 2.0: Business Plan," *BitShares Whitepapers*, 2015.
[6] "BitShares 2.0: Distributed Consensus," *BitShares Whitepapers*, 2015.
[7] "BitShares 2.0: Financial Smart Contract Platform," *BitShares Whitepapers*, 2015.
[8] Daniel Larimer, "Overpaying for Security," http://letstalkbitcoin.com/is-bitcoin-overpaying-for-false-security/#.Ui-p9WTFT7s.
[9] Ripple Labs, "Multisig / Transaction Proposal," https://wiki.ripple.com/Multisign#Transaction_Proposal.
[10] Oleg Andreev, "Blind signatures for Bitcoin transactions (second draft)," http://oleganza.com/blind-ecdsa-draft-v2.pdf.