

This is a snapshot of <https://hackmd.io/gjxd0c4oTp-ZB81PtHdU2w>, taken on 7th Oct 2022 for reference purposes.

Zebra Mining RPC - Scope

This is a rough scope for Zebra's mining RPCs.

Estimates

Estimate - 2 months, including testing:

- 1 sprint to do most RPCs and the mempool bug fix
- 1 sprint to finish `getblocktemplate`
- 2 sprints for testing

It seems less complex than the light wallet RPCs, but we need to check.

Confirming Estimates

1. Find some mining software that is actually used for Zcash mining
2. Check the RPCs it actually uses
3. Open detailed tickets for each RPC, with a list of required parameters and fields, and their definitions
4. Work out how each parameter or field gets used/handled

It seems like most mining pools are closed-source. There seems to be no open alternative to test the RPCs with, like we did with `lightwalletd`. Maybe we could ask ECC engineers if they know which RPCs are actually required by mining pools.

Required Mining RPCs

- `getblocktemplate`
 - `parameters:`
 - `rules: segwit` - not in Zcash
 - `fields:`
 - `height`

- version
 - curtime
 - bits
 - new state request
 - previousblockhash
 - target
 - same state request as bits
 - transactions - get from mempool, needs [consistency fix](#)
 - data
 - hash
 - txid
 - coinbaseaux - undocumented constant
 - coinbasevalue - calculation based on consensus rules
 - default_witness_commitment - not in Zcash (required by SegWit)
- we might need to return an error until Zebra is at the tip
- we might need part of [Limit generated block and transaction size](#)
- [submitblock](#)
 - parameters
 - hexdata
 - send through the block validator, just like downloaded blocks
 - it will get gossipped to peers, and picked up by the next RPC calls
 - we might need to return an error until Zebra is at the tip
- [getblockhash](#)
 - similar to `getbestblockhash`
- [getblockcount](#)
 - similar to `getblockchaininfo.blocks`
- [getblocksubsidy](#)
 - use the existing subsidy calculation code in `zebra-chain`

Questions

Q: Is it worth implementing mining RPCs if we plan on moving to Proof of Stake?

A: We're still going to need to make some kind of block template under Proof of Stake.

How can we minimise the scope of what we implement?

Which mining pool software and production mining pool can we test with?

Reference

List of Miner Source Code

We used this code to discover the RPCs that are actually used by miners:

https://github.com/viabtc/viabtc_mining_server/blob/main/jobmaster/jm_job.c#L881

https://github.com/viabtc/viabtc_mining_server/blob/main/blockmaster/bm_block.c#L79

Outdated?

<https://github.com/braiins/cgminer/blob/braiins-am1/cgminer.c#L662>

Mining Pool Statistics

<https://miningpoolstats.stream/zcash>

Full List of Mining RPCs

Some of these RPCs are not used by most miners.

- [getblocksubsidy](#)
- [getblocktemplate](#)
- [getlocalsolps](#)
- [getmininginfo](#)
- [getnetworkhashps](#)
- [getnetworksolps](#)
- [prioritisetransaction](#)
- [submitblock](#)