Mempool:
What's in our {Past,Future}?

Gloria "glozow" Zhao

6B00 2C6E A3F9 1B1B 0DF0 C9BC 8F61 7F12 00A6 D25C

History of Mempool

2009

The Beginning Of Time (2008)

Original Mempool

2009

Mempool has no limit, txs don't need to pay fees unless big, dust, or too many

```
int64 GetMinFee(bool fDiscount=false) const
{
    // Base fee is 1 cent per kilobyte
    unsigned int nBytes = ::GetSerializeSize(*this, SER_NETWORK);
    int64 nMinFee = (1 + (int64)nBytes / 1000) * CENT;

    // First 100 transactions in a block are free
    if (fDiscount && nBytes < 10000)
        nMinFee = 0;

    // To limit dust spam, require a 0.01 fee if any output is less than 0.01
    if (nMinFee < CENT)
        foreach(const CTxOut& txout, vout)
            if (txout.nValue < CENT)
                nMinFee = CENT;

    return nMinFee;
}
```

2009

Original Mempool

2009

Mempool has no limit, txs don't need to pay fees unless big, dust, or too many, "standard templates" are P2PK and P2PKH

```
int64 GetMinFee(bool fDiscount=false) const
{
    // Base fee is 1 cent per kilobyte
    unsigned int nBytes = ::GetSerializeSize(*this, SER_NETWORK);
    int64 nMinFee = (1 + (int64)nBytes / 1000) * CENT;

    // First 100 transactions in a block are free
    if (fDiscount && nBytes < 10000)
        nMinFee = 0;

    // To limit dust spam, require a 0.01 fee if any output is less than 0.01
    if (nMinFee < CENT)
        foreach(const CTxOut& txout, vout)
            if (txout.nValue < CENT)
                nMinFee = CENT;

    return nMinFee;
}
```

```
{
    // Standard tx, sender provides pubkey, receiver adds signature
    vTemplates.push_back(CScript() << OP_PUBKEY << OP_CHECKSIG);

    // Short account number tx, sender provides hash of pubkey, receiver provides signature and pubkey
    vTemplates.push_back(CScript() << OP_DUP << OP_HASH160 << OP_PUBKEYHASH << OP_EQUALVERIFY << OP_CHECKSIG);
}
```

2009

2010

Some transactions with high “priority” aka “coin age” don’t need to pay fees

Free relay allowed for txs with high coin age

Transaction relay is highly exposed to Denial of Service

Make transactions larger than 100K non-standard #2273

Merged gavinandresen merged 1 commit into bitcoin:master from gavinandresen:txsize on Feb 5, 2013

Reject transactions with excessive numbers of sigops #4150

Merged laanwj merged 1 commit into bitcoin:master from petertodd:reject-excessive-sigops on Aug 12, 2014

2012-2014

Standardness rules: dust, max size, max sigops, OP_RETURN, upgradeable NOPs, push-only P2SH

Relay policies shape wallet behavior

2009

Make transactions larger than 100K non-standard #2273

Merged gavinandresen merged 1 commit into bitcoin:master from gavinandresen:txsize on Feb 5, 2013

Reject transactions with excessive numbers of sigops #4150

Merged laanwj merged 1 commit into bitcoin:master from petertodd:reject-excessive-sigops on Aug 12, 2014

Make 0-value outputs non-standard #1718

Merged gavinandresen merged 1 commit into bitcoin:master from sipa:nozeroout on Aug 25, 2012

Require strictly-standard encodings in mempool #2520

Merged gavinandresen merged 1 commit into bitcoin:master from sipa:strictmempool on Apr 14, 2013

Check for canonical public keys and signatures #1742

Merged jgarzik merged 1 commit into bitcoin:master from sipa:canonical on Oct 20, 2012

Make signatures with non-canonical data pushes non-standard. #3025

Merged gmaxwell merged 2 commits into bitcoin:master from sipa:noncanpush on Feb 11, 2014

2012-2014

Standardness rules: dust, max size, max sigops, OP_RETURN, upgradeable NOPs, push-only P2SH

Relay policies shape wallet behavior

2009

Make transactions larger than 100K non-standard #2273

Merged gavinandresen merged 1 commit into bitcoin:master from gavinandresen:txsize on Feb 5, 2013

Reject transactions with excessive numbers of sigops #4150

Merged laanwj merged 1 commit into bitcoin:master from petertodd:reject-excessive-sigops on Aug 12, 2014

Make 0-value outputs non-standard #1718

Merged gavinandresen merged 1 commit into bitcoin:master from sipa:nozeroout on Aug 25, 2012

Require strictly-standard encodings in mempool #2520

Merged gavinandresen merged 1 commit into bitcoin:master from sipa:strictmempool on Apr 14, 2013

Check for canonical public keys and signatures #1742

Merged jgarzik merged 1 commit into bitcoin:master from sipa:canonical on Oct 20, 2012

Make signatures with non-canonical data pushes non-standard. #3025

Merged gmaxwell merged 2 commits into bitcoin:master from sipa:noncanpush on Feb 11, 2014

Relay OP_RETURN TxOut as standard transaction type #3128

Merged gavinandresen merged 2 commits into bitcoin:master from petertodd:tx_null on Oct 31, 2013

Discourage NOPs reserved for soft-fork upgrades #5000

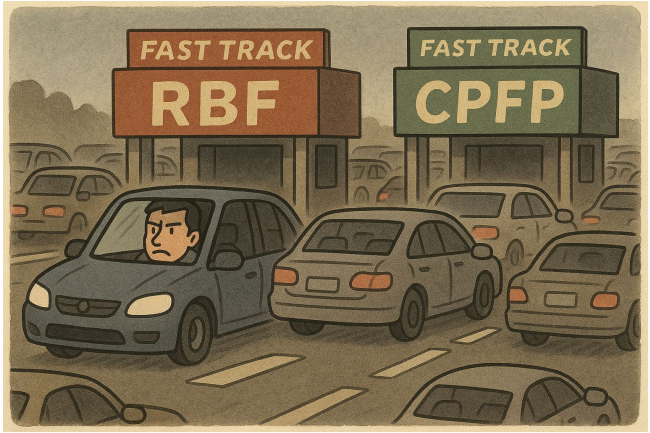
Merged sipa merged 1 commit into bitcoin:master from petertodd:blacklist-upgradable-nops on Nov 20, 2014

2012-2014

Standardness rules: dust, max size, max sigops, OP_RETURN, upgradeable NOPs, push-only P2SH

Relay policies shape wallet behavior

2009



2015

Replace by Fee

Mempool size limit,
dynamic min feerate

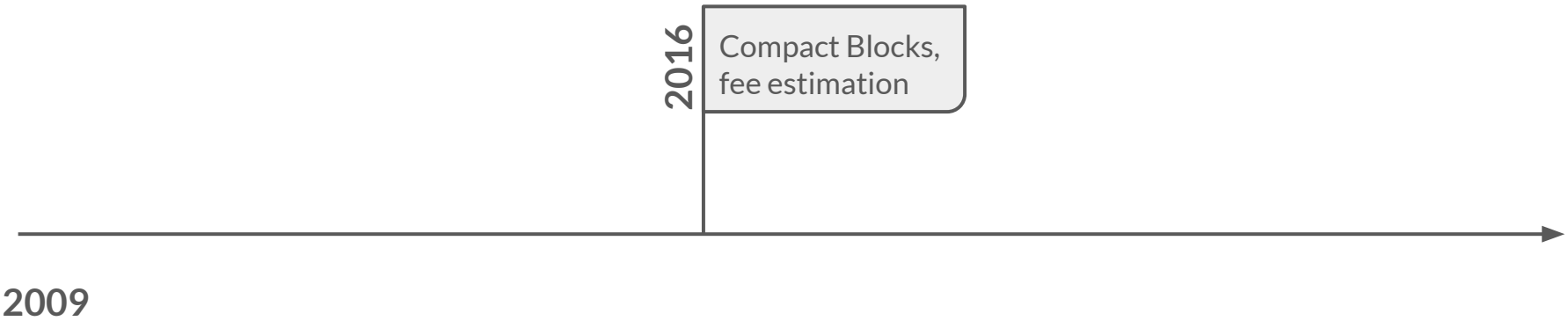
2016

Ancestor/descendant
package tracking,
Child Pays for Parent

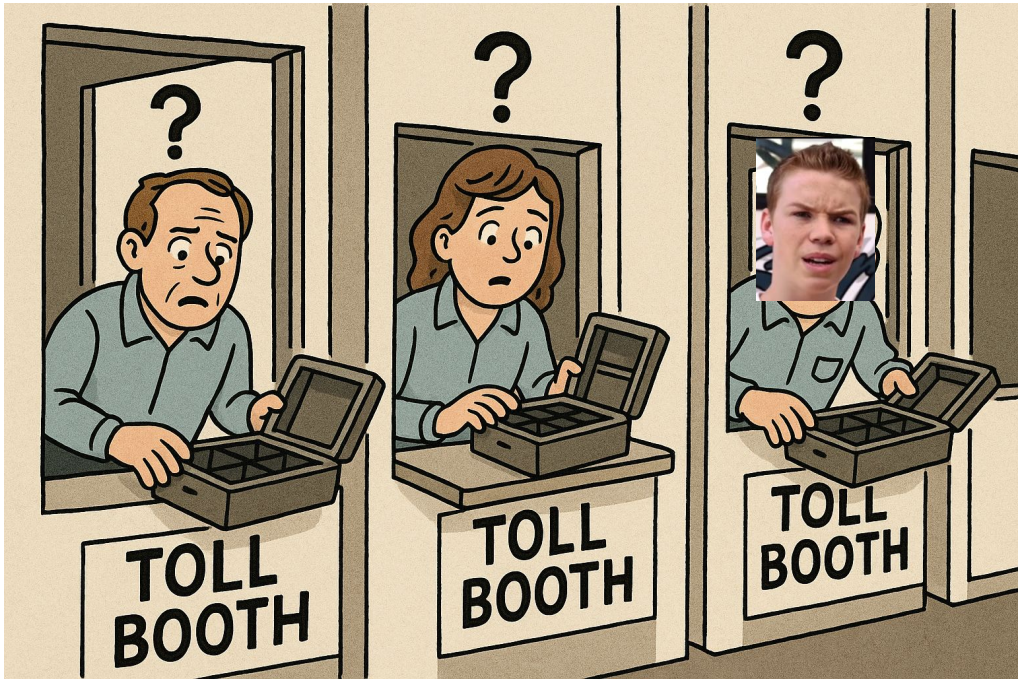
Fee market emerges

2009

Transaction relay isn’t altruistic



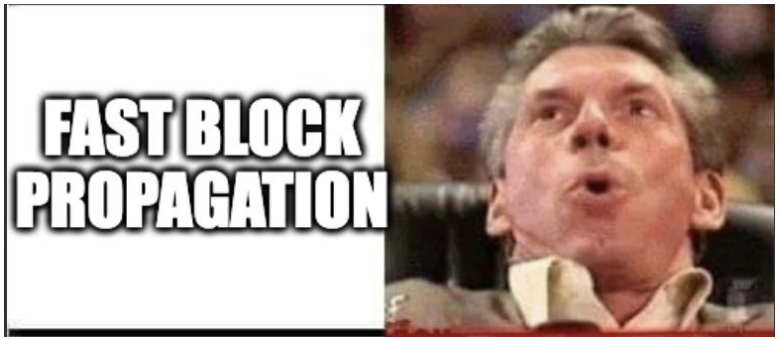
Hold on a second…
non-mining nodes don’t get paid



Individuals with tx cache:

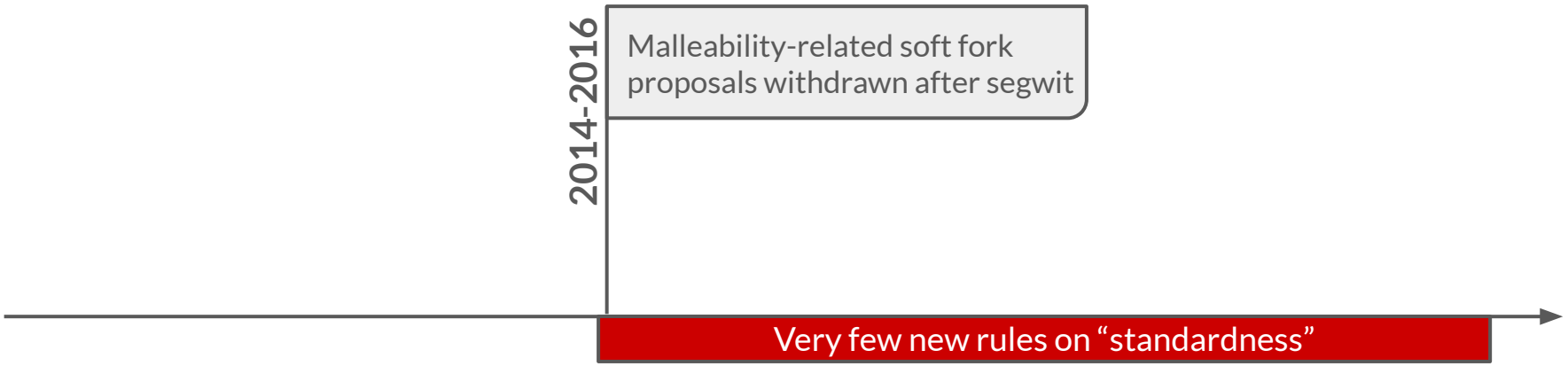
Individuals with tx cache:

Network of warm caches:



Individuals with tx cache:

Network of warm caches:


2014-2016
Malleability-related soft fork proposals withdrawn after segwit
Very few new rules on “standardness”
2009

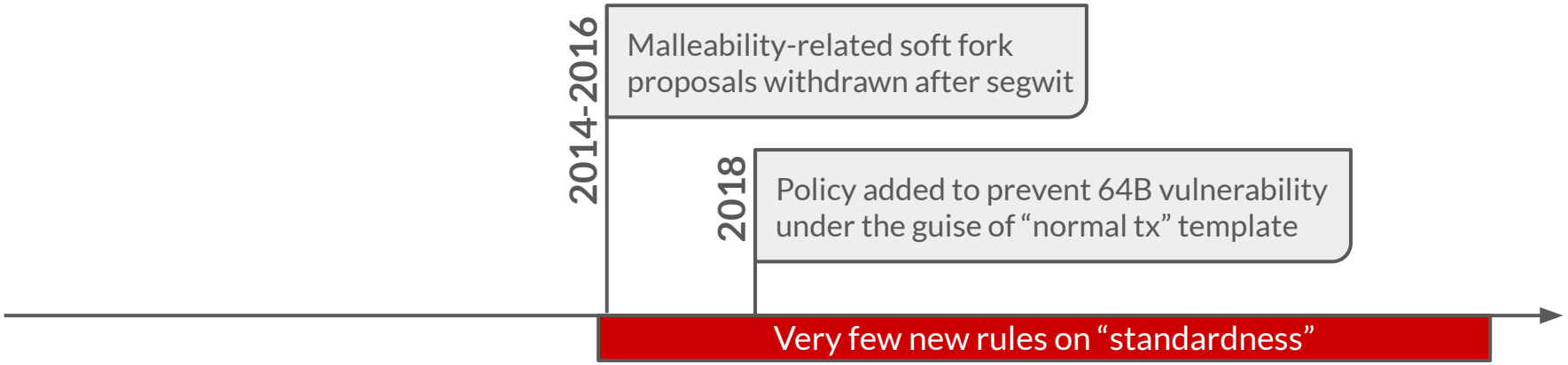
[Policy] Several transaction standardness rules #11423

Merged laanwj merged 4 commits into bitcoin:master from jl2012:const_scriptcode on May 12, 2018

```
25 + /** The minimum non-witness size for transactions we're willing to relay/mine (1 segwit input + 1 P2WPKH output = 82 bytes) */
26 + static const unsigned int MIN_STANDARD_TX_NONWITNESS_SIZE = 82;
```

CVE-2017-12842 PUBLISHED

Bitcoin Core before 0.14 allows an attacker to create an ostensibly valid SPV proof for a payment to a victim who uses an SPV wallet, even if that payment did not actually occur. Completing the attack would cost more than a million dollars, and is relevant mainly only in situations where an autonomous system relies solely on an SPV proof for transactions of a greater dollar amount.



2009

Pinning and fee-bumping are security issues for L2s

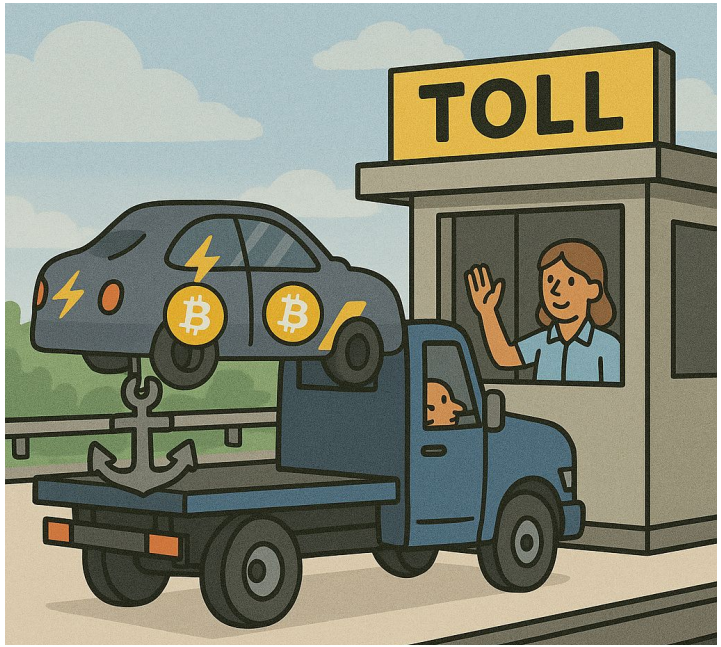
2019 CPFP Carve-Out

2020 Lightning anchor outputs

Policy prioritizes market efficiency

2009

New “templates” for fee-bumping presigned txs



2024

1p1c: TRUC, Package RBF, opportunistic relay

P2Anchor

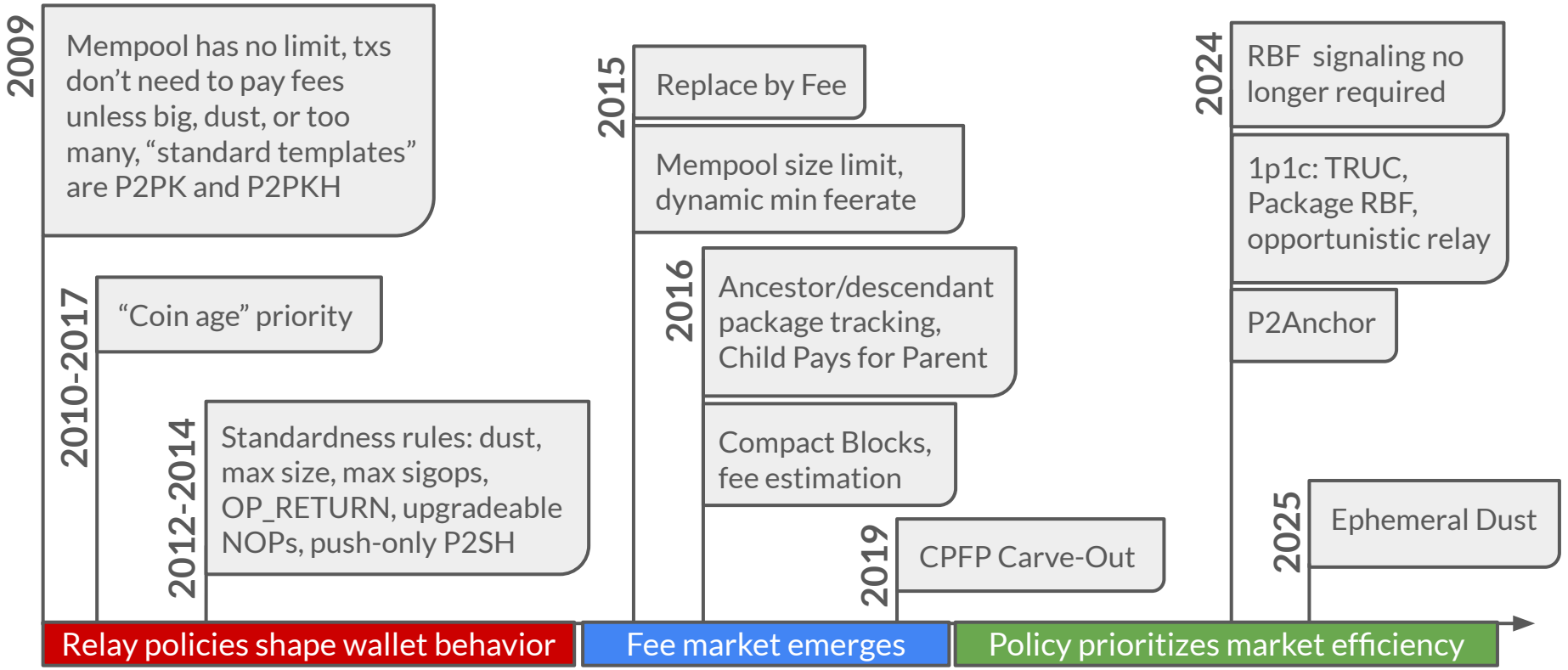
2025

Ephemeral Dust

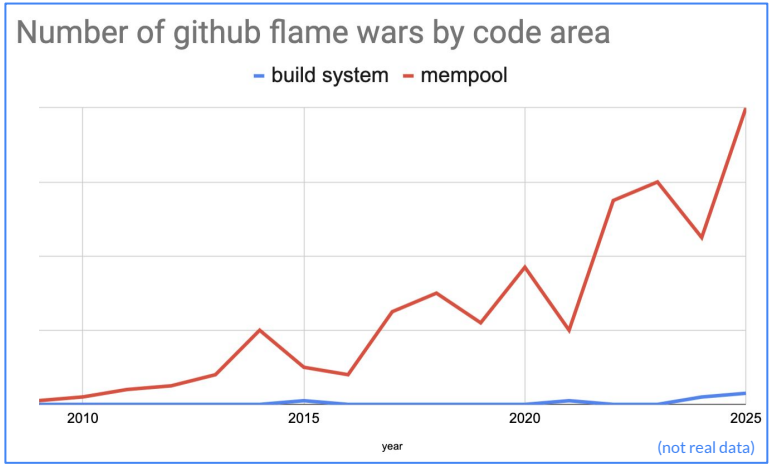
Policy prioritizes market efficiency

2009

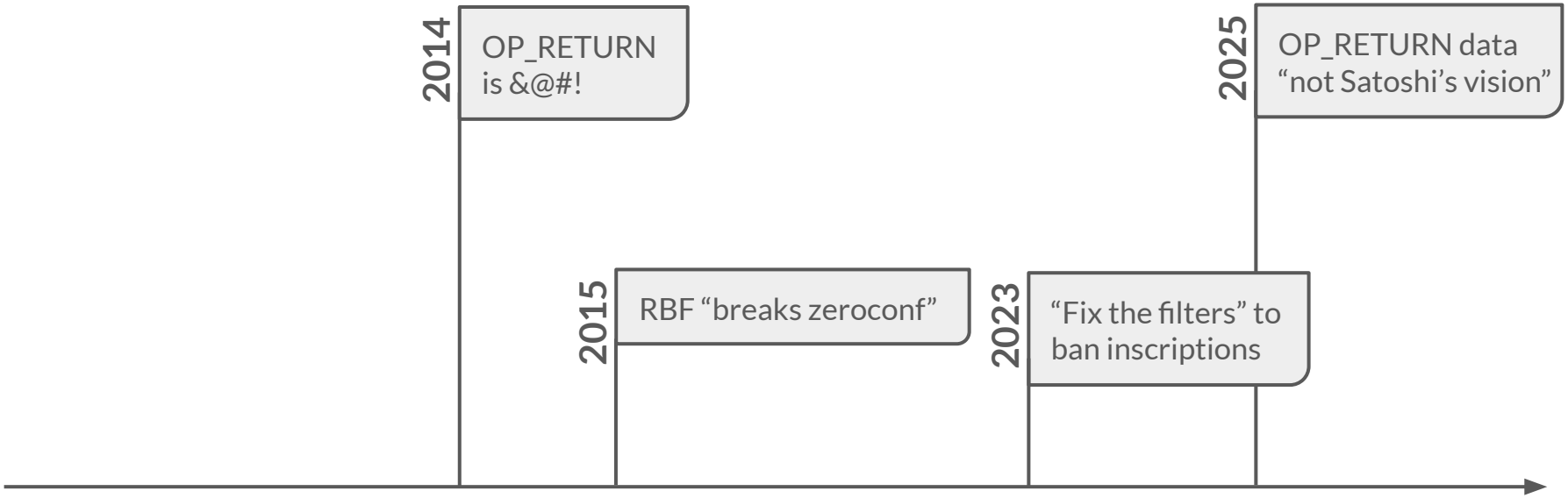
History of Mempool



Not pictured: History of Mempool Config Options



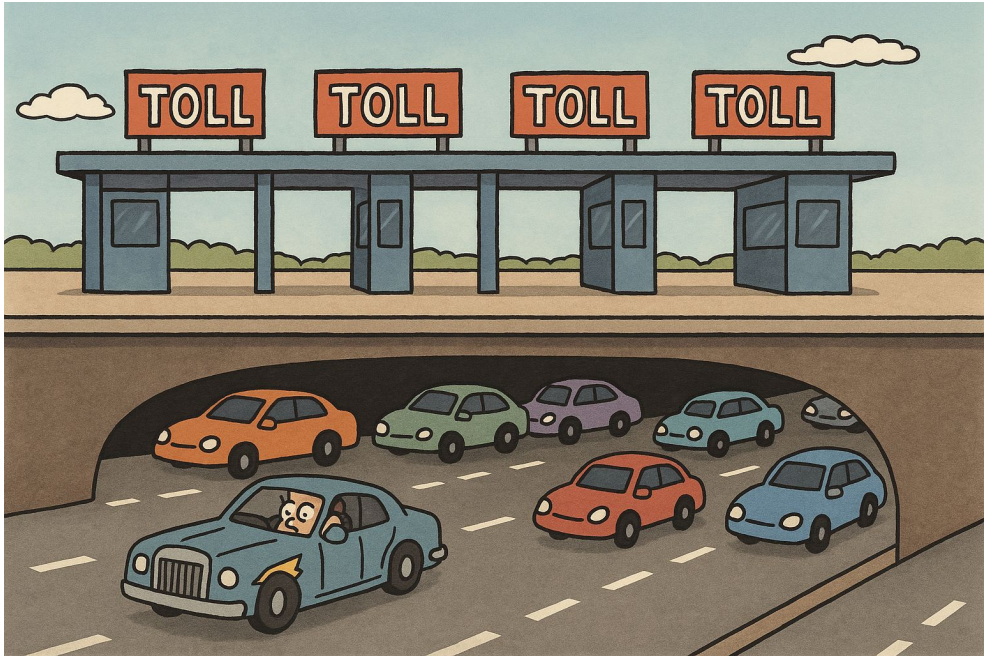
Not pictured: fights about “what mempool is for”

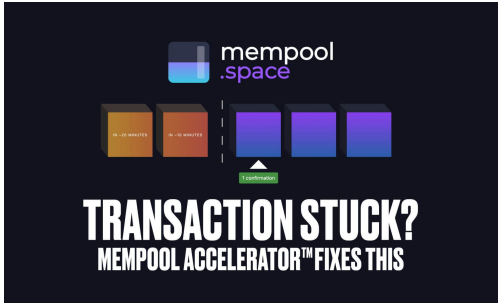


How are we doing today?

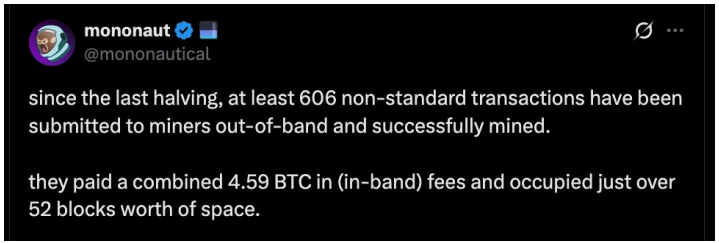
- Pinning is hard
- RBF rules suck
- Fee management is hard

What happens when fee-bumping is inefficient?





How are we doing today?



How are we doing today?

mononaut ✅ 🟦
@mononautical

since the last halving, at least 606 non-standard transactions have been submitted to miners out-of-band and successfully mined.

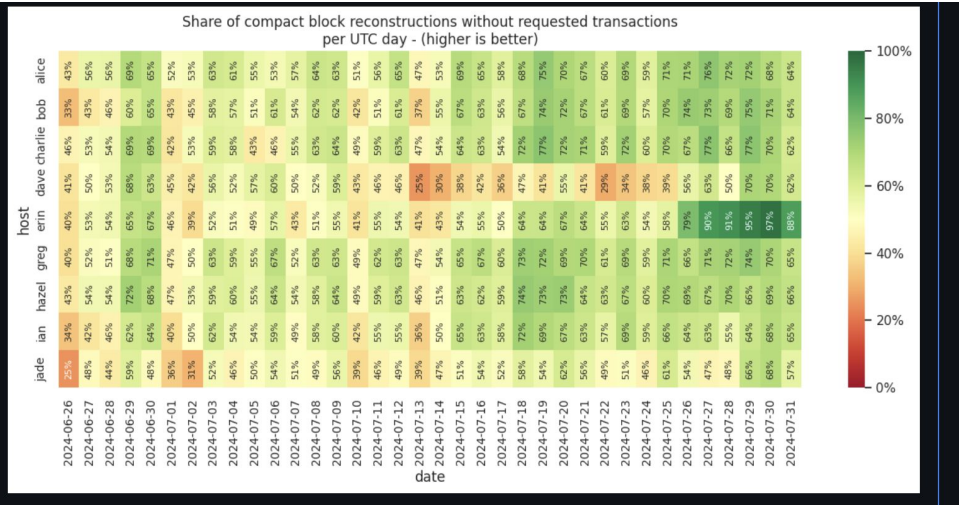
they paid a combined 4.59 BTC in (in-band) fees and occupied just over 52 blocks worth of space.

0xB10C commented on Aug 1, 2024 — Member

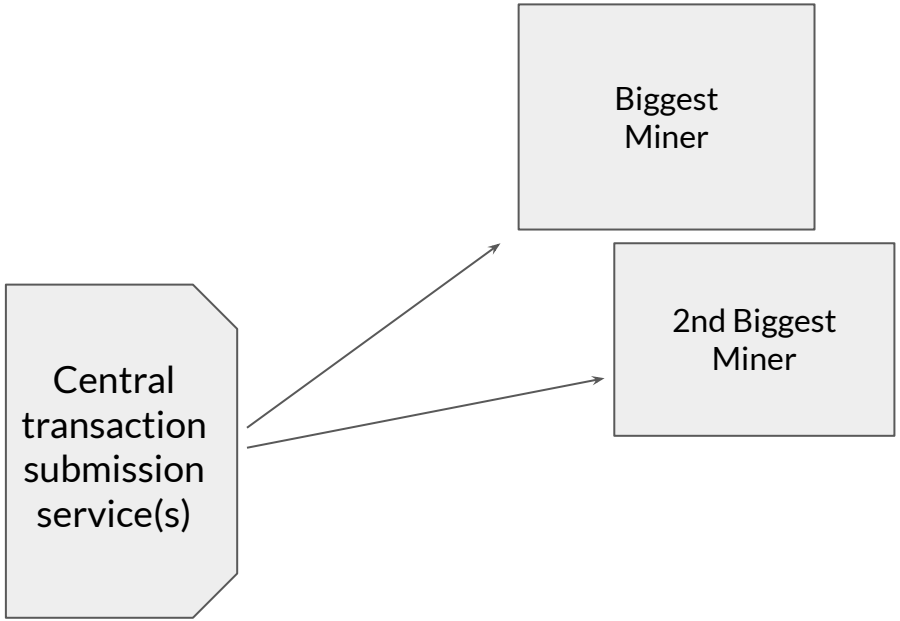
> I shared my concerns with another bitcoiner who likes to monitor network health, they replicated, and got similar results (which I presume they'll share eventually).

Correct, I too found enabling `mempoolfullrbf` to be helpful for compact block reconstruction without requesting transactions.

By analyzing compact block reconstructions in the debug.logs (with `debug=cmpctblock` enabled) of my monitoring nodes, I noticed that many block reconstructions need an additional round-trip to fetch unknown transactions. While this situation has slightly improved over the last weeks (~50% -> 70+%) as mempool activity died down, this still clearly shows a divergence in policy.

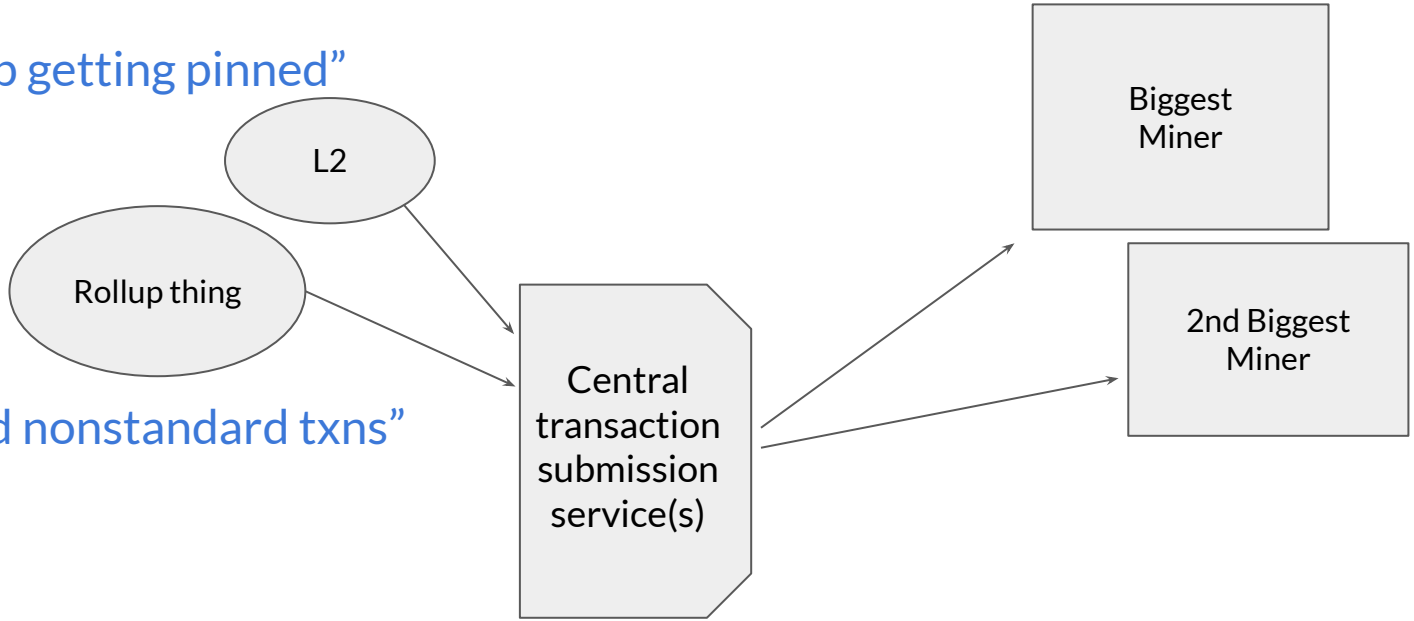


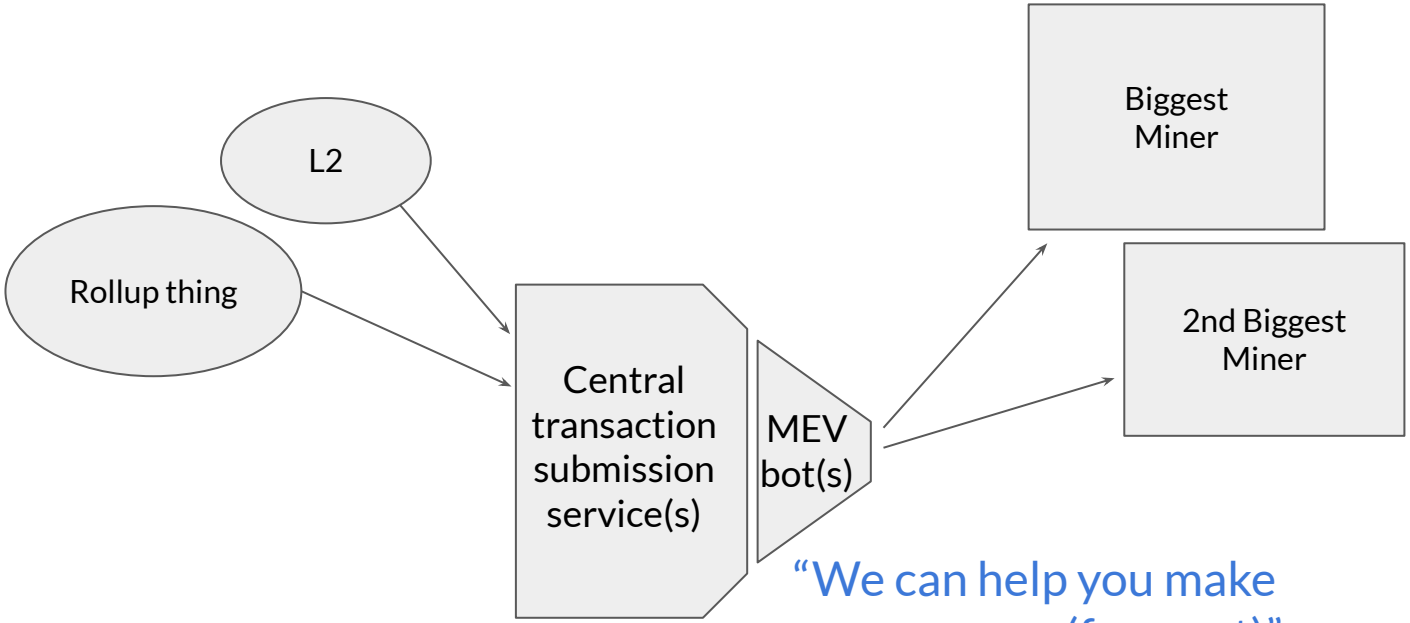
“We want to make more money”

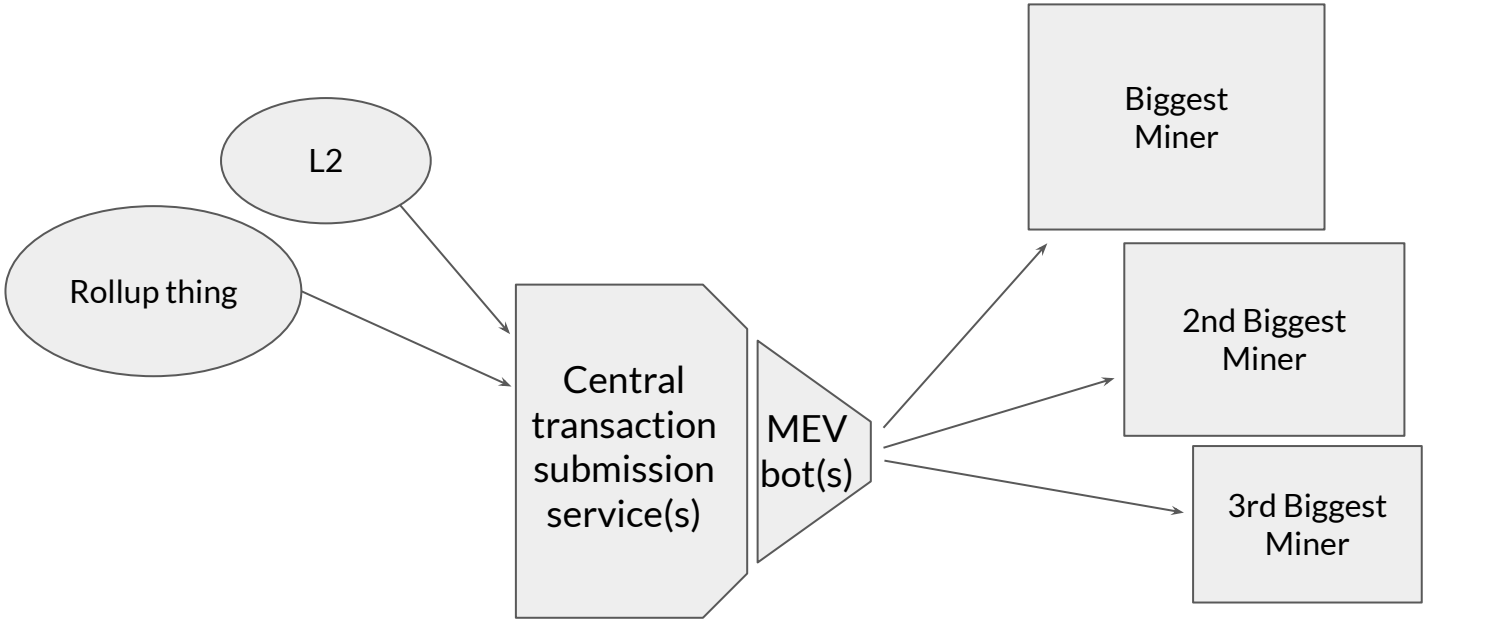


“We keep getting pinned”

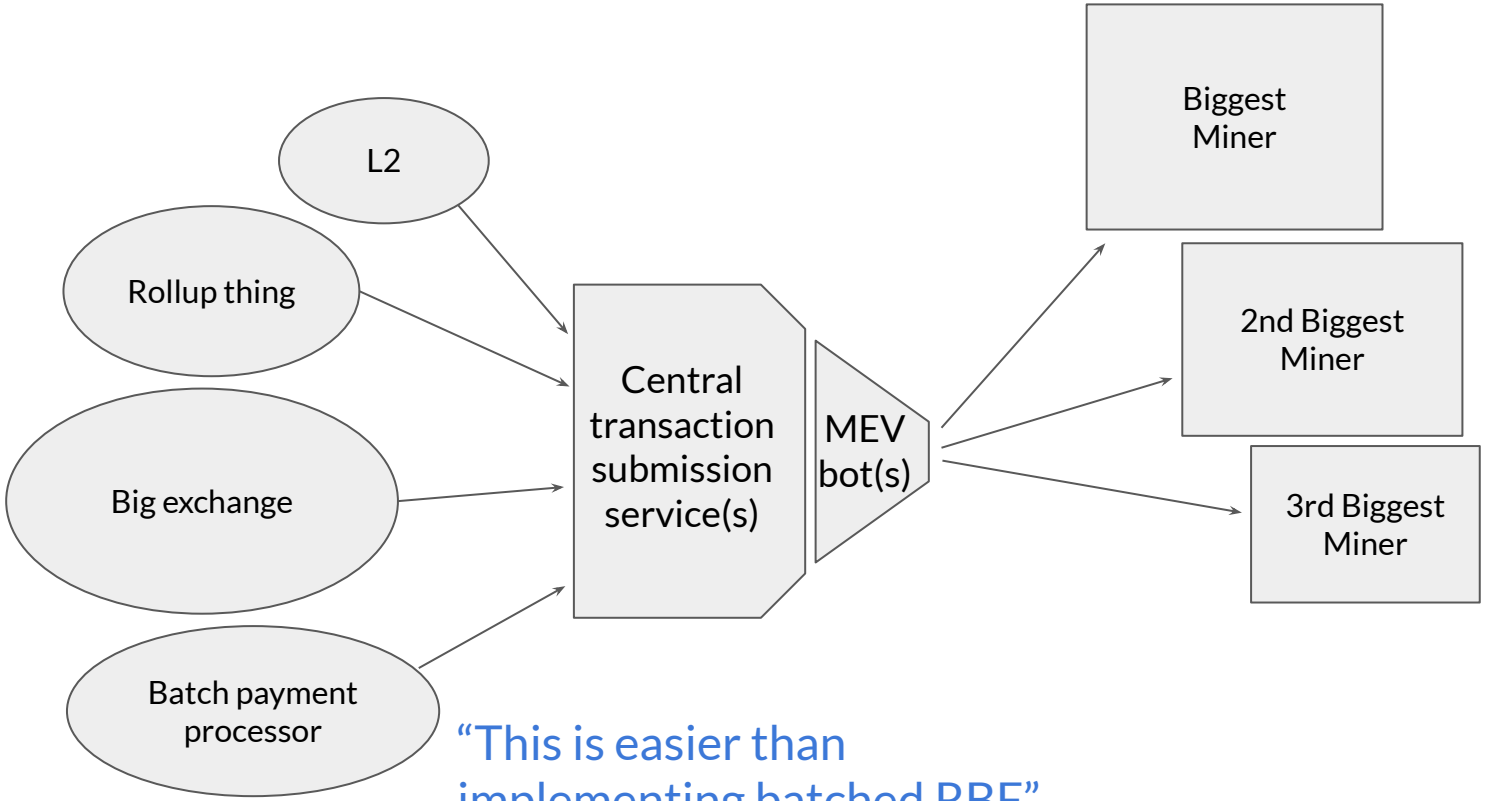
“We need nonstandard txns”



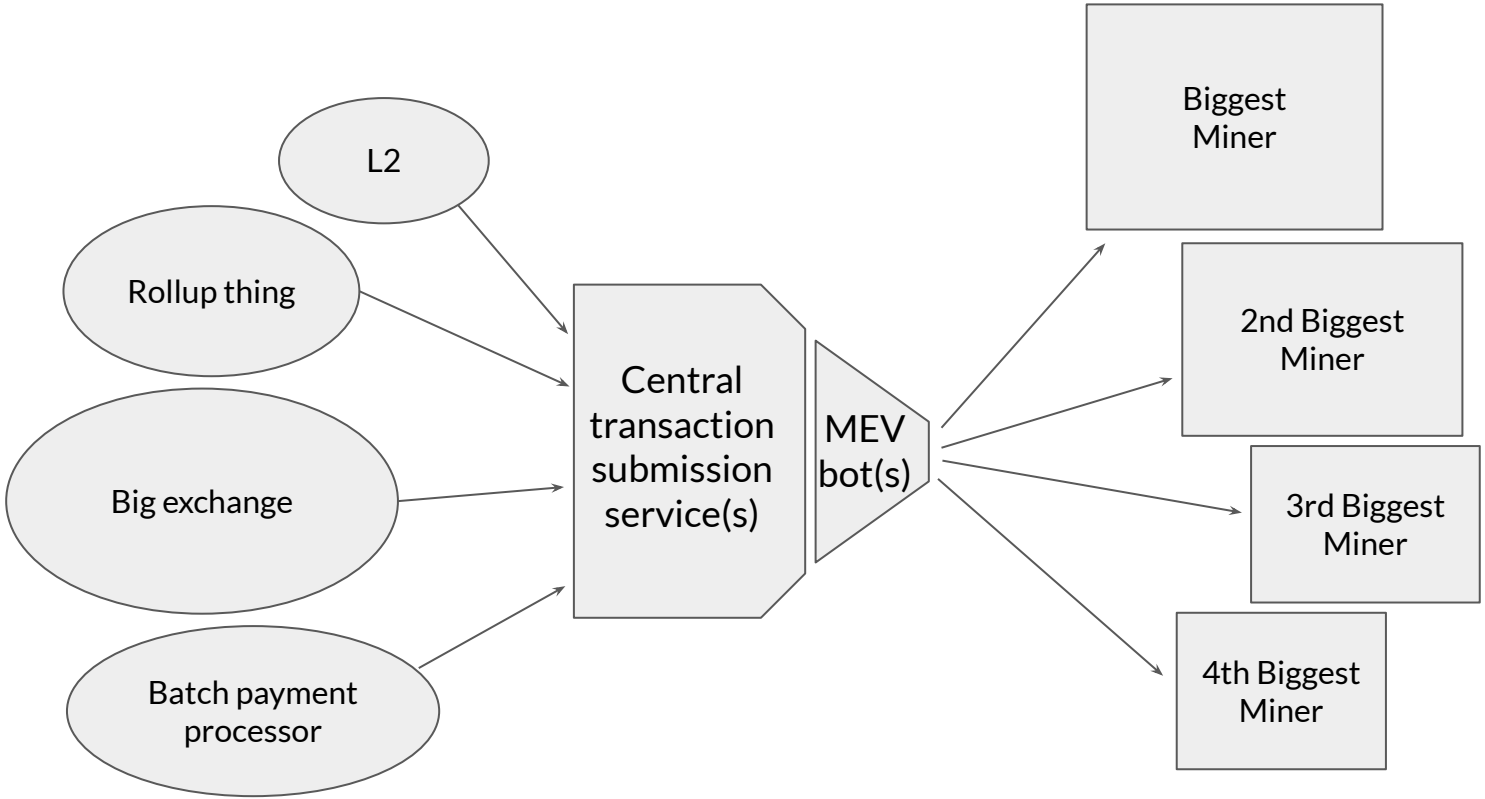

L2
Rollup thing
Central transaction submission service(s)
MEV bot(s)
Biggest Miner
2nd Biggest Miner
“We can help you make more money (for a cut)”



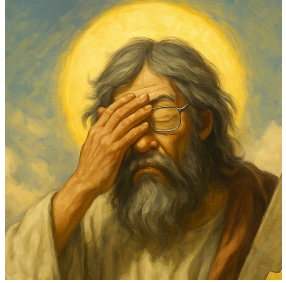
"We need to stay competitive"

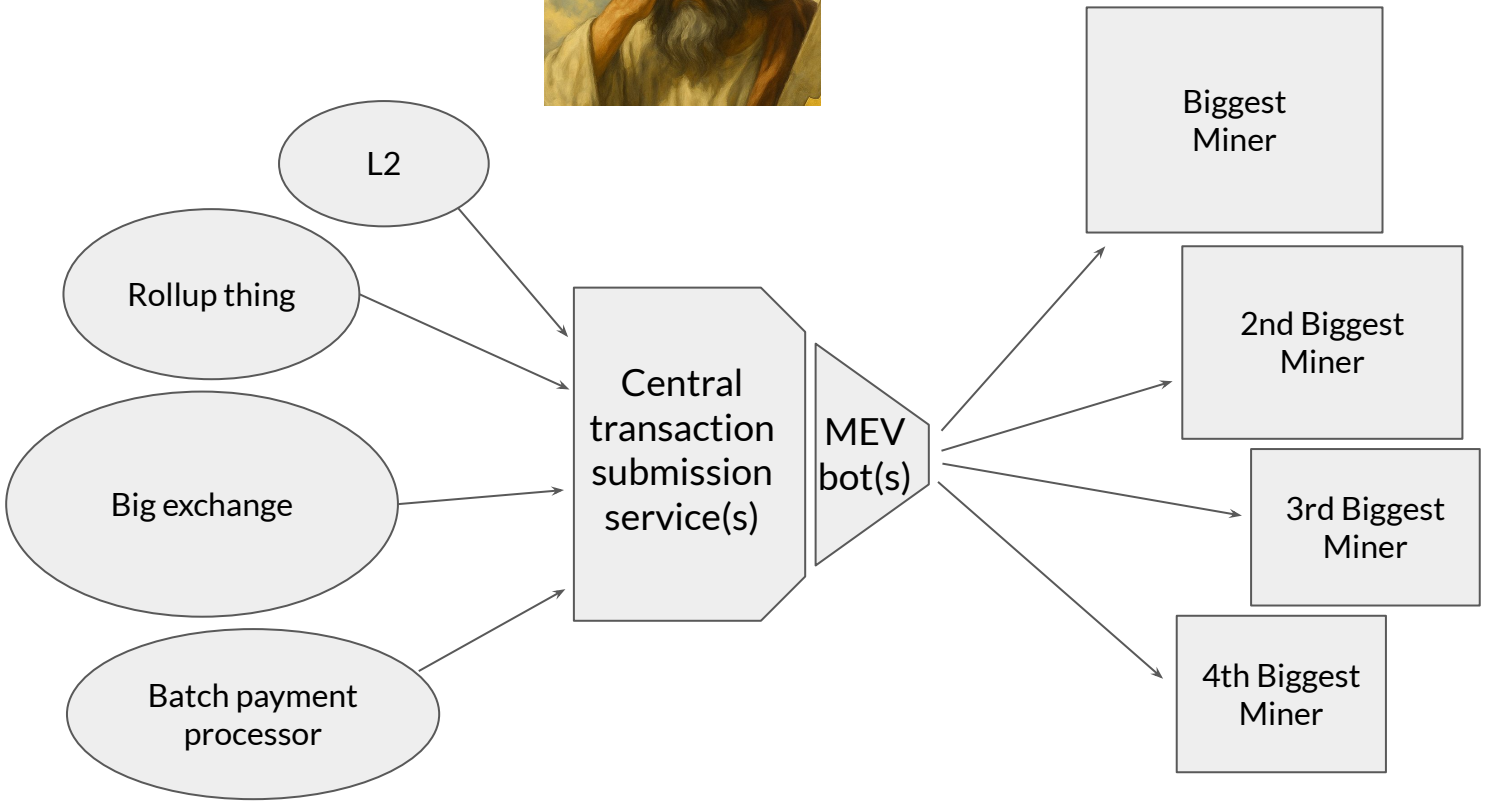


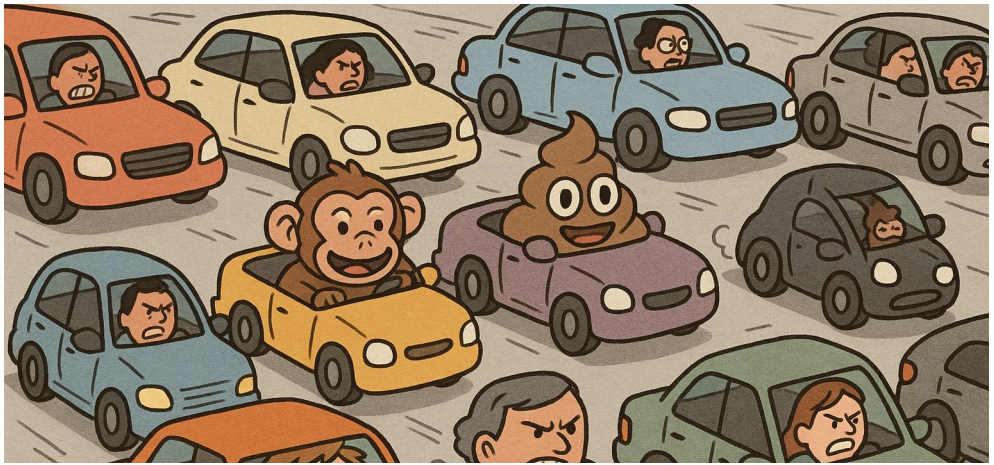
"This is easier than implementing batched RBF"



"I get -30% fees if I don't this"




L2
Rollup thing
Big exchange
Batch payment processor
Central transaction submission service(s)
MEV bot(s)
Biggest Miner
2nd Biggest Miner
3rd Biggest Miner
4th Biggest Miner



2015

The “Filter” Ship

2009

What should we do?

Short to medium term:

- Prioritize fee-bumping efficiency and incentive compatibility
- Don’t build MEVy things
- Drop paternalistic policies that hinder block propagation
- If monkey JPEGs are stupid, price them out

What should we do?

Short to medium term:

- Prioritize fee-bumping efficiency and incentive compatibility
- Don’t build MEVy things
- Drop paternalistic policies that hinder block propagation
- If monkey JPEGs are stupid, price them out

Long term:

- If it’s really that bad, make it a consensus rule
- Maybe ditch exogenous fees

“Ok, uh… just get rid of policy!”

- Denial of Service
- Upgrade hooks
- Templates
- Incentive compatibility

Don’t give up on decentralized mining

- Easy
- Only requires talking to friends

- Hard
- Requires a lot of fighting