MIX Blockchain Documentation

Release 1.0

Jonathan Brown <jbrown@mix-blockchain.org>

Dec 30, 2020

Contents

MIX is an uncontrolled linked data ecosystem living on the blockchain. It is fully public. It cannot be censored and no one can be prevented from participating. It is fully programmable at every level. It is a protocol for interconnected distributed content apps that empower the individual.

The blockchain was activated on Thursday 27th April 2017 09:15:23 AM (UTC).

Useful links

- Homepage
- Blog
- Twitter
- Reddit
- Gitter
- YouTube
- BitChute
- Block explorer
- Network stats
- Issue tracker
- Logo

Table of contents

2.1 The Vision

Twitter, Reddit, Medium, Instagram, YouTube are all broken because the content is trapped in the platform. This leads to censorship and controlled filter bubbles, reducing freedom of expression.

The solution is that rather than publishing to a platform controlled by a company, content should be published using a neutral permissionless protocol using decentralized technologies such as Ethereum and IPFS.

By de-coupling publishing and consumption of content, developers are free to innovate and every user can choose the app or filter bubble they want. This freedom will lead to greater richness of online communities that are better able to attain the goal of empowering individuals through knowledge and understanding.

MIX is based on a clone of Ethereum, providing independence and a huge capacity increase over using the Ethereum main chain.

2.2 The Problem

In Tim Berners-Lee's original vision of the World Wide Web anyone could publish a document and anyone could read any document. In fact, the original WWW software was both a web browser and an editor.

Anyone could spin-up a server to publish and anyone with an Internet connection could connect to any server, i.e. it was decentralized.

Over the years this paradigm has gradually been eroded with publishing platforms such as Facebook, Twitter, Medium, Reddit, Instagram, YouTube, etc becoming popular. These are highly controlled platforms where the user can become disempowered without any right of recourse. Of course users can still publish on their own servers, but in many ways these platforms offer a lot of advantages over hosting your own website.

Centralized platforms are preventing communities from flourishing in a way that would otherwise be possible.

2.2.1 Advantages of current centralized solutions

Social graph

Most successful publishing platforms rely on some form of social organization - and this has been key to their success. On Facebook you can add your friends that you want in your network. On Twitter and Medium you can follow those whose opinions you want to hear. On YouTube you follow your favorite channels.

Voting

Liking, voting up/down, etc has become so vital to the content ecosystem. The open web has never really had this sort of functionality become popular.

Moderation

Reddit and Facebook groups have become very successful in part because of their moderation systems. The open web doesn't have a moderation system, although Google has been filling this role to an extent.

Machine readable data

Originally, web pages were whole documents and CSS was used to for styling. Of course this was completely inadequate. Developers started to put design into HTML and dynamically generating web pages on the backend. This makes it very difficult for a machine to parse the information. In recent years the Semantic Web has been used to re-add the actual machine readable information into web pages.

Content needs to machine readable first. Templating and styling are presentation layer and should be implemented on the front-end. Facebook and Google maps are much better than the open web at publishing information because they are machine-readable first.

2.2.2 Disadvantages of current centralized solutions

Hard censorship

Twitter sometimes bans people entirely.

- Ahed Tamimi
- Milo

On Medium content creators are sometimes ordered to change their content under threat of it being removed altogether.

Large sections of Reddit will sometimes disappear.

Shadow banning

Both Twitter and Reddit will block users in such a way that it will not be obvious to the user https://www.youtube.com/watch?v=64gTjdUrDFQ

India's subreddit https://medium.com/@krantikaari_r/how-indias-biggest-sub-reddit-is-being-silently-censored-16ac656624e6

The ability to publish should be an unaliable right - no should be able to take it away from you.

Platform operators can change anything

Because centralized platforms have operators, they have the power to change anything. For example the CEO of Reddit once edited comments that critized him.

Moderator powers

On Reddit the content locked into a specific sub-reddit can have a moderator team with an agenda and powers to enact it:

- deletion of comments & hiding the deletion with CSS
- deletion of posts
- deletion of posts about deletion of posts
- deletion of posts about requesting removal of mods (censoring community desire for new filter bubble)
- opportunist adjustment of comment ordering
- shadowbanning
- 30/7-day bans

A good example of this is what happened on r/Bitcoin:

Lack of history

On Reddit, you see content from subreddits you are subscribed to. Typically ordered by an opaque algorithm called "hot". Users can vote, but there is no real way to see how this affects what content is displayed. Large sections of the site will sometimes disappear.

Content is locked into the platform

Each of these platforms works in a certain way and has a legal entity that has control over the platform. If a developer wishes to innovate and change how a platform works they need to create an entirely new platform with new content. APIs are available to interact with the existing solutions, but these tend to be limited in scope and do not alow the fundamentals how how a platform operates to be changed.

User tracking

Some platforms, like Facebook, only allow content to be viewed when logged in. This makes it extremely difficult to browse with anonimity.

2.3 How does it work?

MIX is an Ethereum blockchain. A blockchain is a shared database that has a built-in cryptocurrency that is used to financially incentivize the neutrality of the database. No-one can be in charge of a blockchain.

The first blockchain was called Bitcoin. Pretty much the only thing in the Bitcoin database is how much Bitcoin each account has.

Ethereum takes this concept much further. Computer programs called smart contracts can be uploaded into the blockchain and these programs can then store data in the database that everyone has a copy of.

Many big projects are deployed on the Ethereum blockchain. But this presents a problem. One must assume that every software project has fatal bugs that will need to be fixed once they are discovered. Bitcoin had fatal problems early on that needed to be fixed and so did Ethereum.

Unfortunately, when the projects that are deployed on Ethereum need to be fixed they will have a very hard time because they will need to convince the entire blockchain to deploy their fix. This was the problem with The DAO crowdfunding project. An attacker stole $150m because of a bug in the smart contract. In attempting to fix this problem the Ethereum was split into two blockchains: Ethereum and Ethereum Classic.

The only purpose of the MIX blockchain is to run MIX. This means that whenever there is a problem that can only be fixed with a hard fork, it will not be difficult to convince the MIX community to adopt it.

The cryptocurrency of the MIX blockchain is also called MIX.

2.3.1 Item Store

The principle smart contract on the MIX blockchain is itemstore. It uses IPFS as the underlying storage layer. Anyone can store an item get back a itemId. All items are publically readable by anyone.

Item Store has a rudimentary revisioning system so blobs can be updated while retaining their itemId and revision history.

2.3.2 Google Protocol Buffers

In order for the data stored in BlobStore to have meaning, there must be some sort of schema. Google Protocol Buffers (Protobuf) is ideal for this. It has an upgradable schema system. This means that MIX has a system of content type inheritance. A standardized story type could be extended with new fields and still be readable by software that only knows how to read standard stories.

Anyone can make a new type. Example types are user profiles, media metadata, tweets, reddit posts, blog posts, business information, reviews, etc.

Protobuf encodes the data very tightly. This data is then compressed with Google Brotli. This minimizes the cost of storing information on MIX.

2.3.3 Additional smart contracts

Anyone can deploy additional smart contracts onto the MIX blockchain. An important example would be a feed contract that would provide functionality akin to RSS / Twitter / YouTube.

Other examples would be contracts to vote for content or tag content.

2.4 Issuance

Project revenue of 55 million MIX (equivalent to 5 years of mining) is pre-allocated into the revenue smart contract. It will release it over 2000 days at a rate that decreases every 200 days.

The purpose of the revenue is to provide funds for re-investment into the MIX ecosystem and to provide profit for the developer.

The block reward is the same as Ethereum. 5 MIX per block / 30k MIX per day.

2.4.1 Issuance rate

2.4.2 Cumulative issuance

2.4.3 Post-revenue inflation rate

2.4.4 Live circulating supply API end point

https://github.com/mix-blockchain/mix-supply

http://supply.mix-blockchain.org/

2.5 Configuration

Network ID: 76

Chain ID: 76

MIX Blockchain can be synchronized with either Parity or MIX Geth.

When receiving a large deposit it is essential to wait for 64 blocks before considering it as finalized.

2.5.1 Parity

Get Parity from https://www.parity.io/ethereum/

```
chmod +x parity
./parity --chain mix --pruning=fast --pruning-history=64 --pruning-memory=0
```

The pruning options are essential to mitigate the risk of a 51% attack.

2.5.2 Geth

Download MIX Geth: https://github.com/mix-blockchain/mix-geth/releases

```
chmod +x mix-geth-linux
./mix-geth-linux
```

2.6 Filter bubbles

500 million tweets are published every day, but if you go to Twitter you will only see a small subset of these. Every platform has its own filter system to decide which content you see.

On Twitter you see the latest tweets from people you follow (minus those who have been banned), some sponsored tweets, with a "While you were away. . . " section at the top generated by unknown means. People are sometimes banned entirely from the platform.

On Reddit, you see content from subreddits you are subscribed to. Typically ordered by an opaque algorithm called "hot". Users can vote, but there is no real way to see how this affects what content is displayed. Large sections of the site will sometimes disappear.

On Medium content creators are sometimes ordered to change their content under threat of it being removed altogether.

Each of these communities is siloed from each other. They control the filter bubble within their platform.

MIX is fully programmable at every level. Anyone can take existing content on the system and develop new ways to select which content each user should see, or even utilize data from another filter bubble, for example analysing existing votes in a new way.

2.7 Light client

Currently to interact with MIX in a fully decentralized way you need to have a fully synchronized node running locally. It can take a long time to sync and will take up a lot of storage space.

This will all change with Ethereum's light client functionaity that is rapidly nearing maturity. Both Geth and Parity will be able to serve light clients and act as light clients themselves.

It will also be possible for Android and iOS devices to have light client functionality.

2.8 Smart contracts

2.8.1 MixItemStoreRegistry

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0xb7aead157809d83234ae1a9ac42d8846ebceba6e
```

2.8.2 MixItemStoreIpfsSha256

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0x26b10bb026700148962c4a948b08ae162d18c0af
```

2.8.3 MixItemStoreShortId

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0xf40f0ae20067f5095e2b5fe1c21da8b8e61d3cac
```

2.8.4 MixItemDagOneParent

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0x341518c5b28d3564b39ab7560e47b4486ddb762a   - mixin hierarchy
contract: 0x2a37382ea172d0a28905052ee79f802cd7fd74b4   - comment hierarchy
```

2.8.5 MixItemDagOnlyOwner

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0x622d9bd5adf631c6e190f8d2beebcd5533ffa5e6   – feed items
```

2.8.6 MixItemDagOneParentOnlyOwner

Maintains bidirectional connection between content and the token that should be burned to endorse it.

```
solc version: 0.5.12
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0x45093c3dc9e494ac447dc9f4108e436536e7cbd3   – token items
```

2.8.7 MixItemTopics

```
solc version: 0.5.11
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0xcc64d1519d4e2be2b025204f5b3470d5f14a1a99
```

2.8.8 MixItemMentions

```
solc version: 0.5.11
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0xed279A14f93dDF8dCE9A73B4b281687051214E87
```

2.8.9 MixAccountRegistry

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0xbcab5026b4d79396b222abc4d1ca36db10984c73
```

2.8.10 MixAccountProfile

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0x994abe0212b5dcc1fb0b0e7336e7980316c3fe19
```

2.8.11 MixAccountItems

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0xc9ba9507d9f5be1d13ff2dca6f7e43dbfa859645 – account feeds
```

2.8.12 MixAccountItems

```
solc version: 0.5.12
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0xcea97006bb67782e7c4d70246a23b048c2029c90 - account tokens
```

2.8.13 MixTrustedAccounts

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0x70e2e2d6b31cd25e00c034ac9cfc79575efa26a9
```

2.8.14 MixReactions

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0xd7051cd496a3a8373f9cf89476c04a7d51a5cc88
```

2.8.15 MixTokenRegistry

```
solc version: 0.5.12
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0x27e21076d763fc4a83bb3a220369d27f67d6767f
```

2.8.16 MixTokenBurn

```
solc version: 0.5.12
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
contract: 0x8f7e10a2a7255cc90720d34ccd3a649cac91c3de
```

2.8.17 MixCreatorToken (Acuity v1.0.5)

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
```

2.8.18 MixCreatorToken (Acuity v1.0.8)

```
ERC-165: 0x23fb80f7 (MixTokenInterface)
solc version: 0.5.11
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
```

2.8.19 MixCreatorToken (Acuity v1.1.0)

```
ERC-165: 0xa66762eb (MixTokenInterface), 0x8da5cb5b (MixTokenOwnedInterface),
→0xd6559ea1 (MixCreatorToken)
solc version: 0.5.12
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
```

2.8.20 UniswapExchange

```
contract: 0x82c1eb1140068a71d6905a78faf66c0fe9863531
```

2.8.21 UniswapFactory

```
contract: 0x1381a70fc605b7d7e54b7e1159afba1429a4bbb1
```

2.8.22 LinkRevenue

```
solc version: 0.4.10
solc parameters: --optimize
contract: 0x97c7f4f8f0bbf384578a9f5754ae73f37ff49ec2
```

See *Issuance* for more information.

2.8.23 MixAccount (Acuity v1.0.0)

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
```

2.8.24 MixAccount (Acuity v1.0.3)

```
solc version: 0.5.10
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
```

2.8.25 MixAccount (Acuity v1.0.7)

```
ERC-165: 0x527f66d8 (MixAccountInterface), 0x4e2312e0 (ERC1155TokenReceiver)
solc version: 0.5.11
solc parameters: --evm-version byzantium --optimize --optimize-runs 10000
```

2.9 Mine MIX

In order to publish content on the MIX network, or execute any kind of transaction, you must spend some MIX. MIX can be obtained either by mining it or by purchasing it.

In order to mine MIX, you will need to have a fully syncronized node as described in *Configuration*.

You need to have a MIX account. If you do not have one already, go to the MIX console and run `personal.newAccount();`.

Alternatively an Ethereum account can be created with MyEtherWallet and that can be used for MIX.

2.9.1 CPU Mining

CPU mining with Geth is very easy. Run the regular Geth command, but add the `--mine` option. Use `--etherbase` to specify which account will receive the mining rewards. It defaults to using all processor cores. This can be changed using the `--minerthreads` option.

2.9.2 GPU Mining

Genoil

Ubuntu 15.10 or Newer. OpenCL only (for AMD cards)

```
sudo apt-get update
sudo apt-get -y install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install git cmake libcryptopp-dev libleveldb-dev libjsoncpp-dev
→libjsonrpccpp-dev libboost-all-dev libgmp-dev libreadline-dev libcurl4-gnutls-dev
→ocl-icd-libopencl1 opencl-headers mesa-common-dev libmicrohttpd-dev build-essential
→-y
git clone https://github.com/Genoil/cpp-ethereum/
cd cpp-ethereum/
mkdir build
cd build
cmake -DBUNDLE=miner ..
make -j8
```

Ubuntu 15.10 or Newer. OpenCL + CUDA (for NVIDIA cards)

```
wget http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1404/x86_64/cuda-
→repo-ubuntu1404_7.5-18_amd64.deb
sudo dpkg -i cuda-repo-ubuntu1404_7.5-18_amd64.deb
sudo apt-get -y install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install git cmake libcryptopp-dev libleveldb-dev libjsoncpp-dev
→libjsonrpccpp-dev libboost-all-dev libgmp-dev libreadline-dev libcurl4-gnutls-dev
→ocl-icd-libopencl1 opencl-headers mesa-common-dev libmicrohttpd-dev build-essential
→cuda -y
git clone https://github.com/Genoil/cpp-ethereum/
cd cpp-ethereum/
mkdir build
cd build
cmake -DBUNDLE=cudaminer ..
make -j8
```

CD to the `ethminer` subfolder and run the following command

```
./ethminer -G -F http://localhost:8645
```

Claymore

Download from https://github.com/nanopool/Claymore-Dual-Miner/releases

Linux

```
./ethdcrminer64 -epool http://localhost:8645 -allcoins exp
```

Windows

```
EthDcrMiner64.exe -epool http://localhost:8645 -allcoins exp
```

There is great a tutorial for mining MIX with Claymore on Windows: https://klmoney.wordpress.com/link-blockchain-windows-gpu-mining/

2.9.3 Mining Pools

MIX mining pools can be found at https://www.mix-blockchain.org/

2.10 Purchase MIX

MIX cryptocurrency is required to publish on the MIX platform.

2.10.1 STEX

- https://app.stex.com/en/trade/pair/ETH/MIX

2.10.2 Graviex

- https://graviex.net/markets/mixbtc
- https://graviex.net/markets/mixeth

2.11 Contact

If you have any question whatsoever, please email contact@mix-blockchain.org

16 **Chapter 2. Table of contents**

CHAPTER 3

Indices and tables

- genindex
- modindex
- search