Lightweight full verification with Lightning Nodes

How Utreexo helps Lightning Users

https://github.com/kcalvinalvin/bitdevs-sg-09-2024

Lightweight full verification

Lightweight **full verification**

https://github.com/kcalvinalvin/bitdevs-sg-09-2024

What is full verification?

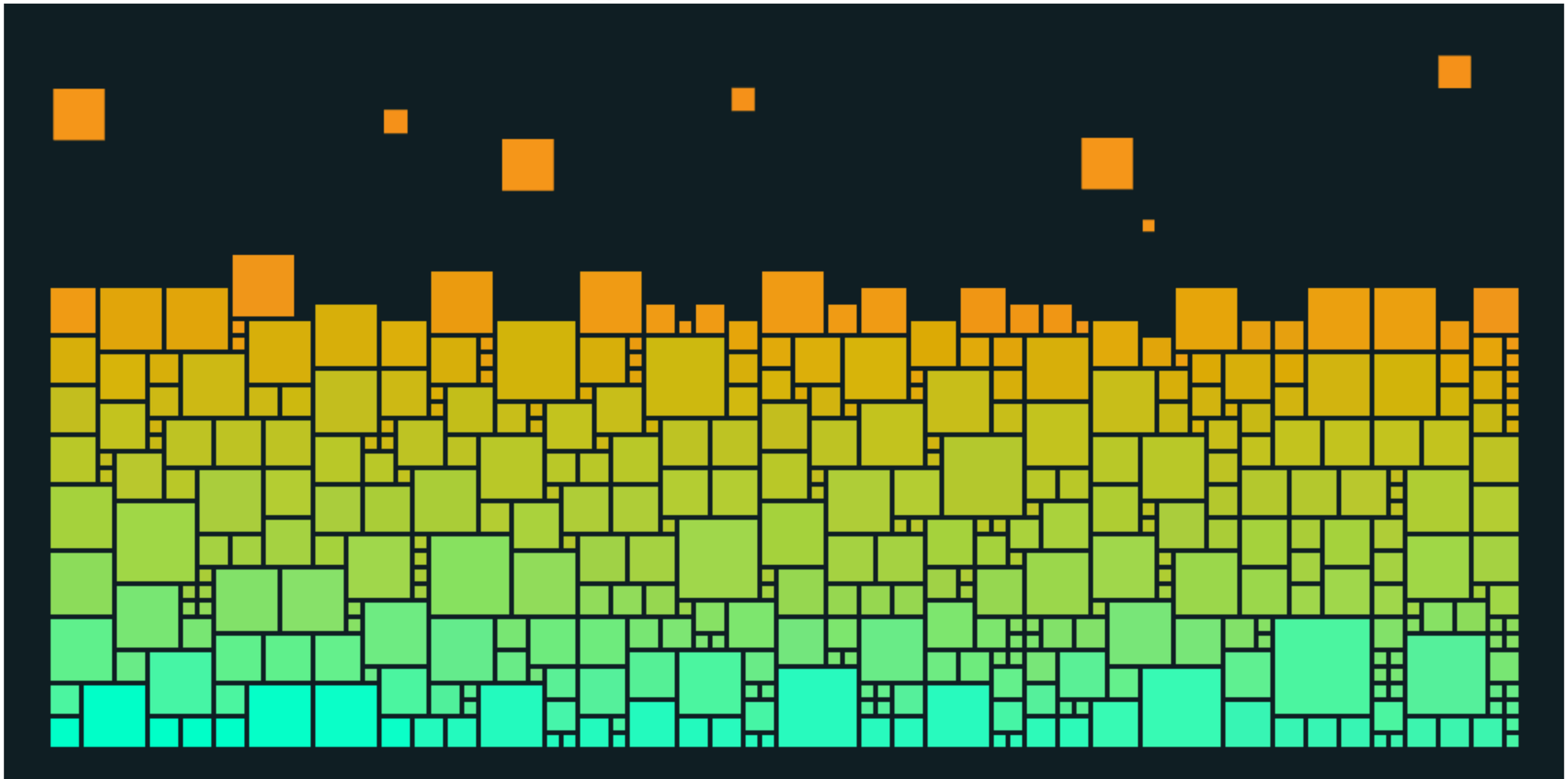
For Lightning nodes

What full verification is

- Ensure that the peer I have a channel with doesn't publish an old state

Loss of funds!

If a lightning node fails to do verification



Scan for any txs that try to scam me

Things need to scan the mempool

For lightning nodes

1. Must be synced to the tip of the blockchain

Things need to scan the mempool

For lightning nodes

1. Must be synced to the tip of the blockchain
2. Must be aware of all the txs in the mempool

These two requirements are handled by the bitcoin full node

Good things about full verification

The pros

Good things about full verification

The pros

- Excellent security

Not so good things about full verification

The cons

Not so good things about full verification

The cons

- More compute resources

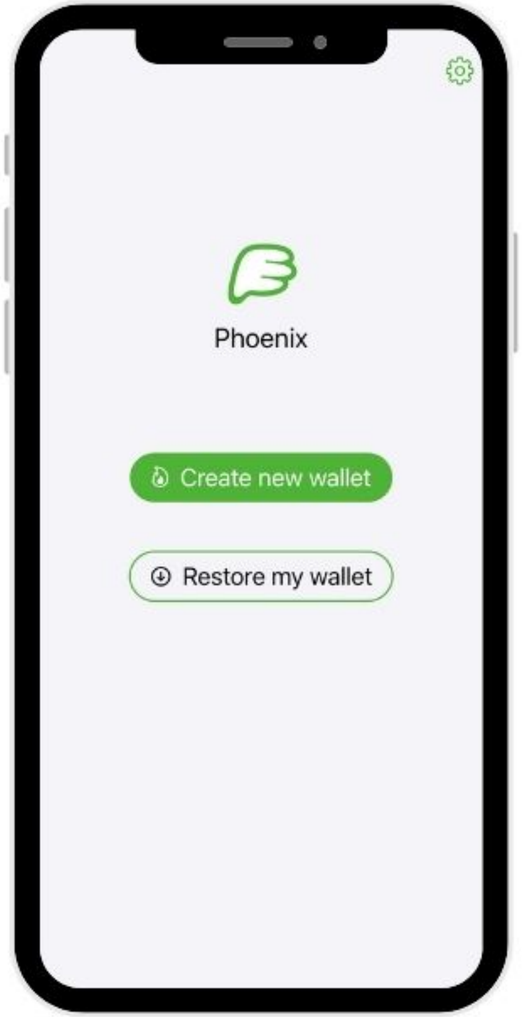
Not so good things about full verification

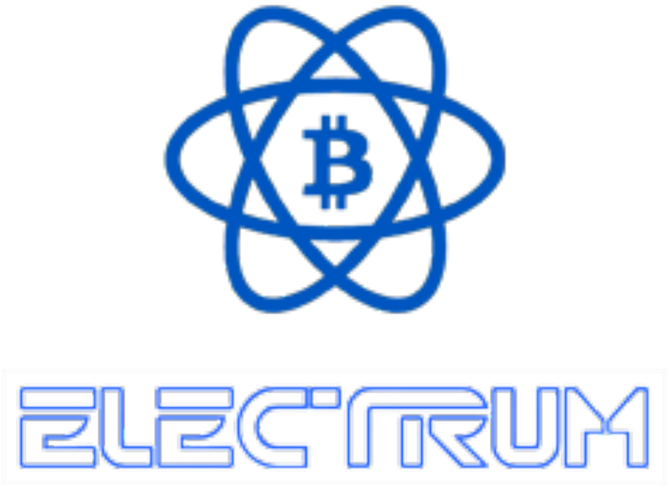
The cons

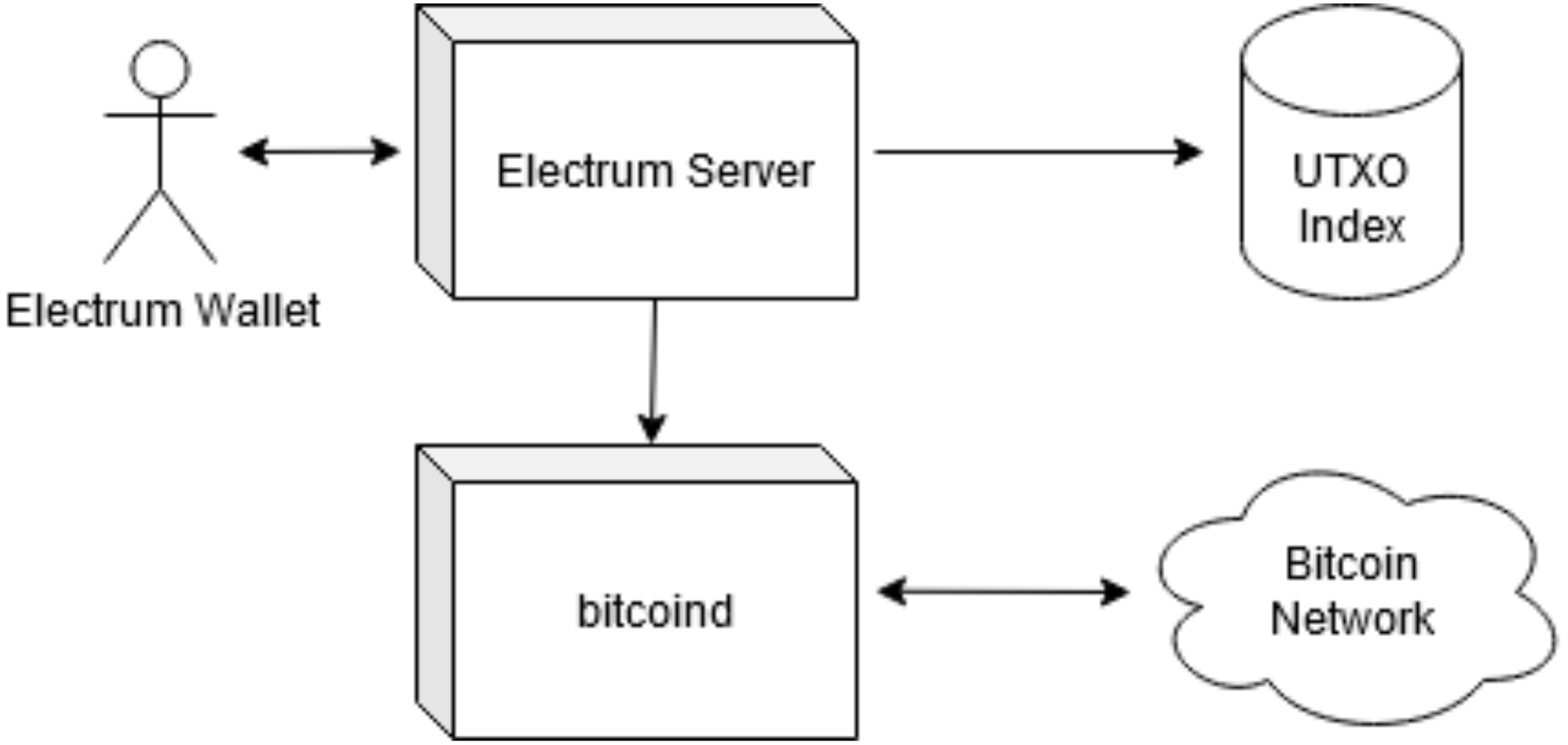
- More compute resources
- Difficult to setup

Lightweight full verification

Phoenix Lightning Wallet




ELECTRUM


Electrum Wallet
Electrum Server
UTXO
Index
bitcoind
Bitcoin
Network

Good things about being lightweight

The pros

- Little compute resources required

Not so good things about being lightweight

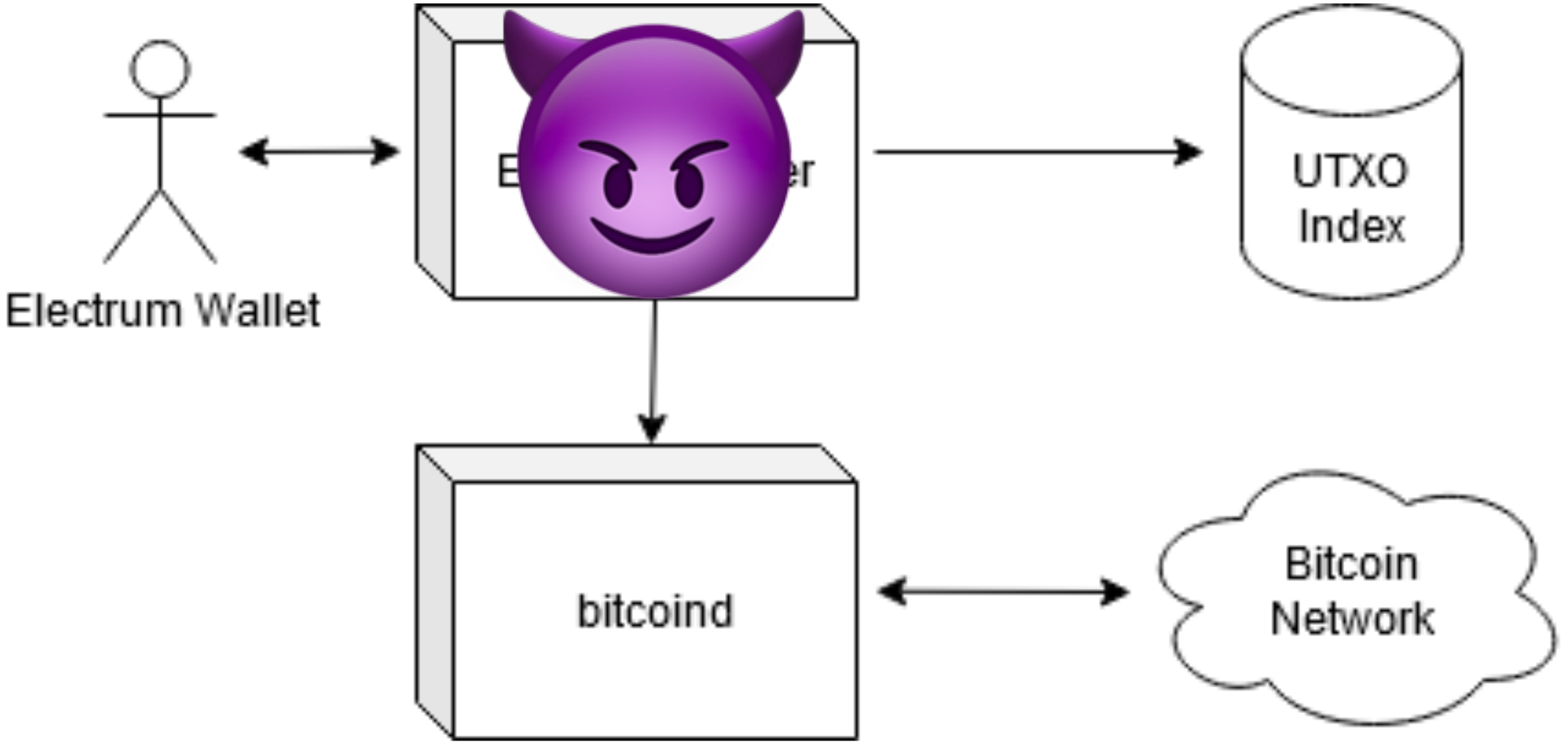
The cons

- Easy for the server to steal my money

Things need to scan the mempool

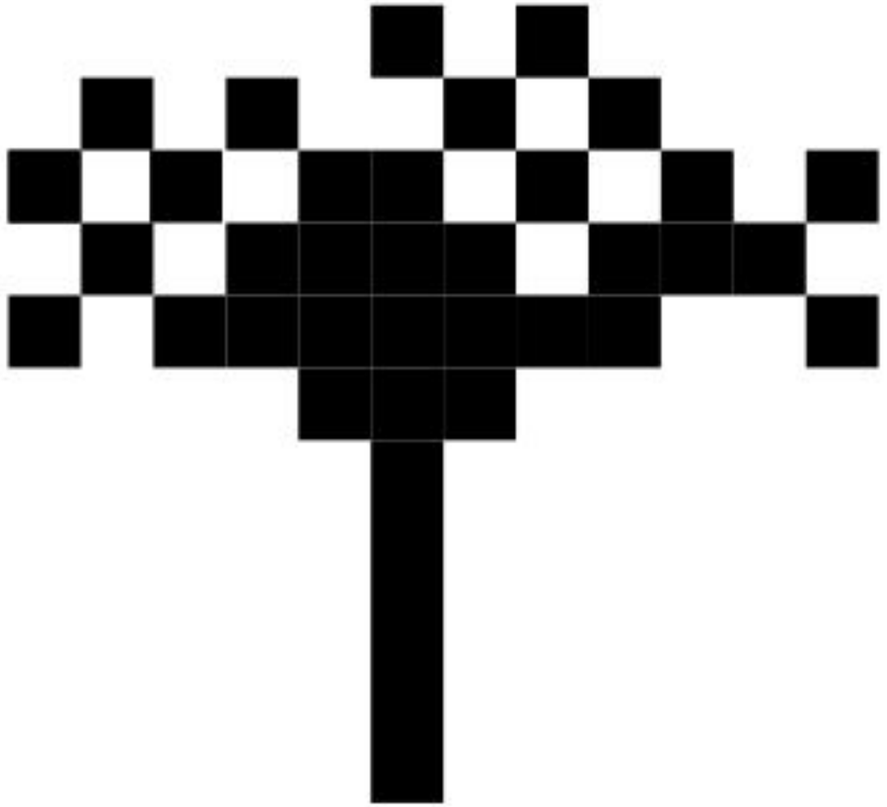
For lightning nodes

1. Must be synced to the tip of the blockchain
2. Must be aware of all the txs in the mempool


Electrum Wallet
UTXO
Index
bitcoind
Bitcoin
Network

Server can purposely omit txs!

Lightweight full verification



Utreexo node

The Pros

- Tiny in size

Utreexo node

The Pros

- Tiny in size
- Performant

What's the catch?

Utreexo node

The Cons

- Not as battle tested

Utreexo node

The Cons

- Not as battle tested
- More compute needed vs electrum

Middle ground between current full node and electrum

Docker Publish passing | Docker Publish passing | functional.yml passing

Floresta

Welcome to Floresta, a lightweight Bitcoin full node implementation written in Rust, powered by Utreexo a novel dynamic accumulator designed for the Bitcoin UTXO set.

This project is composed of two parts, `libfloresta` and `florestad`. `libfloresta` is a set of reusable components that can be used to build Bitcoin applications. `florestad` is built on top of `libfloresta` to provide a full node implementation, including a watch-only wallet and an Electrum server. If you just want to run a full node, you can use `florestad` directly, either by building it from source or by downloading a pre-built binary from the releases.

If you want to use `libfloresta` to build your own Bitcoin application, you can find the documentation here.

github.com/vinteumorg/Floresta

Future plans

Lightning protocol extensions

- Channel announcements need to take utreexo into consideration

Utreexo node + Lightning nodes

Conclusion

- Massive security improvement over electrum
- Massive ease of use gain over current full nodes