
MIX Blockchain Documentation

Release 1.0

Jonathan Brown <jbrown@mix-blockchain.org>

Apr 19, 2018

Contents

1	Useful links	3
2	Table of contents	5
2.1	The Vision	5
2.2	The Problem	5
2.3	How does it work?	7
2.4	Cryptofuel	8
2.5	Issuance	9
2.6	Configuration	9
2.7	Filter bubbles	10
2.8	Light client	10
2.9	Smart contracts	10
2.10	Mine MIX	12
2.11	Events	14
2.12	Contact	15
3	Indices and tables	17

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).

CHAPTER 1

Useful links

- [Homepage](#)
- [Blog](#)
- [Twitter](#)
- [Reddit](#)
- [Gitter](#)
- [YouTube](#)
- [BitChute](#)
- [Block explorer](#)
- [Network stats](#)
- [Issue tracker](#)
- [Logo](#)

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 unalienable 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 criticized 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 allow 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 anonymity.

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 Cryptofuel

Name: MIX

Code: MIX

Symbol:

[BIP44](#) coin type: 76 / 0x8000004c

All transactions on the MIX Blockchain must be paid for with MIX. Transaction size is measured in “gas”. Transactions can be prioritized by paying a higher gas price.

2.5 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.

2.5.1 Issuance rate

2.5.2 Cumulative issuance

2.5.3 Post-revenue inflation rate

2.5.4 Revenue live status

2.5.5 Live circulating supply API end point

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

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

2.6 Configuration

Network ID: 76

Chain ID: 76

Port: 30313

RPC Port: 8645

WS Port: 8646

RPC TLS Port: 8647

```
git clone https://github.com/mix-blockchain/mix-blockchain.git
```

MIX Blockchain can be synchronized with either Geth or Parity:

2.6.1 Geth

Install Geth as described here: <https://github.com/ethereum/go-ethereum/wiki/Building-Ethereum>

```
geth --config mix.toml --datadir ~/.mix-geth init genesis.json
geth --config mix.toml --datadir ~/.mix-geth --rpc
# In a separate terminal launch the console.
geth attach ~/.mix-geth/geth.ipc
```

2.6.2 Parity

Get Parity from <https://parity.io/>

```
parity --chain mix.json --port 30313 --jsonrpc-port 8645 --geth
```

2.7 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.

2.8 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 functionality 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.

2.9 Smart contracts

2.9.1 Revenue

Source

Deployment address: `0x97c7f4f8f0bbf384578a9f5754ae73f37ff49ec2`

Solidity 0.4.10 with optimizations.

See *Issuance* for more information.

2.9.2 Item Store

Item Store is the principle smart contract system for MIX. All base content is registered with Item Store. **IPFS** is used as the storage layer.

Useful links

- [Source code](#)
- [Issue tracker](#)

Properties

Item Store has the following properties:

- **Timestamped** The approximate time that every item is published is recorded on the blockchain.
- **World-readable** Every item published can be read by anyone. The only way to avoid this would be to encrypt the item before publishing it.
- **Immutable** While an item can be “retracted”, it can never really be deleted because the transaction that created it will be archived for eternity on full nodes.
- **Revised** Item Store has a rudimentary revisioning system built-in where an item can have multiple revisions, e.g. for editing posts. More sophisticated revisioning systems can be built on top of Item Store where each item is a revision.
- **Ownership** Each item can have an owner. Only the owner can modify an item, change item settings, or transfer ownership to another address.
- **Configurable** Each item has the following flags that can be set:
 - **Updatable** The contents of the item can be changed. Once disabled this flag cannot be re-enabled.
 - **Enforce Revisions** When updating the item a new revision must be created. It is not possible to retract revisions. Once enabled, this flag cannot be disabled.
 - **Retractable** The item in its entirety can be retracted. This is unaffected by Enforce Revisions. The itemId of a retracted item can never be used again. Once disabled this flag cannot be re-enabled.
 - **Transferable** The item can be transferred to another user (if they accept it), or disowned completely. Once disabled this flag cannot be re-enabled. At creation time items can also be flagged as anonymous to not have an owner associated. An alternative to transferable items is to use a proxy account with transferable ownership as the item owner.
- **Scalable** Only a bare-minimum of information is stored in contract state. IPFS is used for actual content storage.
- **Upgradability** Item Store is an upgradable system. Due to a security vulnerability, new storage system, or gas cost improvements a new Item Store contract may be deployed.
- **Unit tests** Item Store has tests written in Solidity using the [Dapp](#) framework.

itemId

Each item has a 20 byte itemId that is unique to the contract that generated it. The itemId can be further classified to indicate which contract the item is on.

- **Off-chain** There is considered to be an ordered list of Item Store contracts. For example, the first contract would be #0. The next one would be #1. 20 byte itemIds can be prefixed to indicate which contract the item is in, or this can be assumed from context.

Client software should have a hard coded whitelist of Item Store contracts that it knows how to read from.

- **On-chain** New Item Store contracts register with the Item Store registration contract. When an `itemId` is stored in a smart contract, it must be stored with the 12 byte Item Store `contractId`. This way the smart contract can look up the address of any Item Store contract in the registration contract. This is essential to future-proof smart contracts that need to communicate with Item Store contracts.

Deployments

All current deployments of Item Store contracts on MIX were compiled with Solidity 0.4.16 with optimizations enabled.

Git hash: `b517451720bcb59d8f731974dfefbf80d2ade57d1` [link](#)

ItemStoreRegistry contract address: `0x3ea61ec502f053ee89b906fdc3abe2e5f9eb4a7f`

Item Store #0

IPFS with SHA256 hash.

contract address: `0x2af783f7d5a607098057ce715b55a82e707a7cab`

BlobStore `contractId`: `0xb639900808b2633054720f33`

2.9.3 Mixin Registry

Source

Deployment address: `0x99be57fca49270b69306cdbe89df416d769dbd06`

Solidity 0.4.16 with optimizations.

Every item of content stored on MIX is composed of one or more mixins. Examples of potential content types are tweets, comments, blog posts, media metadata and user profiles.

MIX has a hierarchical system of mixins. There is a [root mixin](#) that all other mixins are derived from.

Each mixin is identified by an integer, the `mixinId`. The `mixinId` for each mixin is issued by the Mixin Registry smart contract. The `mixinId` of the root mixin is 0.

When creating a new mixin, the `addMixin()` method must be called. `parent` must be the `mixinId` of the mixin that is being extending. `uri` should be a link to an immutable commit in a repo that is derived from the parent mixin.

2.10 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 synchronized 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.

2.10.1 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.

2.10.2 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
↳libjsonrpcpp-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
↳libjsonrpcpp-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/>

2.10.3 Mining Pools

- MINERPOOL.NET
- EncryptGlobe

2.11 Events

2.11.1 2018

Anarchapulco (Gold Sponsor)

February 15-18
Princess Mundo Imperial
Acapulco, Mexico

MIX Party San Francisco

Thursday, March 15, 19:00 - 22:00
Brick & Mortar Music Hall
1710 Mission St
San Francisco

Satoshi's Vision (attending)

March 23-25
Tokyo

2.11.2 2017

Bangkok Satoshi Square

Monday, June 26, 18:30 - 21:30
American Bar & Grill
4/26 Sukhumvit Soi 8
Bangkok

Kuala Lumpur Blockchain Evening - video

Friday, July 14, 19:00 - 22:00
Einstein Boardroom
WORQ
Level 3A, Glo Damansara
No. 699, Jalan Damansara
Kuala Lumpur

Dojo Bali Cryptocurrency Mastermind (Dojo members only)

Tuesday, August 1, 19:00 - 21:00
Dojo Bali Coworking Space
No. 88 Jalan Batu Mejan
Canggu, Echo Beach
Kuta Utara, Kabupaten Badung
Bali, Indonesia

MIX Blockchain: The successor to the World Wide Web

Thursday, August 17, 20:00 - 23:00
Care Hub Cafe
30 Bà Huyn Thanh Quan
H Chí Minh City, Vietnam

2.12 Contact

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

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`