
Lonero Documentation

Release 0.4.3

The Lonero Foundation

Jun 07, 2023

1	About Us	3
2	What is Lonero?	5
3	Seed Node	7
3.1	API Wiki Stats Status Open Hub Help Center	7
4	Contents	9
4.1	Crypto	9
4.2	Protocol	15
4.3	Governance	16
4.4	Architecture	17
4.5	Make a Mining Pool	17
4.6	OS	19
4.7	LonScript	20
4.8	Documentation	23
4.9	Hardware Cryptography	27
4.10	Main Installation Methods	28
4.11	Up-to-Date Related Academic Work	32
4.12	Lonero ATM Project	32
4.13	Kinetic Energy Miner	34
4.14	PeerTop Lonero OS Laptops	35
4.15	ChainBoard The Next Gen Wireless Dev Board	35

Informational landing page in markdown to go along with the [website](#)

CHAPTER 1

About Us

Lonero utilizes Cryptonote's protocols and our HashBolt masternode consensus, taking security, transaction speeds, and mining profits to a new global [standard](#).

CHAPTER 2

What is Loner?

- Utilizes our HashBolt masternode consensus that allows for mining on ASIC and Non-ASIC systems.
- Our decentralized technology allows both staking and mining in order to keep the technology ahead of the market.
- Loner outperforms leading privacy coins by integrating HashBolt, higher block size, segregated witnesses, as well as a lightning network.
- Prioritizes safe and secure transactions by utilizing Cryptonote's ring signature protocol and our custom multi-layer encryption.

CHAPTER 3

Seed Node

```
host: 142.93.171.115  
port: 34414
```

If seed node doesn't work, try one of the nodes in the [beta](#) config file

Configs: [hpool](#) | [NOMP](#) | [OSMP](#) | [YAAMP](#) | [UMOMP](#) | [P2Pool](#) | [Stratum](#) | [Coinium](#) | [CoinHive](#) | [CPUMiner](#) | [MultiMiner](#) | [MiningCore](#) | [Poolserver](#)

3.1 [API](#) | [Wiki](#) | [Stats](#) | [Status](#) | [Open Hub](#) | [Help Center](#)

CHAPTER 4

Contents

4.1 Crypto

Table of Contents

- *Crypto*
 - *Intro*
 - *Abstract*
 - *Cryptonight Integration*
 - *The Loner Foundation*
 - *What are Merkle Trees?*
 - *Defining Cryptocurrency*
 - *Multi-Path Layer Security*
 - *Analyzing Polynomial Time*
 - *Merkle Mathematical Proofs*
 - *The Ring Signature Protocol*
 - *Problem with Current Mining*
 - *Applying Euclidean Geometry*

4.1.1 Intro

Loner is a newly emerging cryptocurrency that aims to be the future of digital cash. The problem with current privacy coins, alt coins, and emerging cryptocurrencies relies mainly on stability, security, and an

unknown future. We aim to fix many of these stability issues, be extremely decentralized, and have technology that actively supports the community. We are utilizing some of the most advanced technological protocols in order to do so. This includes our consensus protocols that support less barriers amongst ASIC vs. Non-ASIC systems, along with the Cryptonote foundation's widely trusted ring signature protocols, multiple layers of security, and a utilization of "Proof of Work" and "Proof of Space" mining. Among the technology we are also integrating a Decentralized Autonomous Organization or DAO, bounty programs, and a mathematical proof that allows further decentralization. We believe that mining rewards shouldn't merely be limited to hardware specifications but also to total mathematical knowledge, so we provide a fair market game to everyone involved.

4.1.2 Abstract

The components of what makes Lonero unique is split into four different parts: The HashBolt Masternode consensus, the DAO integrations, Cryptonote's ring signature integration, and multipath layer security. In terms of the masternode consensus, we aren't only allowing both ASIC and Non-ASIC systems to profitably mine our cryptocurrency. We are also granting large server infrastructures and x86 computer systems an easier way to mine Lonero. The DAO integration that we are doing is less of an organization and more of an integrative, curated, and regressional artificial intelligence. We set parameters that prevent spamming or abusing of the system to curate within our network and provide a non-redundant experience for our users. This allows us to be fully private, without running into compilations in our infrastructure or allow usage Lonero in cryptojacking or distributed bot nets.

We aim to be one of the industry's first privacy coins that is institutionally compliant while still allowing a libertarian like usage. We support an open free market for our users as Lonero truly is, as stated, the future of digital cash. The utilization of cryptonote's ring signatures, while allowing multi-path layer security creates higher technological stability among the transactional usage of our cryptocurrency. Over time, we aim to be secure enough where financial institutions and FinCEN regulated entities would allow an integration of our system amongst theirs.

Our masternode consensus known as HashBolt allows for an extremely decentralized crypto. It wouldn't just break traditional hardware mining barriers, but components such as difficulty targets don't solely determine technological value. Some mathematical components of our crypto can also be solved by building recursive functions for analyzing polynomial time or integrating merkle tree proofs for certain transactions. The fact that we utilize the ability for segregated witnesses and the algorithmic capabilities of what we are creating, allows for easy scalability of the network over time.

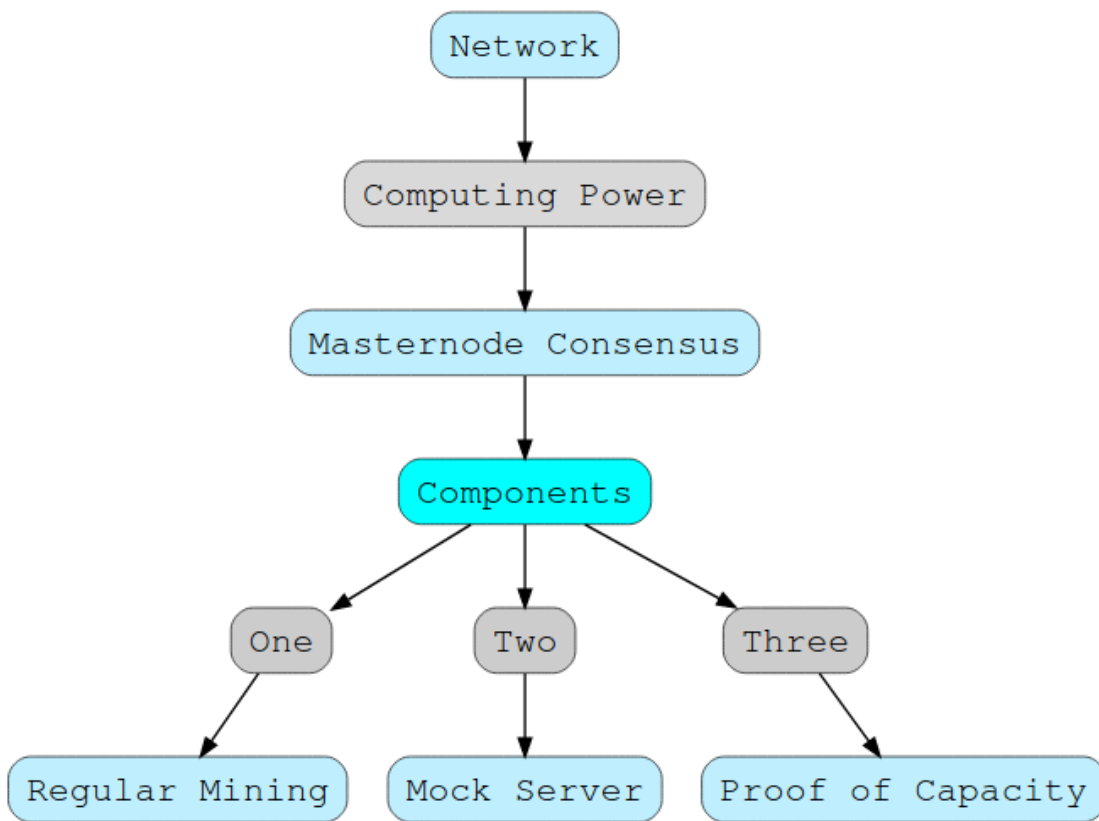
4.1.3 Cryptonight Integration

Part of our integration will be utilizing Cryptonote's Cryptonight algorithm for our PoW hashing. Just like Bytecoin, Monero, and Digibyte, we are aiming to be on the verge of the next big cryptonote based privacy coin. While traditional usage of cryptonote's algorithm works best on CPU cores, we aim to have driver support for all major mining methods, in terms of profitability. This includes servers, GPU mining rigs, and x86 desktop computers. Lonero also plans on having major mining integrations. This will include planned integration with P2Pool as well as further plans to expand configurations for UNOMP, Nanopool, and many Cuda drivers. The allowance of Lonero being integrated with core technologies related to open source mining pools or traditional drivers allows for mass adoption of Lonero as a highly sought after PoW/PoC based cryptocurrency.

In addition to the cryptonight integration, we are utilizing a PoC system for verifying regressional times in our blocks. As mentioned earlier, it is merely traditional hardware proof of work that allows you to develop custom mathematical solutions on your hard drive for transactional verification with minimal hardware. This can be accomplished based on the mathematical proofs we are utilizing for Lonero's cryptocurrency. Therefore, we identify as more of a hybrid form of mining that utilizes cryptonight's hashing system, rather than just pure cryptonight in general.



A crucial technological innovation that makes Lonero superior is our HashBolt masternode consensus. Traditional masternodes allow a form of PoW system for processing transactions under a PoS type solution. They are usually extremely costly on the added premise that you can earn a much higher amount of crypto by operating a masternode. Eventually when the network becomes so congested, a masternode's ROI is less than the current masternode cost. We have decided to change this. We are offering a masternode type consensus for our community for free.



This means that not only are regular users able to mine, but we will allow numerous extensions for people

to run a masternode type system for Lonero. This will grant users with superior hardware specifications or mathematical capabilities to host mining farms, verify signatures more intuitively, and allow both ASIC and Non-ASIC systems to take place in healthy competition.

4.1.4 The Lonero Foundation

The Lonero Foundation is the group behind the further development of Lonero and its community. Our goal as the Lonero Foundation is to drive the prosperous usage of Lonero (LNR) and the technological potential of cryptocurrencies. We aim to ensure that projects built on top of Lonero are beneficial for the industry and plan on making strategic partnerships for continued sustainability and growth. We will always advocate for privacy, security, and social awareness of topics that effect the industry. This includes: “Supporting a free market”, “Decentralized usage of crypto”, “Avoiding market monopolies”, “Privacy in our digital age”, and sustaining our community. We believe a free market society that encourages healthy competition is part of what could make cryptocurrency great.

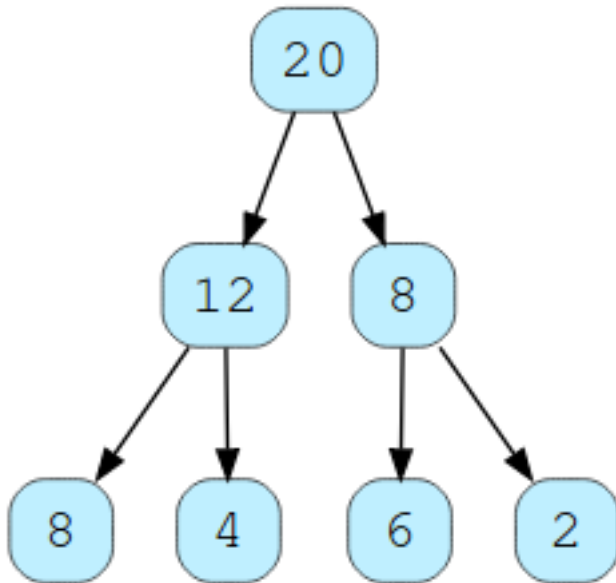
Economically struggling third world countries could greatly benefit from the use of cryptocurrencies and spawn a new era of societal development. The world around us needs a free market society based off of technological advantages, opportunities, and a truly decentralized P2P system. The original goal of Bitcoin was to provide a decentralized, point to point electronic cash system free from centralization by the policies of a single entity. We believe the original Bitcoin Core still holds true to that mission, as well as many other cryptocurrencies in the market today. However, the decentralization aspect can be taken even further. The greatest issue facing other cryptocurrencies in comparison to Bitcoin is the illusion of demand without a fully perceived technological value.

Major companies in essence are using ICOs and the exchange of cryptocurrencies as a monopoly. Projects are raising hundreds of millions and reaching multi-billion-dollar market caps without a finished product or a Minimum Viable Product. This trend needs to end, as it is problematic for the continuation of a sustainable community. The Lonero Foundation also believes that it isn’t the lack of regulation that is hurting the cryptocurrency community, as much as a combination of greed and technological ignorance. With the technologies featured by Lonero, having one of the largest Initial Coin Offerings is very possible, however we find no need to do so. If the team can develop the technology and believe that it can grow purely on the basis of technological value, an ICO is meaningless.

Privacy is an extremely important topic for us. Many people in the technology industry are beginning to advocate that privacy is dead, and a critical problem is the misuse of our personal data. While it is true that people allow this misuse to some extent, many companies still go far with what they use our data for. We at the Lonero foundation aim to never collect the personal data of our users without informing them prior to doing so, as well as never using your personal data for political or monetary gain.

Transparency is important to us, and one of the values that the Lonero Foundation closely adheres to. Without transparency, it is fully difficult for a community to flourish. A proper community always flourishes with trust and a great product. Throughout our mission we aim to be one of the most transparent and community focused teams in the industry.

4.1.5 What are Merkle Trees?



Merkle trees are an important part of the transactional structure of most cryptocurrencies. Lonero's own infrastructure will rely on a merkle tree proof but have layers of complexities integrated in it. This allows for a better integration of its current technologies. Current merkle trees utilize an infrastructure that starts with the root and creates a "leaf and tree node" for transactional hashes. What we do is integrate a merkle proof for cryptographic complexity then utilize a ring signature verification protocol for auditory proof of the transaction. All this happens while we are integrating higher capacity path wall layers for encryption and compressing the block in order to prevent the network from overloading. Since we are utilizing this protocol, we can prevent network congestion.

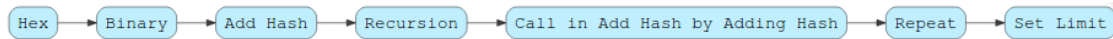
The allowance of transactional nodes, allows us to integrate with multipath layer security and the ring signature protocol much more efficiently. Since we are integrating PoW and PoC mining in our consensus, we can offset the network cache while still allowing for a high method of complexity for the security of Lonero's blockchain. The best way to state it is the following:

1. Start the root: (Merkle Tree's origin for the transaction)
2. Integrate the sub root (Nodes of the transactions)
3. Verify the ring signatures (Before going to the next node)
4. Add a layer of complexity for the hashes upon the next node
5. Repeat for every transaction

4.1.6 Defining Cryptocurrency

A Cryptocurrency is a medium of exchange that integrates hash based algorithms or cryptography for more secure transactions. Most cryptocurrencies utilize peer to peer communication for decentralized networks, and have some sort of proof based algorithm for the feasibility of the network. The most common proofs are: PoW (Proof of Work), PoC (Proof of Capacity/Space), PoS (Proof of Stake), Proof of Signature, and Hybrids. Many cryptocurrencies also utilize an origin block and transactional merkle trees.

4.1.7 Multi-Path Layer Security



The Lonero Foundation plans on integrating multipath layer security into our network. This applies to the ring signature and the transactional nodes. The allowance of multipath layer security will keep our network stable for long term user support and growth. The security protocol is broken into a series of parts. First, one must start with the original transactional hash. Then, once the ring signatures are verified, layers of hashes are added to the complexity and difficulty target. For the next node, the same process repeats recursively. Eventually, the network keeps getting higher in security and transactional nodes are nearly impossible to crack. The better this is integrated by algorithmic means to the network, the more stable future Lonero will have as a cryptocurrency.

Later, this can inspire newer projects or cryptocurrencies, and even way more advanced methods like Quantum encryption can still learn from what has been the past. The allowance of multipath layer security also establishes trust in the network and allows third party vendors the confidence to integrate Lonero with their point of sale systems. Over time, you are establishing a mathematical standard for trust through proof of cryptography for every single new string of nodes created. Just as money used to have a gold standard, cryptocurrency needs to have both a cryptographic and mathematical standard. These topics are highlighted more in our mathematical proofs.

4.1.8 Analyzing Polynomial Time

$2O(\log n) = \text{poly}(n) \rightarrow \text{Karmarkar's theorem}$

$2^{\text{poly}(\log n)} \rightarrow \text{Quasi-polynomial time}$

Karmarkar's theorem is utilized for solving linear problems in polynomial time. This is where statistical analysis will allow for regression. By utilizing Quasi-Polynomial time, it creates complex sequential rings and patterns for the blockchain's ring signature. This allows for the ability to create more mathematical proofs for solving a block. Another primary aspect to recursive functions are patterns that fall under the same patterns as Volterra integro-differential equations. This is especially true with sequential data, in which having hybrid mining algorithms typically utilizes a piecewise polynomial within its logic. Since you are analyzing polynomial regression, you can create algorithmic extensions to solving certain aspects of the blockchain such as computational benchmarks and predictive analysis for having a better luck chance of solving the block.

4.1.9 Merkle Mathematical Proofs

The Merkle tree proof that we are utilizing is the integration of the ring signature verification, multipath layer security and the utilization of polynomial time. The sequential rings and patterns integrated on the blockchain allow for more efficient hashing and regression for further nodes in the transactional roots. Because of the mathematical proof we have, people can create their own custom mining algorithms to go along with what will be readily available in the market. Later, depending on community support, we may also create open source tools and developer SDKs centered around algorithmic capabilities and PoC mining for end users.

4.1.10 The Ring Signature Protocol

We utilize cryptonote's ring signature protocol for verification of transactional PoW. Traditional methods utilize a public and private key integration for verifying the original signature of a transaction. However, ring signatures add an extra layer of complexity and security. They utilize several public and private keys for the shared transactional hash to verify the block. This means instead of just one public to private key

verification, you are further increasing security by adding more signatures to your transactions. Utilizing this group verification method is known as “Ring Signatures” and is similar to the original signature method but integrates multiple signatures for verification. Since you are using multiple signatures, the original transaction root still knows the original public and private signers once the verification for a transaction hash is verified.

In terms of using large scale systems, ring signatures allows for the transactions to be eventually untraceable and a more secure form of exchange. However, methods can still be implemented to prevent abuse of the system. Having, a ring signature type protocol allows for a more open market, and when utilized correctly, you can still have a network with high speeds. Compliance is still possible with advanced forms of exchange and allowing a technological implication of these protocols can make actions such as cross border transactions faster, safer, and significantly more secure.

4.1.11 Problem with Current Mining

Traditional mining systems are extremely outdated and inefficient. They require higher capacity hardware for greater profitability, which leads to the opportunity for market monopolies. Market monopolies can hence be created for mining hardware, GPU systems, or even centralized towards people who have data centers or mining farms. While, it is true that people with greater hardware capacity may profit more using traditional mining on Lonero’s systems, there are still ways around it. Creating the ability for “Proof of Capacity” mining solutions, (as talked about earlier), allows for people to solve the blockchain utilizing their mathematical capabilities and understanding. Now, not only wealth is rewarded, but mathematical knowledge.

4.1.12 Applying Euclidean Geometry

We have plans to integrate computer vision technologies later with Lonero’s core technological infrastructure. Part of this will be Euclidean geometry and an A.I. that can integrate with merkle tree proofs. This will be integrated in a series of different areas such as the DAO creation, compression on the blockchain, and sequential pattern recognition amongst the network. Imagine a merkle tree being geometrically analyzed and manipulated as is a data structure. Obviously, this will be a long stretch goal in terms of integrating with Lonero’s core development and will be laid out for the community over time.

For more info read [Beyond Blockchain: The Futurist Guide to Innovative Technologies](#) or visit the [Lonero website](#).

[Everipedia](#) | [Announcement](#)

4.2 Protocol

4.2.1 About

The protocols that make Lonero what it is also allow for expanding use cases for Lonero across a variety of different sectors.

Among those usecases includes:

- Crypto Mining
- IOT and “Smart Devices”
- Grid Computing Networks
- Software Defined Networking

- Parallel Computing Architectures

4.2.2 Crypto Mining

Current cryptocurrency mining is limited to special types of hardware. However, Lonero is working on leveling the playing field. More details about this is shown under the [crypto section](#) or our whitepaper.

4.2.3 IOT and “Smart Devices”

Lonero’s [API](#) is extensively built to allow smart device and IOT compatibility with its [decentralized-internet SDK](#) and pipeline.

4.2.4 Grid Computing Networks

Outside of working great with software such as [PM2](#), at the core our decentralized internet is built with grid computing in mind. Infact, you can build custom DAPPS with grid computing compatibility or shared computing.

4.2.5 Software Defined Networking

Software defined networking and dynamic network centered applications or configurations work great with Lonero’s decentralized internet, or even connecting to our blockchain. Projects utilizing SDN demos of Lonero have even included [underwater wireless networks](#).

4.2.6 Parallel Computing Architectures

Running and syncing data in parallel or multi-threaded processes, as well as parallel processing implementation is a huge part of our network’s overall design.

4.3 Governance

This is under construction

4.3.1 Abstract

We will present technological methodologies and perspectives related to a modern day socioeconomic system proposal. The basis of this system will be centered around anarcho-capitalistic concepts, decentralization, and deontological ethical premises. This framework will mention technologies such as the [decentralized-internet SDK](#), [UDIs](#), and technological case studies. This is primarily a computational engineering and blockchain-centric proposal for new methodologies that change the way current socioeconomic structures function in society. The case for a tax-free technologically advanced society backed by blockchain, and the potential politech implications of decentralization through mathematical and logical methods will also be presented. This proposal aims to provide insight to the world of blockchain, algorithms, ethics, politics, sociophysics, economics, and financial economic theory.

4.3.2 Introduction

The **Lonero** project integrates its own cryptocurrency, as well as a series of open hardware and collaborative efforts, in order to create its own framework for a decentralized pipeline of systems. **Lonero's** crypto **LNR**, has a technological integration for a series of different proofs that is compatible with ASIC and non-ASIC systems. Its mining validation system utilizes both "Proof of Work" and "Proof of Space". The way **Lonero's** systems can be integrated in place goes beyond mining, distributed computing or shared networks. One can further apply concepts of governance and disruptive socioeconomic models.

Purpose

Lonero's technology is centered around decentralization and decentralized use-cases, including its very own **decentralized-internet SDK**. In current modern times, in regards to socioeconomic systems and the political state of things, politech startups are looking for massive disruptive societal models and philosophies. The current allusion of choice in regards to a two party system or currently existing systems out there is quite noticeable. One perspective has been to look into the adaptation of anarcho-capitalist types systems and change through the means of pure logic and a science. This is a huge component in regards to something **Lonero** wants to work on.

4.4 Architecture

4.4.1 About

The technological architecture for **Lonero** is primarily focused on being compatible with most modern day computational systems.

4.5 Make a Mining Pool

4.5.1 The Deps.

```
sudo apt-get install aptitude
sudo aptitude update
sudo aptitude install -with-recommends build-essential autotools-dev
↪autoconf automake
libcurl3 libcurl4-gnutls-dev git make cmake libssl-dev pkg-config libevent-
↪dev libunbound-dev
libminiupnpc-dev doxygen supervisor jq libboost-all-dev htop
apt-get install build-essential libtool autotools-dev autoconf pkg-config
↪libssl-dev
apt-get install libboost-all-dev git npm nodejs nodejs-legacy libminiupnpc-
↪dev redis-server
add-apt-repository ppa:bitcoin/bitcoin
apt-get update
apt-get install libdb4.8-dev libdb4.8++-dev
curl -sL https://raw.githubusercontent.com/creationix/nvm/v0.31.0/install.sh
↪-o install_nvm.sh
bash install_nvm.sh
source ~/.profile
nvm install 0.10.48
nvm use 0.10.48
```

(continues on next page)

(continued from previous page)

```
nvm alias default 0.10.48
nvm use default
sudo apt-get update
sudo apt-get install apache2
sudo ufw app lis
sudo ufw allow 'Apache Full'
```

4.5.2 Install Beta

```
git clone https://github.com/Lonero-Team/Lonero-Beta.git
cd Lonero-Beta
```

4.5.3 Start Daemon

```
./forknoted --config-file configs/lonero.conf
```

4.5.4 Start the Wallet

```
./simplewallet --config-file configs/lonero.conf
```

4.5.5 Add the Software

```
git clone https://github.com/forknote/cryptonote-universal-pool.git pool
cd pool
npm update
```

4.5.6 Configure the Pool

```
cp config_example.json config.json
```

Change ports to 34414 and 34415

If your IP is static, change the host to your static IP.

4.5.7 Start the Mining Pool

```
run node init.js
```

4.5.8 Host the Pool's Front End

Copy files from the website to html directory:

```
sudo cp -rf admin.html config.js custom.css custom.js index.html pages/
↳ themes/ /var/www/html
```

4.5.9 Customize the Mining Pool

Update your pool's IP in the configs:

```
cd /var/www/html
```

```
sudo nano config.js
```

Change the name of the pool:

```
sudo nano index.html
```

You can also update the CSS and JS files for styling and functionality as needed.

Special thanks to:

The [Lonero Website](#)

The [CoinWiki Website](#)

[Cryptonote Universal Pool](#)

4.6 OS



Lonero OS is based off of the Lubuntu OS, and has a few adding things such as:

1. The uncompiled source binaries of: GitPrep, SMesh, Kademlia, Beaker Browser, and VPNBook
2. Telegram Desktop installed
3. DuckDuckGo should be the default browser

This project is part of an aim to create an open, distributed, censorship resistant operating system, and this is the beta of that.

4.6.1 Features

- Support freedom and decentralization
- Has pre-installed source codes of P2P modules
- Already has bluetooth, and a bit-torrent transmission client (as does Lubuntu)
- An operating system easy to run decentralized blockchain applications on
- The same software modules used here is the same operating system for “ChainBoard” which is a hardware board used by startups such as Stark Drones Corporation

Install via Docker: `docker pull gamer456148/lonero-os`

Pull Source w/ SourceForge:

```
git clone https://git.code.sf.net/p/lonero-os/code lonero-os-code
```

4.7 LonScript

- Utilizing ANTLR Parsing
- Allows for Multi-State Machinery
- Truly built for Scientific Computing

and so much more...



4.7.1 Lonero CLI

The Lonero CLI is currently under construction, but once it is build should integrate all its core features.

Install via: `npm install lonero-cli`

Commands list:

```
lonero-cli decent-test [FILE]
lonero-cli dev [FILE]
lonero-cli hello [FILE]
lonero-cli help [COMMAND]
lonero-cli inquirer
lonero-cli lon [FILE]
lonero-cli lon-matrix [FILE]
lonero-cli lon-multi [FILE]
lonero-cli plugin [FILE]
lonero-cli produce [FILE]
lonero-cli stage [FILE]
lonero-cli test-compile [FILE]
```

4.7.2 Lonero IDE

The Lonero IDE is also actively being built.

Install via `npm install lonero-ide`

This is the official IDE for Lonero. In order to start, make sure you have Yarn installed and NVM as well as Node version 10. If cloning from the GitHub repo, make sure to install all the dependencies. You don't need to run the `npm install` and `yarn build` command if directly installing from NPM.

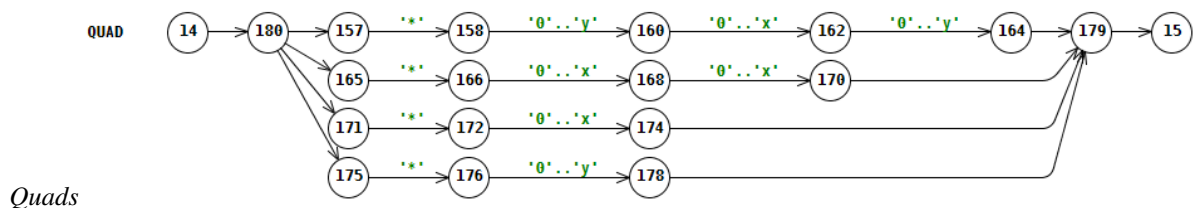
Run the following:

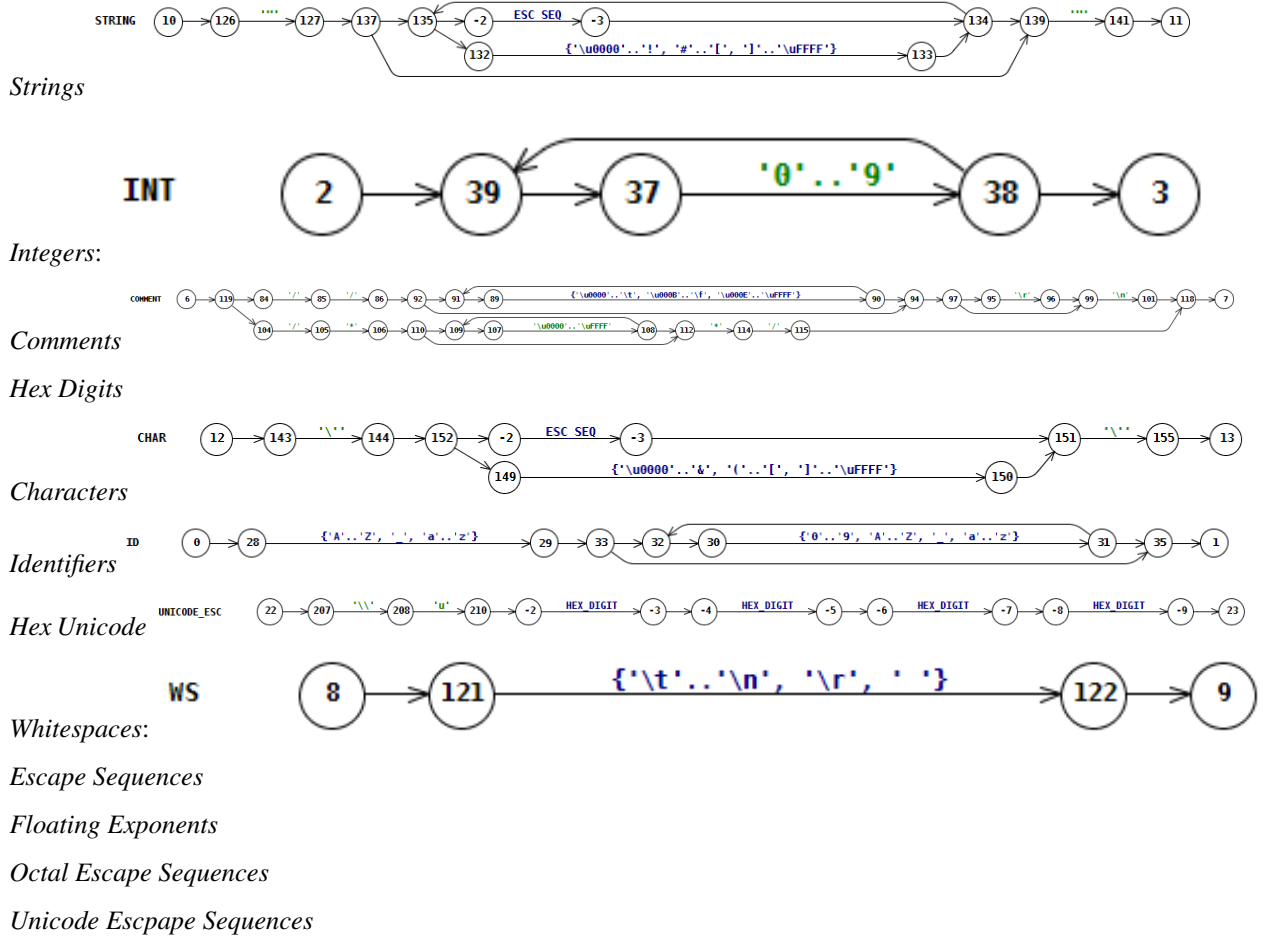
```
$ nvm install 10
$ nvm use 10
$ npm install
$ yarn build
$ yarn theia start
```

In order to shut down, run: `kill $(lsof -t -i:3000)`

The default port this uses is 3000.

4.7.3 Syntax Charts





4.8 Documentation

||||

4.8.1 Critical Components

- Tools and Protocols
- Distributed Computing
- Decentralized Web Model

4.8.2 Basic Usage Introduction

This is the same usage guide that is available in [LeanPub](#) and [other](#) places.

One way to get started is by installing the SDK. Let us start by running:

```
npm install decentralized-internet
```

Now let us:

```
$ cd node_modules
$ cd decentralized-internet
$ ls
```

Now, multiple folders are setup for routing, but the easiest folder to cd into for looking at the components is add-on so:

```
$ cd add-on
$ ls
```

Now you can see there are four main components to this project:

- CluserPost
- GridBee
- LNRChain
- Reinvent the Internet

Those are the core folder/component names of what we will be looking at.

What is Clusterpost?

[Clusterpost](#) is a grid computing tool that allows you to offload processes to a CouchDB database. However, for this SDK the migration is suggesting to use [BigChainDB](#) over CouchDB on the basis of decentralization opportunities. Clusterpost is also utilized for remotely scheduling and executing computational processes.

Here you can look at the code so:

```
$ cd clusterpost
$ chmod 777 ./setUpDevelopment.sh
$ ./setUpDevelopment.sh
$ sudo nano conf.production.json
```

Let us start with looking at the following configs in the conf.production.json file

```
{
  "host": "localhost",
  "port": 8180,
  "plugins": {
    "vision": {},
    "inert": {},
    "lout": {},
    "h2o2": {},
    "hapi-auth-jwt": {},
    "clusterpost-auth": {
      "privateKey": "someRandomKey",
      "saltRounds": 10,
      "algorithm": {
        "algorithm": "HS256"
      },
      "algorithms": {
        "algorithms": [ "HS256" ]
      }
    }
  },
}
```

Here, you see the default port it should run on is 8180, you are welcome to change that if needed or open the port. Also you see the plugins or default NPM packages the production environment uses and the default algorithm for clusterpost's authentication.

Next let us change the credentials as needed:

```
},
    "mailer": {
        "nodemailer": "nodemailer-stub-transport",
        "from": "clusterpost server <insertemail@here.com>",
        "message": "Hello @USERNAME@,<br>Somebody asked me to send_
→you a link to reset your password, hopefully it was you.<br>Follow this <a_
→href='@SERVER@/public/#/login/reset?token=@TOKEN@'>link</a> to reset your_
→password.<br>The link will expire in 30 minutes.<br>Bye."
    },
    "userdb" : {
        "hostname": "http://localhost:5984",
        "database": "clusterjobs"
    }
},
```

You need to setup the login either through nodemailer or the database program you are using and change the credentials so that clusterpost will be able to authenticate into the database.

Next while running the database you could run some tests, so:

```
$ cd test
$ nautilus .
```

You are free to look into the configs and test code as needed, but you can get started testing the default setup right away running: `npm run` or `node filename.js`

A good example would be

```
$ cd test
$ node createNewJob.js
```

The GridBee Framework

Now, let us take a look at the GridBee Framework, which is an open source library that allows web browsers to act as clients and communicate with BOINC. It is coded in JavaScript and the HaXe programming language.

Make sure to download [HaXe](#) and [FlashDevelop](#). FlashDevelop is an IDE but is needed to compile the project file GridBee.hxproj.

If you are still in the clusterpost test folder then here is what to run in the terminal:

```
$ cd ../
$ cd ../
$ cd gridbee-framework-old
$ cd GridBee
$ nautilus .
```

Now you can see the project file GridBee.hxproj, right click it to open w/ FlashDevelop and press F8 to build.

You can also merge local storage with the BigChainDB that clusterpost is utilizing, and http requests to offload processes from your application and start optimally building a pipeline, but this is beyond the scope of this introduction.

Let's Look at LNRChain

The LNRChain folder includes tendermint, and the basic app.js demo file, and is meant to be utilized for:

- 1) Part of a pipeline where you can connect Tendermint to BigChainDB *OR*
- 2) A [sidechain](#) for the SDK and [LNRBeta](#) and [Bitcoin](#).

To access LNRChain, if you are still in the GridBee folder:

```
$ cd ../
$ cd ../
$ cd LNRChain
$ sudo chmod 777 tendermint
$ sudo nano app.js
```

Now you have just opened the app.js sample in the terminal, you should see:

```
// app.js
let lotion = require('lotion')

let app = lotion({
  initialState: {
    count: 0
  }
})

function transactionHandler(state, transaction) {
  if (state.count === transaction.nonce) {
    state.count++
  }
}

let connect = require('lotion-connect')
app.use(transactionHandler)

app.start().then(appInfo => console.log(appInfo.GCI))
```

This is Lotion's multi-state sample file which allows you to run multiple states for the blockchain project you want to setup. Lotion is the npm module powered by the Tendermint consensus for you to make your own blockchain apps.

Component #4: Reinvent the Net

Now is time for the final core component of this SDK. Let us start by accessing the source files:

```
$ cd ../
$ cd Reinvent-the-Internet
$ unzip The APIs & Shell.zip
$ nautilus .
```

Let us open the MSP430G2001.ccxml file with a text editor, i.e. "right click":

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<configurations XML_version="1.2" id="configurations_0">
  <configuration XML_version="1.2" id="configuration_0">
    <instance XML_version="1.2" desc="TI MSP430 USB1" href="connections/
    ↪TIMSP430-USB.xml" id="TI MSP430 USB1" xml="TIMSP430-USB.xml" xmlpath=
    ↪"connections"/>
    <connection XML_version="1.2" id="TI MSP430 USB1">
      <instance XML_version="1.2" href="drivers/msp430_emu.xml" id=
      ↪"drivers" xml="msp430_emu.xml" xmlpath="drivers"/>
      <platform XML_version="1.2" id="platform_0">
```

(continues on next page)

(continued from previous page)

```

        <instance XML_version="1.2" desc="MSP430G2001" href="devices/
↪MSP430G2001.xml" id="MSP430G2001" xml="MSP430G2001.xml" xmlpath="devices"/>
        </platform>
    </connection>
</configuration>
</configurations>

```

The fourth module (as one can tell), is hardware-oriented. The XML config shows that the driver is for a Texas Instruments dev board. This also means that the IDE that the fourth component was developed on was likely [CCS Cloud](#).

Now you may be confused at this point as to what was the point of this module or what is it even for?

This part of the project was actually utilized for a hardware project in which somebody garnished underwater wireless signals through sonar conversion. However, similar use-cases can be done for anybody who wants to have a software defined network. This is your chance to expand the pipeline through low level hardware or code integrations and an optional part of the SDK.

To summarize we are:

- 1) Offloading data
- 2) Communicating Data
- 3) Building our Blockchain
- 4) Building our own SDN and wireless protocol

This is why things have been setup the way they are.

4.9 Hardware Cryptography

Our [decentralized-internet SDK](#) and lots of the technologies that [Lonero](#) works on can be utilized in hardware cryptography.

- Headless Computing Systems
- Hardware Cryptography as a Service
- Post Quantum Cryptographic Research
- Enterprise Level Cryptography Networks

4.9.1 Headless Computing Systems

Designs for headless computers and cryptographic hardware keys can be implemented with our SDK as well as our network. Mutli-Path Layered hardware cryptography and a flashed cryptographic hardware key network is also feasible. We even have a key design that is built untop of the [Lonero OS](#) through firmware connectivity.



4.9.2 Hardware Cryptography as a Service

Another phase of service offerings can include hardware cryptography as a service or even [Wireshark](#) integrated or network cache monitoring solutions. This can work on both decentralized and traditional networks.

4.9.3 Post Quantum Cryptographic Research

We are doing research in a variety of different Qubit compatible cryptography solutions. We aim to lay foundational work for the era of cryptography in a post-classical computing world.

4.9.4 Enterprise Level Cryptography Networks

Our cryptographic solutions aim to be compatible and fully scalable for large scale enterprise systems.

4.10 Main Installation Methods

```
Sketch Plugin: wget https://git.io/Jv2pk
Use Julia: Pkg.add("DecentralizedInternet")
Install via NPM: npm i decentralized-internet
Install via DUB: dub add decentralized-internet
Install via YARN: yarn add decentralized-internet
Install via PIP: pip install decentralized-internet
Install via APM: apm install decentralized-internet
Install via GEM: gem install decentralized-internet
Install via PNPM: pnpm install decentralized-internet
Install via CNPM: cnpm install decentralized-internet
Ember Installation: ember install decentralized-internet
Install via Spack: ./spack install decentralized-internet
```

(continues on next page)

(continued from previous page)

```
Raco Installation: raco pkg install decentralized-internet
Install w/ Meteor: meteor add startup:decentralized-internet
Install via VS Code: ext install Lonero.decentralized-internet
Other Meteor Method: meteor npm install decentralized-internet
Install through Leiningen/Boot: [decentralized-internet "0.1.0"]
Install via SNAP: sudo snap install decentralized-internet --edge
Install via Bower: bower install Lonero-Team/Decentralized-Internet
Docker Installation: docker pull gamer456148/decentralized-internet
Use Clojure CLI/deps.edn: decentralized-internet {:mvn/version "0.1.0"}
Use Gradle:Compile 'decentralized-internet:decentralized-internet:0.1.0'
SourceForge: git clone git://git.code.sf.net/p/decentralized-internet/git
Use wget: sudo wget -O decentralized-internet.tar.gz "https://git.io/JvR7b"
```

Sysget Users

```
sysget install decentralized-internet
```

Pick either option: 4, 14, 15, 18 or 20

Using Maven

```
<dependency>
  <groupId>decentralized-internet</groupId>
  <artifactId>decentralized-internet</artifactId>
  <version>0.1.0</version>
</dependency>
```

Install via Dart

Add to your pubspec.yaml file:

```
dependencies:
  decentralized_internet: ^3.4.1
```

Run: pub get

Install via Cordova

```
cordova plugin add https://github.com/Lonero-Team/Decentralized-Internet.git
```

Or you can add via search

```
cordova plugin searchcordova-plugin-decentralized-internet``
```

Or via name instead of repo

```
cordova plugin add cordova-plugin-decentralized-internet
```

Export Components

```
bit export decentralized-internet.lonero_decentralized-internet
```

Use this Module via Puppet

Add this to your Puppetfile as a declaration:

```
mod 'gamer456148-decentralized_internet', '5.2.1'
```

Next run the command:

```
bolt puppetfile install
```

Instead of the above, you can also try:

```
puppet module install gamer456148-decentralized_internet --version 5.2.1
```

This mod line is for those who use r10k or Code Manager

Learn more [here](#)

Arch Linux Installation Method

```
git clone https://aur.archlinux.org/snapd.git
cd snapd
makepkg -si
sudo systemctl enable --now snapd.socket
sudo ln -s /var/lib/snapd/snap /snap
sudo snap install decentralized-internet --edge
```

4.10.1 For Installing on DigitalOcean

First Step: In order to install via one click app, click the “Create Droplet Button” via the [Marketplace](#) page.

Second Step: In regards to usage, I recommend reading our SDK’s generic usage guide [that is here](#). The same instructions apply to droplets.

Please keep in mind you can skip the `npm install`, as the node modules should already be pre-installed in the droplet but not the below configurations.

To access the program in root use `cd node_modules/decentralized-internet`. The dev files should be in `cd addon`. You can then install the below configurations in root, and proceed with the core component development.

Configurations:

For ease of simplicity, I recommend installing the [Slap IDE](#) over VIM or GNU Nano.

```
curl -sL https://raw.githubusercontent.com/slap-editor/slap/master/install.  
↪sh | sh
```

Also keep in mind if you decide to use slap, it utilizes Node.js so make sure you have the latest version installed:

```
sudo npm install -g slap@latest
```

You can now run the Slap commands seen [at their repo](#), for editing code files. For HaXe, I recommend you do the default PPA Ubuntu Installation:

```
sudo add-apt-repository ppa:haxe/releases -y  
sudo apt-get update  
sudo apt-get install haxe -y  
mkdir ~/haxelib && haxelib setup ~/haxelib
```

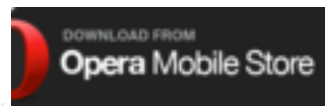
Or run the Debain installation:

```
sudo apt-get install haxe -y  
mkdir ~/haxelib && haxelib setup ~/haxelib
```

The rest of the configuration instructions in regards to Debian, can be seen [at the HaXe site](#).

After you install Slap and Haxe on the Droplet, I recommend you use Slap for editing needed code files, and the `haxelib` commands in the terminal in regards to HaXe code files.

This project was created in order to support a new internet. One that is more open, free, and censorship-resistant in comparison to the old internet. An internet that eventually wouldn’t need to rely on telecom towers, an outdated grid, or all these other “old school” forms of tech. We believe P2P compatibility is an important part of the future of the net. Grid Computing also plays a role in having a better means of transferring information in a speedy, more cost-efficient and reliable manner.



GetJar Soft32 GitLab Aptoide Softpile TideLift AppAgg Apptoko GitHub App Stackshare AlternativeTo Software Informer

For citing this software:

Kamal, A. M. decentralized-internet. npm (2020). Available at: <https://www.npmjs.com/package/decentralized-internet>. (Accessed: 30th September 2020)

4.11 Up-to-Date Related Academic Work

4.11.1 The Journal of Decentralized and Grid Computing

This is a journal to publish your decentralized, parallel processing, distributed and grid computing papers in support of research that contributes to advancement of the field.

[Paper 1](#) | [Paper 2](#) | [Paper 3](#)

Journal Features

- Add a DOI to your paper
- Quick turn around time for review
- Increase impact factor when others cite your work
- Integrate code, latex, or markdown for journal entries
- Just went live and targeted towards decentralization and distributed computing

Website: <https://authorea.com/inst/20499>

4.12 Lonero ATM Project



The **Lonero ATM Project** is a fork of [openbitcoinatm](#), but for the concept of Lonero. Part of the reason this was made was seeing projects like [Bitstop](#), inspired us to want to create something open source that would make Lonero more accessible. Please keep in mind this is intended for educational usecases and aspects only. Please keep in mind regulatory laws, and that ATM or crypto businesses may need a [Money Transmitter License](#). Also please keep in mind this crypto is a utility, and this repo as well as community hardware project is primarily intended for educational purposes. To quote [Jay Clayton](#) who was an SEC chair, “*Cryptocurrencies are replacements for sovereign currencies. ... [they] replace the yen, the dollar, the euro with bitcoin. That type of currency is not a security*”. Us at Lonero believes the same applies at our case. Lonero was released as open source and without an ICO. We are actively working on many technological aspects that make [Lonero](#) unique in hopes of mass adoption.

4.12.1 Materials

- 5V Power Supply
 - Resistor and Leads
 - Arduino SD Card Shield
 - Arduino Uno ATmega328 or Rev3
 - Apex 3000 Cash Acceptor or (3400)
 - Mounting Squares, Velcro, and Zip ties
- Make sure to have the required tools and a knowledge of soldering

Get Kit w/ Just the Materials (No Case): \$450

See Terms Below

Please allow 10 to 15 extra days for shipping

4.12.2 The Diagrams

4.12.3 Terms of Service

Please keep in mind these instructions are on an *AS IS* basis. Proper knowledge of soldering and basic IT is recommended. This may take at least few hours of building and setup time. This will likely take more time if you're constructing your own case. Please keep in mind that many states have money transmitter license rules, and certain regulations ATM businesses need to adhere to. We are not to be held liable or responsible for any usage, misuse, or any liabilities related to this project, ideation, or the contents of this repo. The same applies for other related Lonero projects or software. Also, be sure to follow proper safety guidelines when going about constructing a project. We are not responsible for any liabilities related to the construction process, or any other known or unknown liabilities. Make sure to do your own research.

If you are interested in implementing something outside of **OpenATM**, perhaps check the [Lamassu configs](#) or look into the [Open Franchise Model](#).

4.13 Kinetic Energy Miner



4.14 PeerTop | Lonero OS Laptops



Small Teaser, *details coming soon

4.15 ChainBoard | The Next Gen Wireless Dev Board



4.15.1 About

ChainBoard is the next big thing for hardware enthusiasts, tinkerers and makers wanting to go “decentralized”.

Having worked on an Underwater WiFi project before, I have had the experience of utilizing multiple microcontrollers in order to achieve what I wanted. Having a bunch of pieces of hardware wired together seemed inefficient, and I looked at what the main components I needed were. This led me to make ChainBoard. ChainBoard is meant to be a high end microcontroller and personal computer that has many components needed to build complex LoRaWAN, bluetooth, wireless and sensory projects. You essentially have what is needed for low-energy bluetooth, mesh networking, LoRa transmission, UART-based communication modules, and many more all into a single board.

*Some of the items that were in the above table included the below main components:

- 32-bit MCUs
- 915MHZ Lora Transceiver
- 928MHZ Lora+Bluetooth Module
- Bluetooth 4.2 BLE Module
- IEEE 802.11 b/g/n Low Energy Bluetooth Module
- USB to UART Bridge Controller

Not only can you use ChainBoard for the project ideas I demonstrated, but there are ways it can be utilized for much more. For example, one can create a smart plug or smart home system with ChainBoard. One can be able to have ChainBoard act as the mainframe for a small mesh network with multiple low cost bluetooth devices. One can create their own LoRaWAN transmission client. Somebody can even make a microcontroller controlled generator type system or utilize ChainBoard for an interconnected security system.

Given the capabilities of the components, and its cost, we didn’t want ChainBoard to be minimal in its applicable use cases. We wanted to create something that people may start building untop of. We wanted components that can be utilized for decentralization or P2P style connectivity. This may seem like just a extravagant project board, but we envision something much bigger. However, in order to make history, we feel we need to target the hobbyists first.

