Ngin: Search Engine Based Ecosystem

1. Inroduction

Ngin (pronounced “engine”) is a new search blockchain-based search engine. Our audacious goal is to build a decentralized system will bring new benefits to users that large centralized systems no longer can.

1.1 Challenge

Search engines form part of the Internet’s critical infrastructure. A search engine is typically where a user starts their journey online and it’s their guide to the web’s vast resources. Without a search engine, the 5B people who have Internet would have great difficulty combing through trillions of web pages. Over the past decade, however, the way search engines operate has drastically changed. Privately-owned search-engine companies are subject to the laws of more than 193 countries around the world and must develop means to address each of those different laws. Competition between laws, businesses and people has resulted in a landscape that’s highly concentrated with just one or two major operators in each market.

Today, searches are no longer the kind and quality that they once were: we don’t get the same kind and quality of information that we once used to get. Searches are faster from these companies than ever before and they more curated, but the options for users are more restricted because of imposed search-term filtering. We are also increasingly concerned about the privacy issues that these companies handle on users behalf. We want to break that market and give users another option with blockchain.

1.2 Vision

As stated above, centralized search engines no longer address our needs. We want to bring search back into the hands of users. We launched Ngin as a no-premine, community project in order to bring an alternative blockchain-based search engine to users.

2.0 Project Overview

We have a broad vision of how we want to develop search powered by blockchain: we want clean, untraceable, portable, convenient, no-ads, even profitable features.

Once this model is working, we envision several ways that AI can augment the experience for users and the public. In just one example, we might develop a product that allows users to tailor their own needs and formulate requests that are theirs alone.

2.1 Technical Introduction

We've written a custom miner that uses Ngin in a way that allows users to have dual-use of their resources. At a high level we have Proof of Fetch and Proof of Work at the same time.

- *Proof of Fetch (PoF)*. This our term for search-engine spider activity. While running the mining software, when a search is requested, the CPU crawls the web, pulls the information and is rewarded with NGIN for the effort.
- *PoW Mining.* CPU cycles are used for PoW mining in parallel to mine the NGIN.

As the project and blockchain develops, we anticipate collaborating with the community on changes that may be needed to optimize the ratio of PoF to PoW, as well as other variables.

2.1.1 Custom daemon

We launched PoW at first to gather more attention and to build the community. Over the course of the next few months, the every part of daemon `(ngind)` will be replaced with FuncNodes' code. This will be the transition into the PoF phase.

2.1.2 Blockchainspam and webspam

We anticipate spam to be a major challenge, and to reduce spam, noise, and phishing the system will check every page and each url & detail by "pool" nodes. Once the url & page has been sent to FuncNodes, and has then passed 10 other nodes' confirmations, the miner may get more rewards than hashing.

2.1.3 Mining software

Ngin has developed its own mining software in order to accomplish the dual use for PoW/PoF (although the miner is currently only working on PoW). Similarly, we have customized software for the mining pools to accommodate our network. All of our coding is provided open-source.

2.1.4 Future aspirations

By combining Ngin with the search functions, there are several combinations and experiments that can be undertaken. For example, we might be able to offer some kind of "premium search" or other functions that would incentivize miners and/or incentivize the free market (who may not be mining) to purchase Ngin. Some of these advanced features might include:

- user-selected privacy settings;

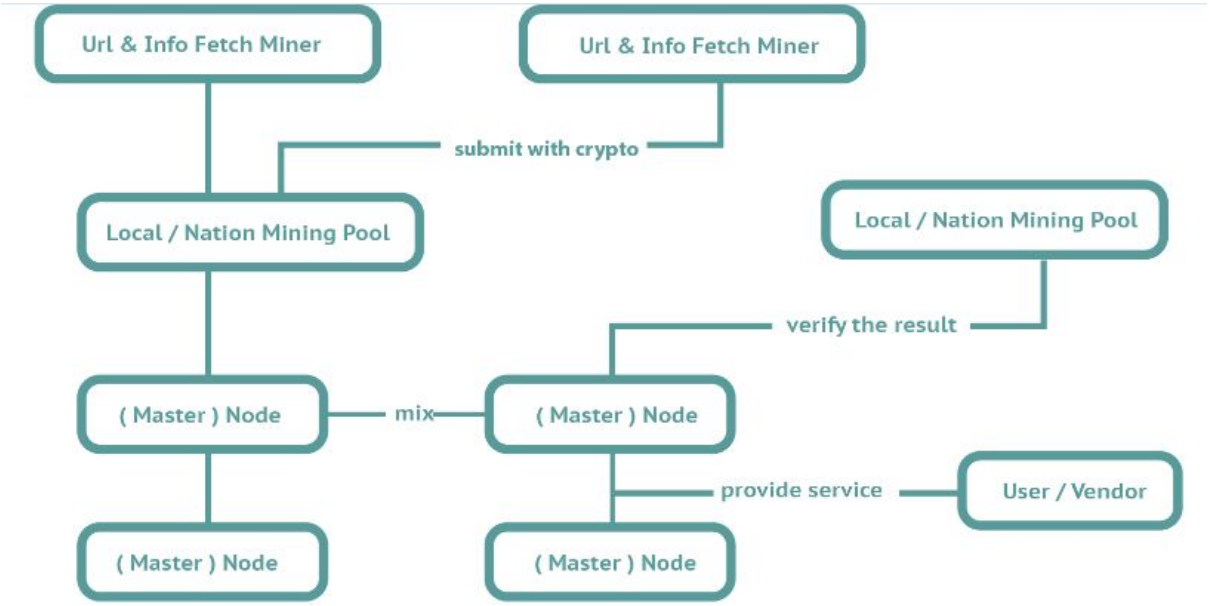
- ➢ advanced anonymity products that provide additional security;
- ➢ a faster search experience (e.g., premium vs. standard)

3.0 Ecosystem integration strategy

The Internet was built on open standards with an invitation to any actor or player to join, so long as they cooperated by using the same standards and tools. Similarly, the blockchain is built on open standards and has a rich exchange of innovation.

Badly needed: Blockchain APIs. In spite of the growing repositories of open code, an application like Ngin needs new APIs that can interface with others in the ecosystem: other blockchains, the Internet, and for certain purposes search engines like Google, Bing and DuckDuckGo. We plan to develop many of these APIs ourselves and help build the ecosystem to support it.

Below is an illustration of how Ngin will connect with the world.



3.1 Potholes

The advantages for a decentralized blockchain are numerous; but so are the potholes. These are some of those challenges that we are thinking through. These are the kinds of things that decentralized communities of thinkers can help resolve:

3.1.1 Speed and latency

Ngin will provide vaset advantages in privacy, anonymity and other features, but one of the most important tradeoffs may be speed. Anyone that uses Google notices the report on how long it took, frequently a small fraction of a second for thousands of results. We do not

expect Ngin to operate with that kind of efficiency because it's optimized for other things. However, user experiences will want speed.

- ➢ We believe that we can solve for some of this by developing APIs ourselves that would interface with the centralized search engines so users can have the experience they know;

- ➢ *In addition*, users will have privacy, anonymity, and additional features available to them (including finding otherwise censored or removed content) that may not currently be available to them.

In short, we recognize the loss of speed and efficiency, although we believe that we can mitigate this by merging the experience with other anonymity, privacy and scope options.

3.1.2 Scale limitations and partner development.
As the Internet grows to 1,300 Petabytes we recognize the challenge of building a decentralized system on top of one that's already been efficiency centralized by large companies.

- ➢ We see the challenge ahead of us as one that requires blockchains to build on each other and to offer a growing, adaptive solution to search needs.

- ➢ In order for this to be realized we envision a future with the possibility of multiple side chains specific chains that can integrate with the ecosystem and allow infinite scaling.

- ➢ Our ecosystem vision and our collaborative philosophy may lead us to cooperation with other similarly-minded, decentralized projects (e.g., for storage, AI, exchanges, etc).

3.2 Incentive Mechanism

Although Ngin is a mined cryptocurrency it's service within our blockchain acts as a *token,* because it has both a compensation function (in PoW) and a confirmation function (in PoF). either, and ngin is the only coin/token in our network.

Currently all coins are POW, and in the future we will transition to MN, FuncNode, PoF. We do not yet have a fixed date for the transition but we commit to transparently discus with the community and make the decision together with them.

3.3 Backend

The main Ngin backend program is a fork of `geth` called `ngind` (means Ngin Daemon). As the result, all interfaces like `jsonrpc` and `web3` are similar to Ethereum. So, if a user needs to convert an Ethereum contract to an Ngin component, we just need to change all `"web3 methods"` to `"eth,"` and the prefix into `"ngin."` We've designed our code in this way so as to integrate with the Ethereum blockchain while maintaining Ngin's short block times, decentralization, and other features.

4.0 Tech Spec

More about the technical foundation of Ngin.

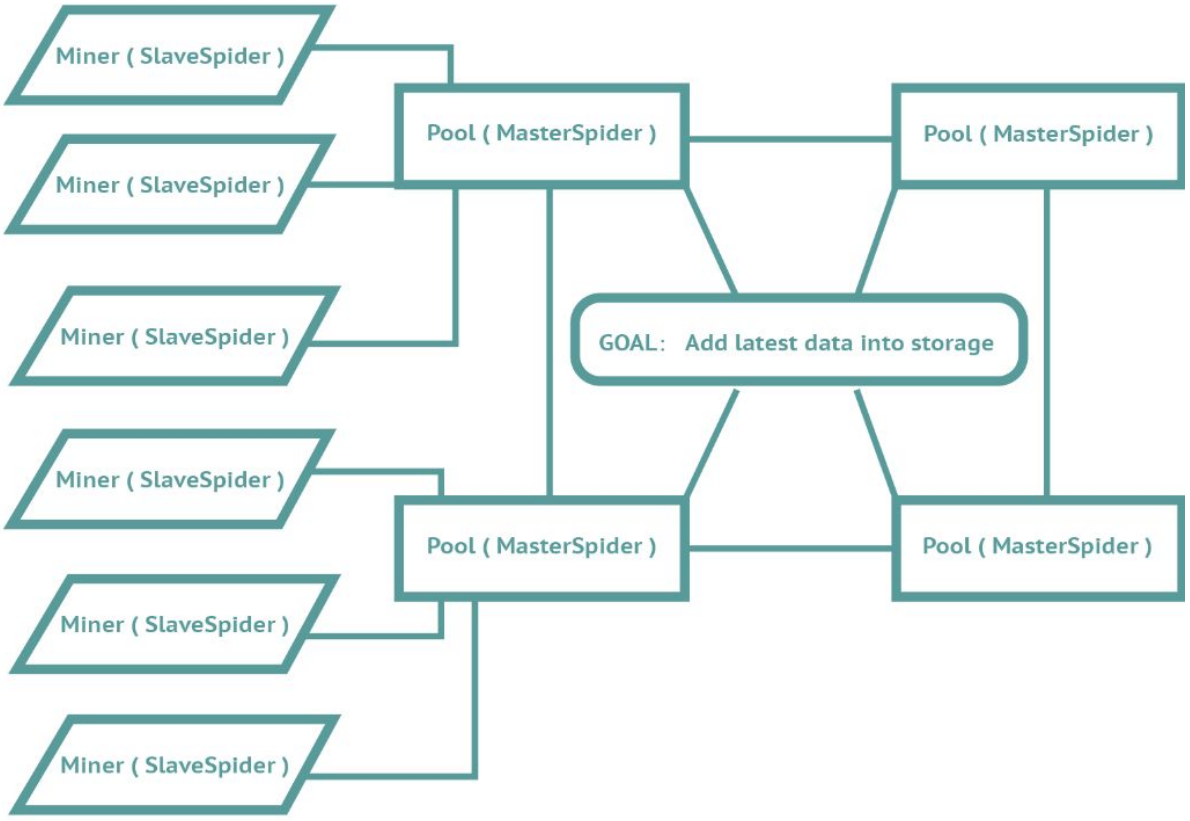
4.1 Spider (Fetch & Update)

The Spider will follow PoF, introducing two working methods to the Spider function.

First is the Peer to Peer, aka `p2p`. The blockchain is constituted with several peer nodes. When a node/spider fetches a new/updated website info packet, it will format it and compress it into a `msg`. The `msg` will flow from one to another to make itself checked by most nodes. After that, all nodes' database will appended the `msg`

Second, to alleviate some of the inefficiencies of p2p networks, the Ngin PoF has a master-slave architecture (MSA). Almost all search engines work in an MSA architecture and `Fetch` task will be divided into different part. For example, the `Slave` will download raw data, and the `Master` will do more on data format and assign small tasks to `Slaves`, even manage database and make indexes.

By combining the two we get a more resilient and efficient structure as illustrated:



4.2 Index

Search engines take the query to pull their results from centralized databases and libraries. Deploying RDMS like MySQL and MsSQL results in inefficiencies and latency. One of the reasons for the latency is the lack of full-text index. Here the `PostingList` is key (recall: this is the inverted index data structure). Once a forward index is developed it is then itself inverted to `PostingList`. This is a more efficient result than querying the forward index because no sequential iteration is needed to validate the matching document.

5.0 Aspirations

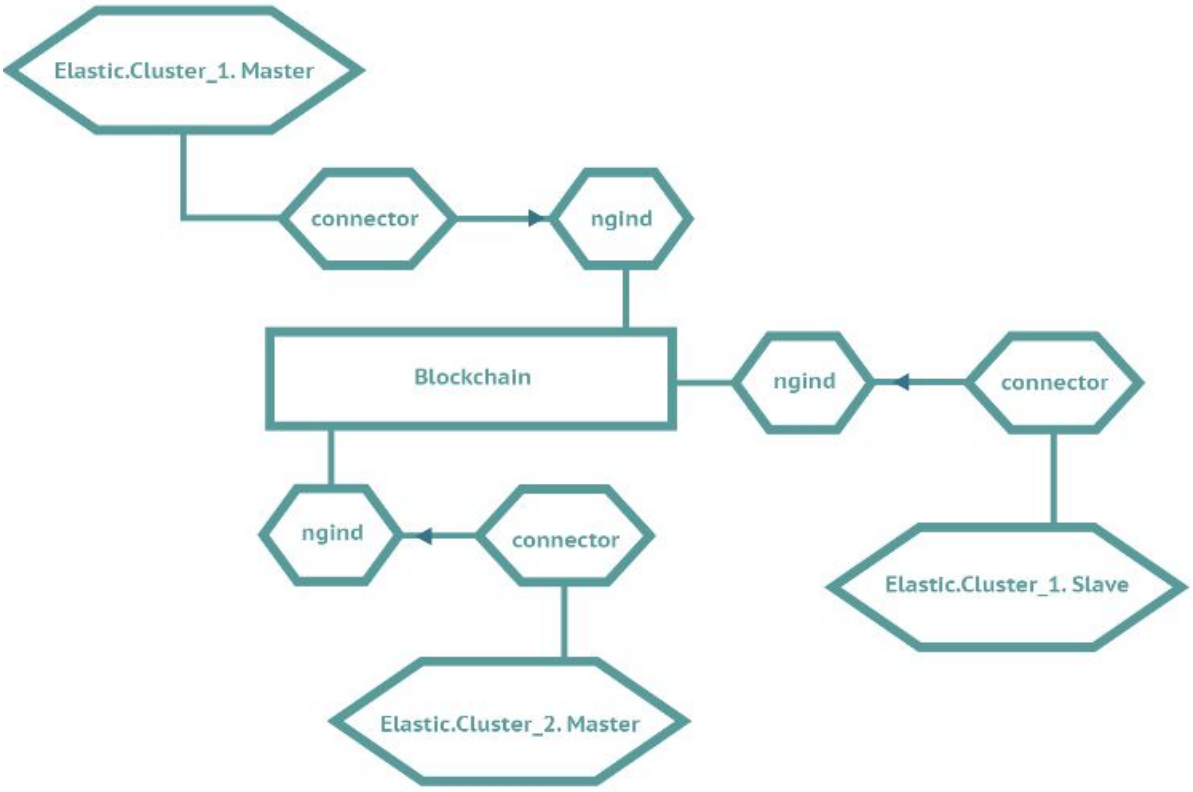
Ngin's goals and aspirations are very long-term. Some of the initiatives that we have started for the project include the Elastic Integrated Network and several improvements to mining.

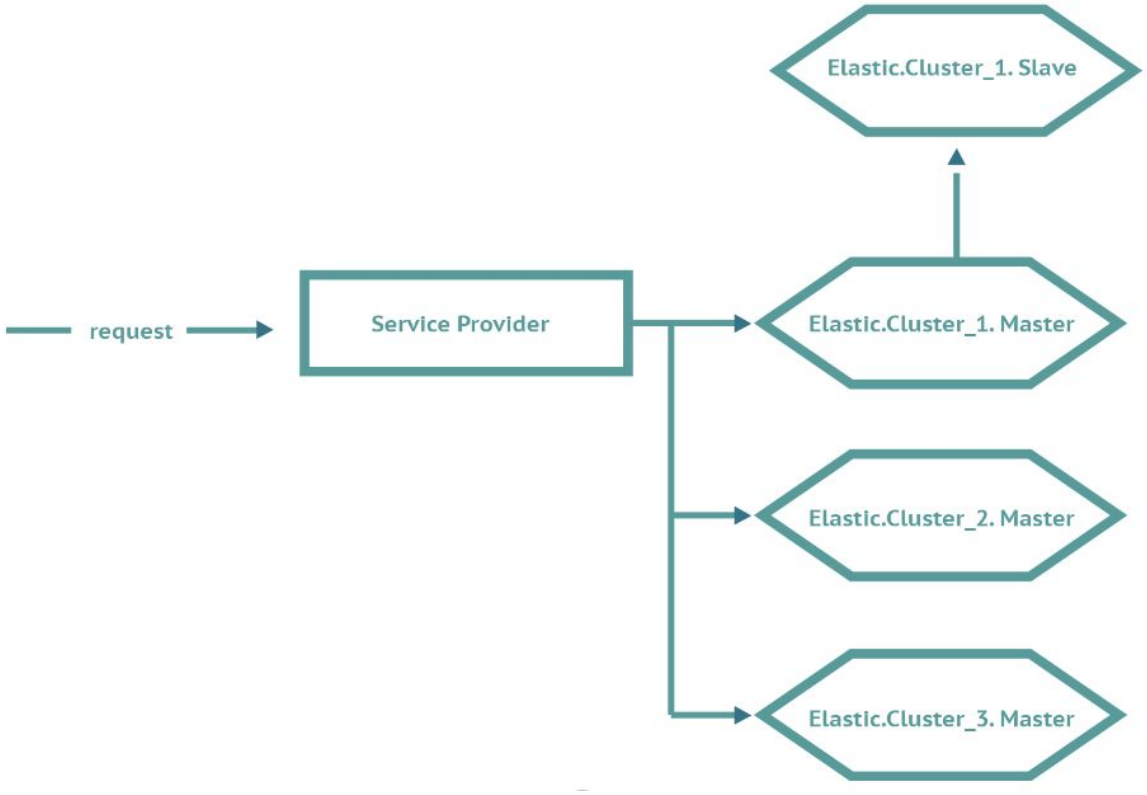
5.1 Elastic Integrated framework

The integrated framework that we are developing includes three products: `Elastic` (Lucene) + `connector` (e2eBridge) + `ngind` (contract).

ElasticSearch (Elastic) is a search engine based on Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. ElasticSearch is developed in Java and is released as open source under the terms of the Apache license. According to the DB-Engines ranking, ElasticSearch is the most popular enterprise search engine followed by Apache Solr, also based on Lucene.

In this solution, Elastic acts as a search engine & local database storage solution provider. And the Ngin network just acts as the data conveyor.


Elastic.Cluster_1. Master
connector
ngind
Blockchain
ngind
connector
ngind
connector
Elastic.Cluster_1. Slave
Elastic.Cluster_2. Master



.
.
....
.

5.2 Code rebase to golang (ported from ngind)

We are in the mid-phase of porting ngind to golang. The new design will be ours from top-down and and Ngin will not only work as the data conveyor but also as the decentralized index and for searching tasks. At that point our internal project is not only a blockchain solution but includes local search; also, it's portable, scalable, accurate and powerful. W

5.3 FuncNodes Types & Rewards:

We are currently experimenting with three different plans, each with different rewards:

5.3.1 Plan #1

Server	**Description**	**Rewards**
Elastic	Full-content search on blockchain	HIGH
Pool	Updates data	MEDIUM
Host	Web-based search services	SMALL
Minimal	Online required, data cync	SMALL

Server	**Description**	**Rewards**
Miner	Add instance to the blockchain	MEDIUM
Search user	Rewards for click (in case of ads)	LITTLE

5.3.2. Plan #2

Server	**Description**	**Rewards**
Miner		MEDIUM
NodeHolder	ngind	MEDIUM
PoolNodeHolder	ngind + ngPool	LARGE

6.0 Frequently Asked Questions

Q: Is this project a scam?
A: This is one of the most common questions worth asking about any project. No, it's not a scam, and it would be very difficult to pull off a scam from a project that has no premine.

Q: How do team get benefit from this project?
A: Ngin isn't just building another cryptocurrency, our ambition is much larger. We're building a new ecosystem in a decentralized way to challenge some of the world's largest companies. We see this as a very long term future and do not intend to prescribe the model today (similarly, Google famously declared that it had no idea how it was going to make money until the ads opportunity presented itself, and still represents more than 90% of the company's income). Within this there are several more specific areas of benefit, like: (1) arbitraging price between posting ad's price and reading ad's reward; (2) consulting and professional services for enterprises; (3) development of additional value-added AI products like search assistants.

Q: Why don't choose DApp?
DApp was our first choice, however, when we looked at our roadmap for the long run, we realized that we could not depend on another chain or ecosystem. Although a master node is possible on ETH and EOS, our FuncNode is not, and things get more complicated with our unique miner, how it interfaces with pools etc. That's why we think that using our public chain would give more options for us as developers and we would also have more control over the development of the chain, rewards, any relevant monetary policy, etc.

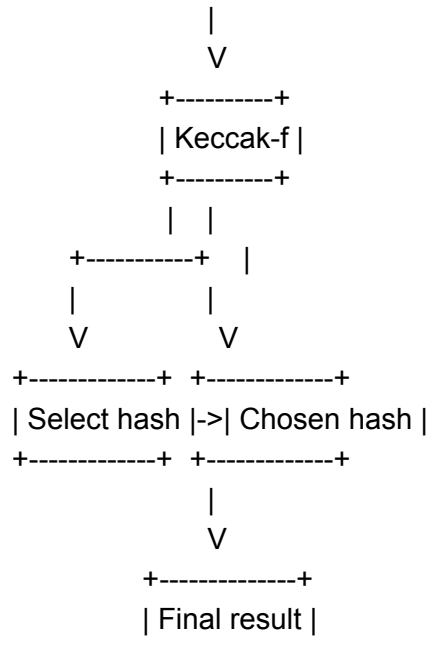
Q: Why didn't you choose a token?
A: Because tokens can have blockchain characteristics, but our goal is to build a new blockchain based on community principles, no premine, and a long-term vision. For this the best organizational character is as a community project, because this enables any member of the team to remain anonymous and to avoid the problems with ICOs.

Q: Why does your White Paper keep changing? It looks different than before.
A: We're focusing on our strengths: that's coding, not prose in English. Luckily an anonymous but highly qualified community member (who refused any payment) worked with us in the past few days to understand our intent so that it could be delivered more accurately. Note: if there are any English tech writers that would like to join our community, we would community help. We realise that we need to tell our story but we'll do it better with some support.

7.0 Launch Algorithm - M00N

We developed the M00N algorithm for our project. M00N Is a variant of Cryptonight although we modified it to require only 1MB for the scratchpad. As the result, M00N was born. M00N is an algorithm which frequently change its built-in sub-algorithm.

M00N is a variant of cryptonight, but it takes only 1MB for the scratchpad. Additionally M00N adds a new algorithm into the last "Select hash" step: instead of the default cryptonight is 0=BLAKE-256 [BLAKE], 1=Groestl-256 [GROESTL], 2=JH-256 [JH], and 3=Skein-256 [SKEIN]., current version of M00N adds the BMW algo. We intend to continue the development of our unique algos and the integration with our custom miner. From the cryptonight paper:



+--------------+

We're building Ngin and it's miner so that it can be mined equally on every machine. GPU mining does not bear the efficiencies needed and we are committed to fork if there is any GPU miner on the horizon. (There may be other reasons to fork, for example, for functionalities or features that the team and community may agree are useful).

6.2 Origin fork source

Ngin is a fork of Ethereum Classic.

6.3 Ngin Specs

Name:	Ngin
Ticker:	NG
PoW algo:	M00N
Block Time:	≈ 13 sec
Era Length:	100,000 blocks (~2 weeks)
Reward:	≤ 10 NG
Total Supply:	≈ 360,000,000 NG
P2P Port:	52520
RPC Port:	52521
WS Port:	52522
Premine:	NO
Presale/ICO:	NO
Contracts:	ERC-20, ERC-223

6.4.1 Block Reward

As above, the Total Block Reward is 10 NG. But it's only in Era #0 that each block reward is 10 NG, there is coefficient to calculate the reward in other era.

- $MaxReward = 10$
- $Coefficient = \frac{249}{250}$
- $BlockReward(EraNum) = Coefficient^{EraNum} \times MaxReward$

6.4.2 Circulating Supply

- $EraLen = 100000$
- $CirculatingSupply(EraNum) = \sum_{i=0}^{EraNum} EraLen \times BlockReward(EraNum)$

6.4.3 Community

DISCORD: https://discordapp.com/invite/xxr748Y

TWITTER: https://twitter.com/NginProject

BTC ANN: https://bitcointalk.org/index.php?topic=5041386.0