

# Current state and the future of wallets

Building On Bitcoin  
3th of July 2018



dev@jonasschnelli.ch

PGP: CA1A2908DCE2F13074C62CDE1EB776BB03C7922D



BitcoinCore

**Privacy**

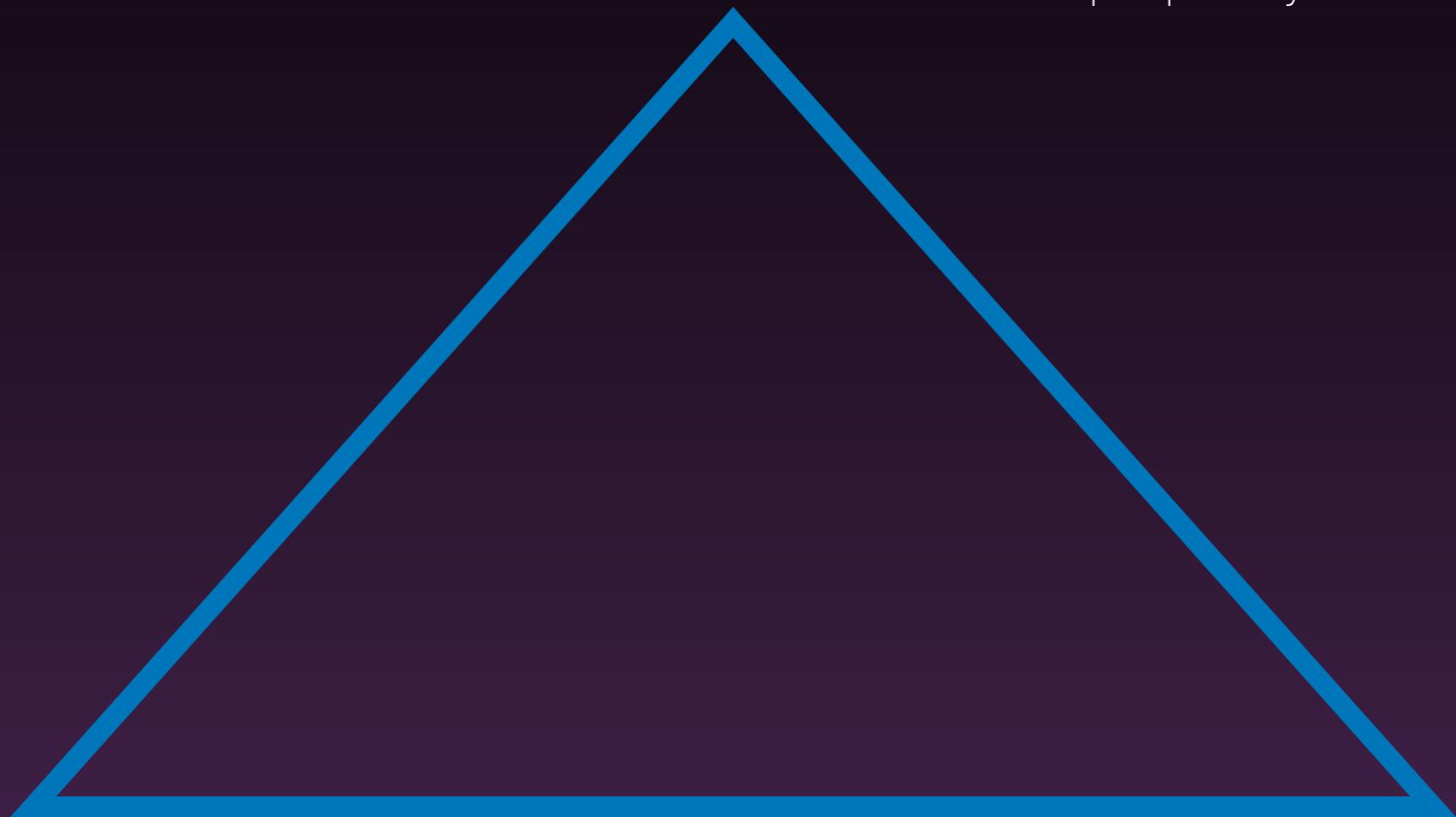


**Security**

**Trust**

# Privacy

Transaction / scripts privacy



# Security

Keystorage  
Cold-Storage

# Trust

No-trust required  
Chain-Validation  
Consensus



## ✓ Privacy

No scripts sharing



## Security

Missing cold storage

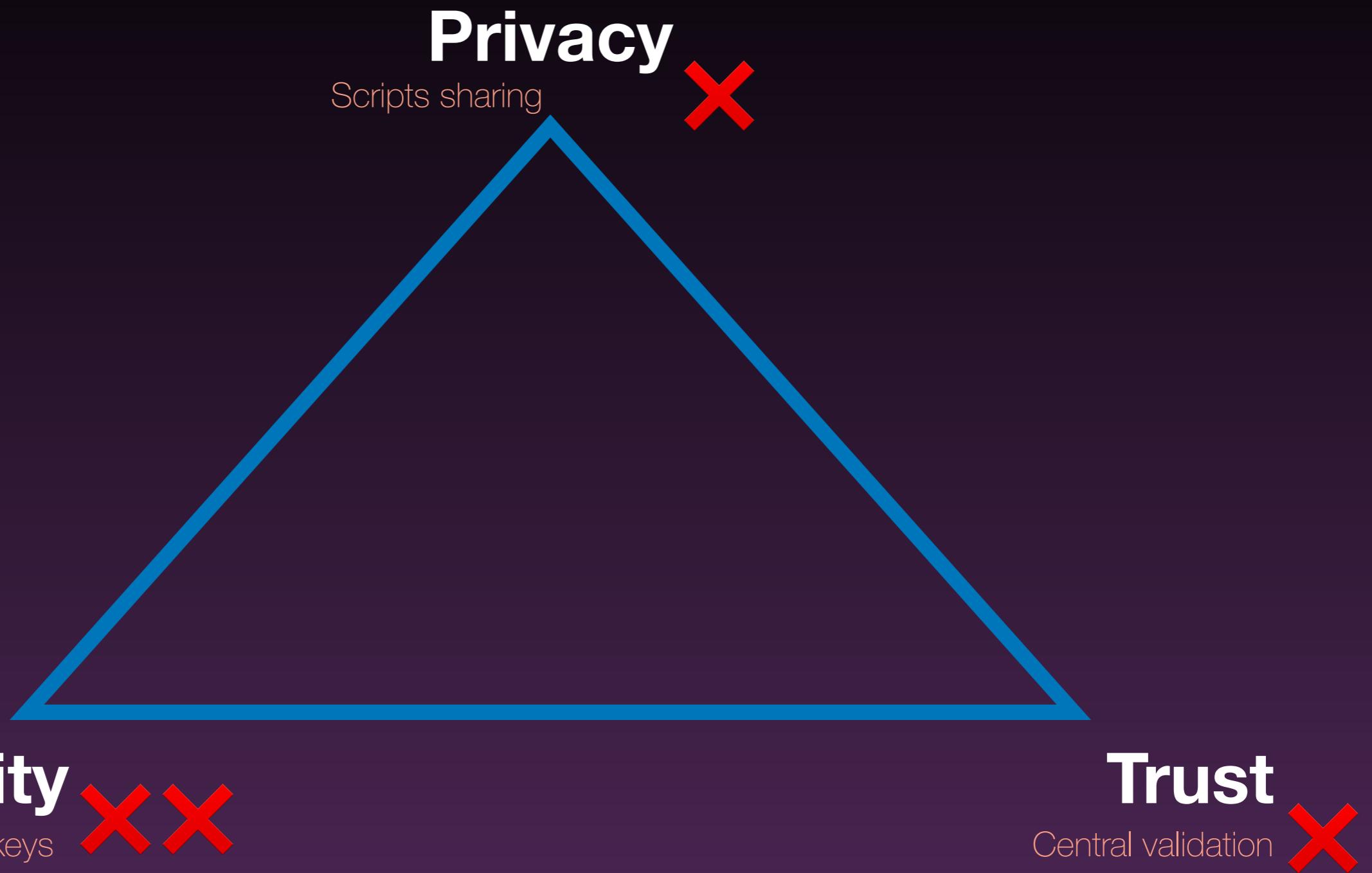


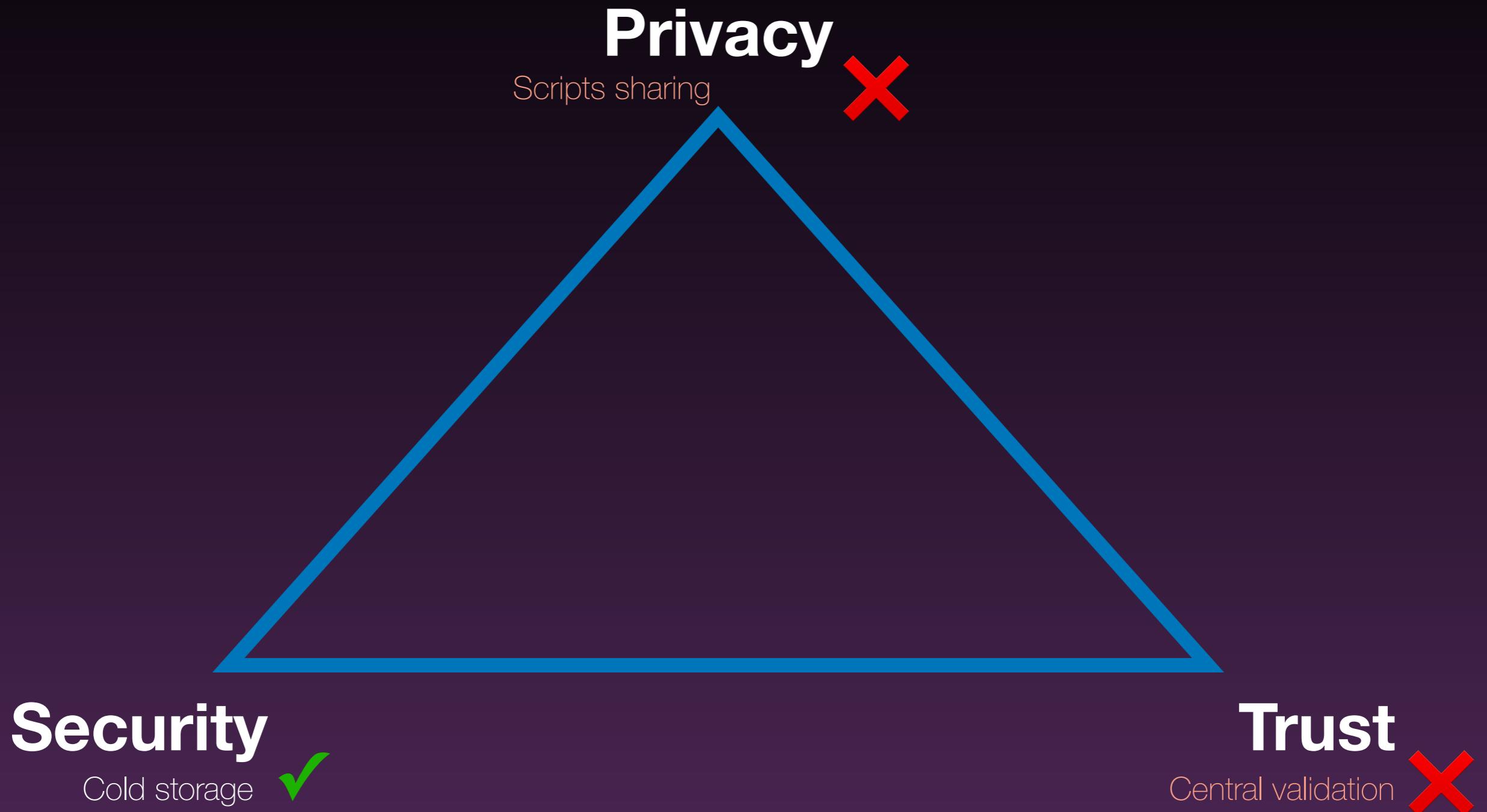
## Trust

Full validation  
Consensus



# coinbase







BreadWallet



Android Wallet

# Privacy

Scripts sharing



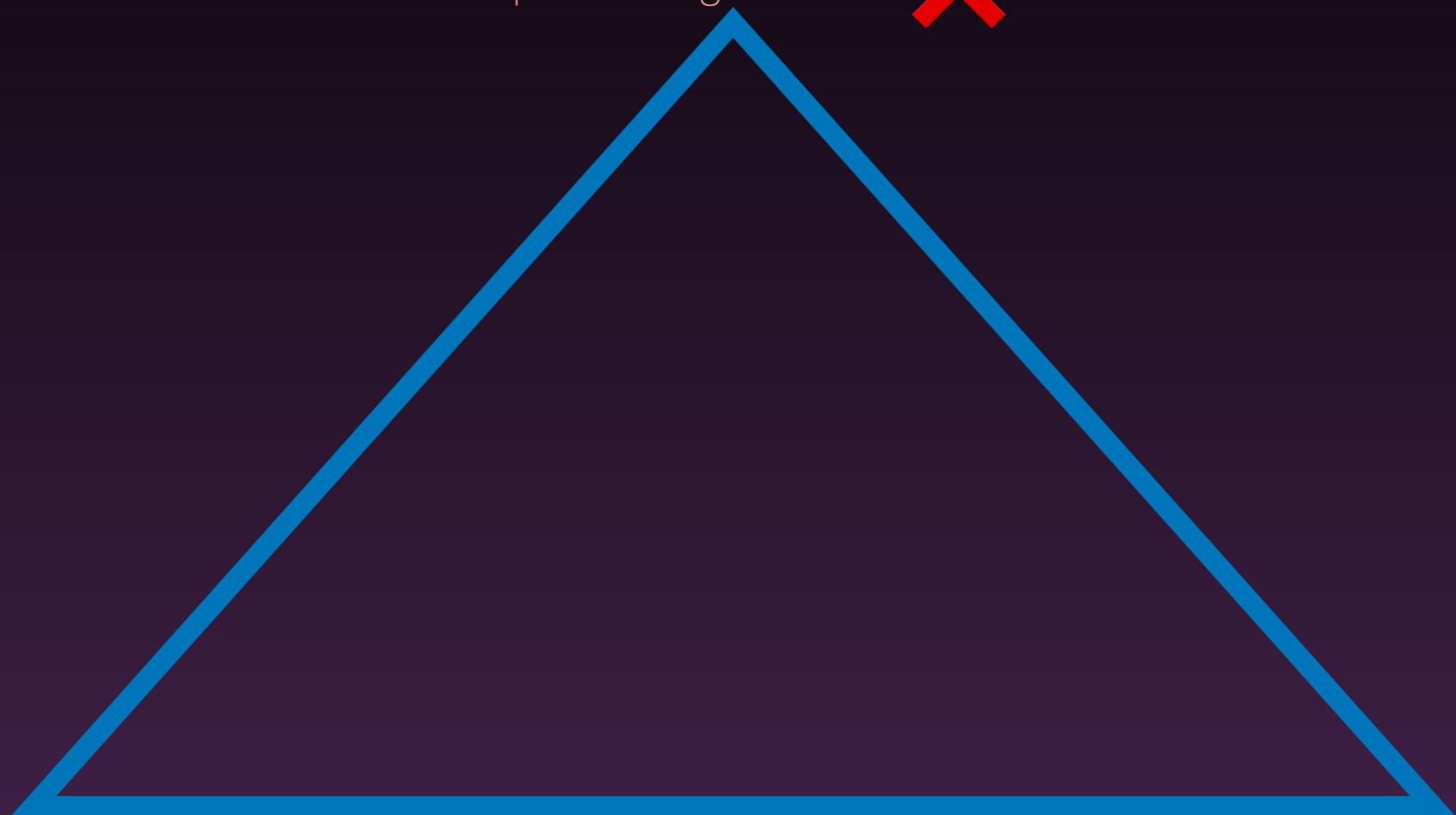
# Security

Exposed keys



# Trust

SPV validation





Electrum

# Privacy

Scripts sharing

✗

# Security

✗

Missing cold storage

# Trust

✓

SPV validation

# Centralized validation



# Current state:

New/novice users **tend** to use centralised validation.

# Current state:

New/novice users **tend** to use centralised validation.

- x Required validation device
- x Validation lead time
- x Bandwidth and CPU requirements

# Centralized validation in practice



ElectrumX

- **~200GB+ disk space** (large indexes)
- **Heavy disk I/O** through indexing
- **Full validation** „underneath“ (Bitcoin Core)

# Downsides of centralized validation

- ✗ **Fake** transactions / transaction **omission**
- ✗ No control over the **consensus layer**
- ✗ Abandons **privacy** completely

# Advantages of centralized validation

- ✓ **Immediately** ready to use
- ✓ Fast wallet **recovery**
- ✓ Very **low bandwidth** costs
- ✓ Can serve **large amount of wallets**

# Centralized key-storage





Centralized key-storage

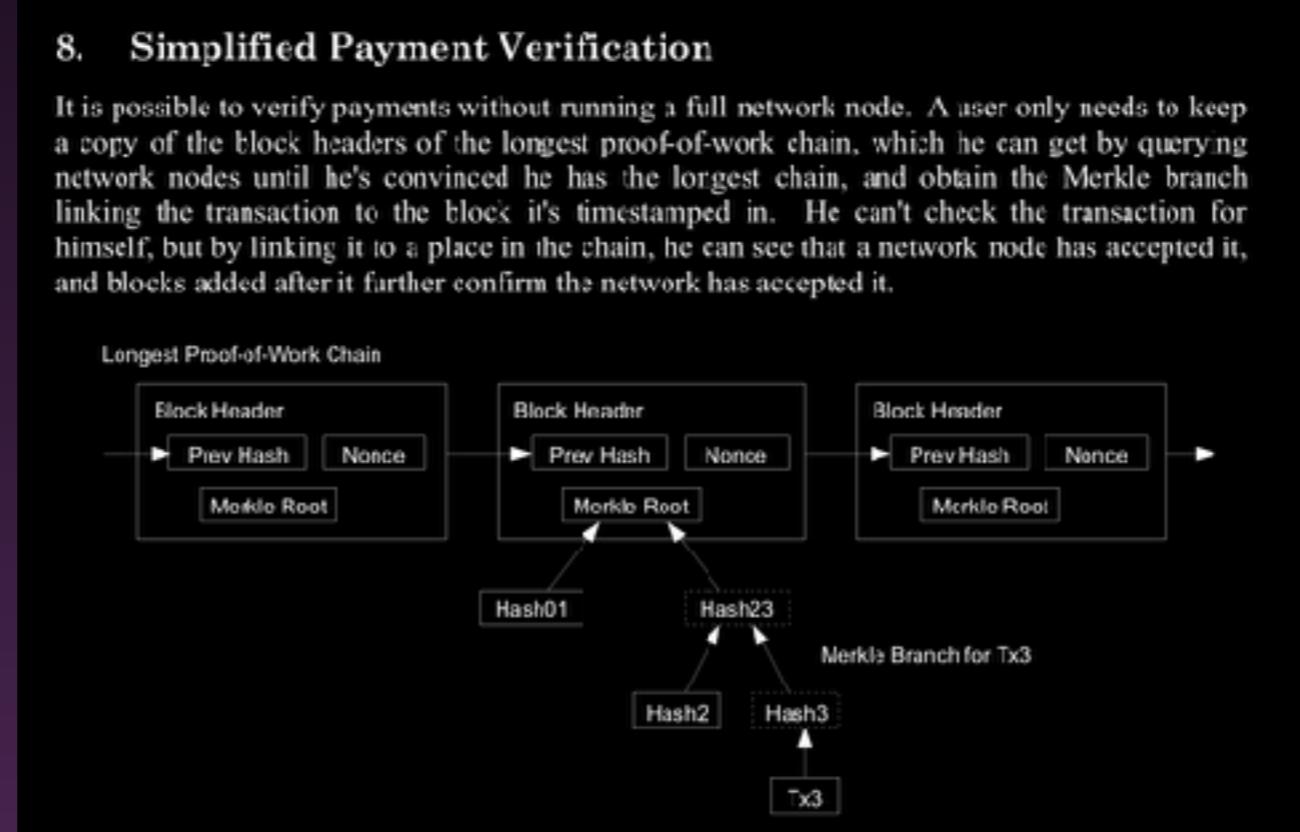
- ✓ No security setup required
- ✗ „Owns“ **no** Bitcoins
- ✗ „Owns“ only the **right** to eventually access and move Bitcoins

Users are often **not aware!**

# SPV

## 8. Simplified Payment Verification

It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.



# SPV

Verify headers ✓

Can check some  
**consensus rules** ✓

Weak **0-conf** handling ✗

Network „**leeches**“ ✗

Rely on a „**free service**“ ✗

**Fee estimation** is  
probably impossible ✗

Often rely on DNS  
seeds ✗

# SPV

Acceptable **Bandwidth**  
consumption

Acceptable amount of  
**decentralization**

# SPV

privacy?



# SPV privacy

BIP37 - Bloom Filters 

- ✓ Low bandwidth
- ✓ Can filter mempool
- ✗ No privacy
- ✗ Personal filtering  
(incentive model)

Electrum SPV 

- ✓ Low bandwidth
- ✓ MITM protection through SSL
- ✗ (No privacy)
- ✗ Personal filtering  
(incentive model)

BIP158 - Compact Block Filters 

- „more“ bandwidth
- ✓ Privacy (?)
- ✓ Widely useful filter structures
- ✓ Committable through soft-fork
- ✗ not (widely) deployed
- ✗ **no (useful) technique to filter mempool**

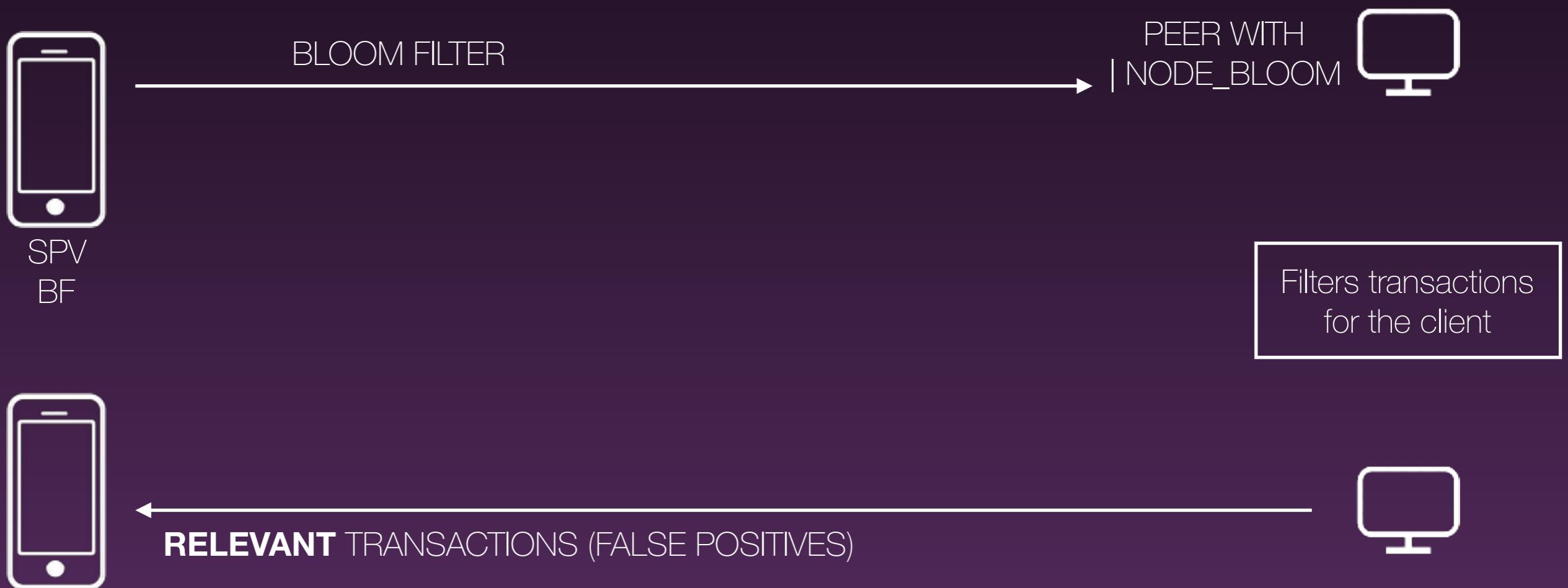
# SPV privacy

Full block SPV 

- ✗ „high“ bandwidth costs
- ✓ Can „migrate“ to full validation
- ✓ Privacy
- ✓ Reduced consensus checks

# understanding **SPV** filtering

## **BIP37** - Bloom Filters



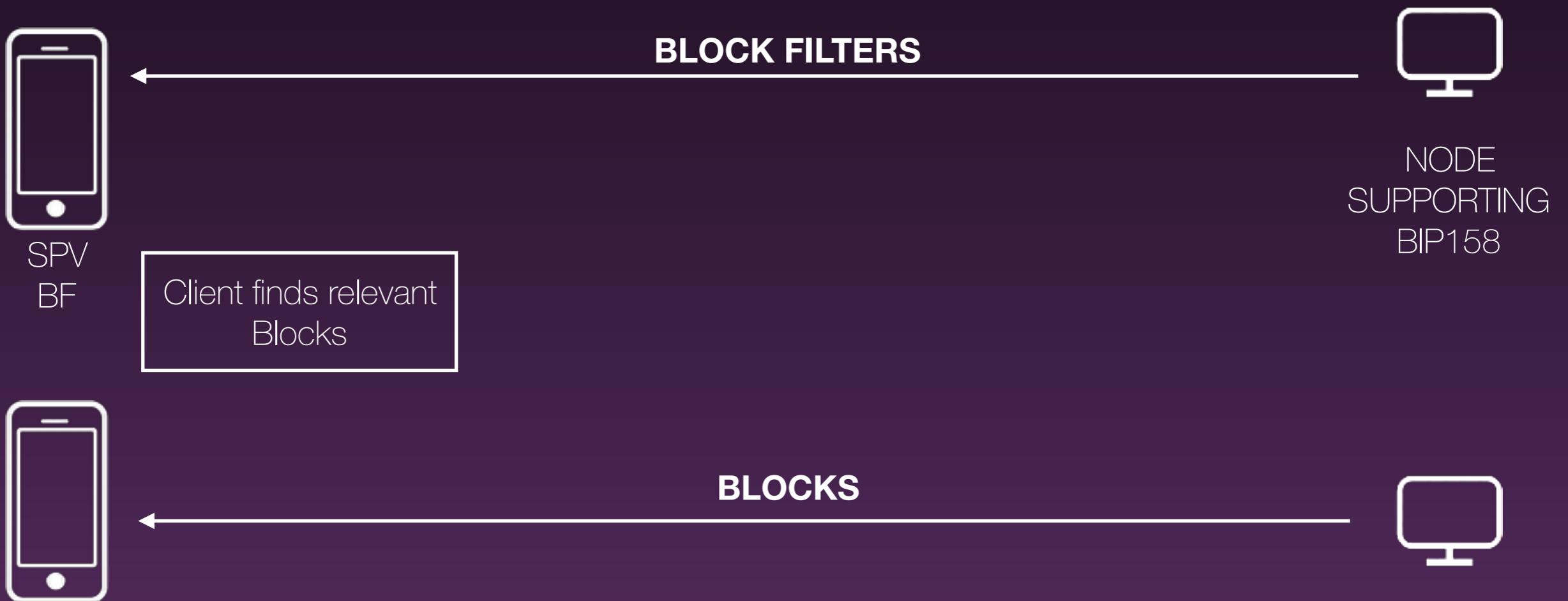
# understanding **SPV** filtering

## BIP37 - Bloom Filters



# understanding **SPV** filtering

## **BIP158** - Client Side Filtering



# understanding **SPV** filtering

## **BIP158** - Client Side Filtering

144 blocks  $\sim=$  144MB

—

Filtersize: ~2%

**1 day** = ~2.88MB  
**30 days** = ~86.4 MB

**7 days** = ~20.16 MB  
**90 days** = ~259.2 MB

# understanding **SPV** filtering **FULL BLOCK / HYBRID**

144 blocks  $\sim=$  144MB

**1 day** = ~144MB  
**30 days** = ~4.32 **GB**

**7 days** = ~1'008 MB  
**90 days** = ~12.96 **GB**

Resource costs



Centralized validation solutions



SPV



Electrum (SPV)

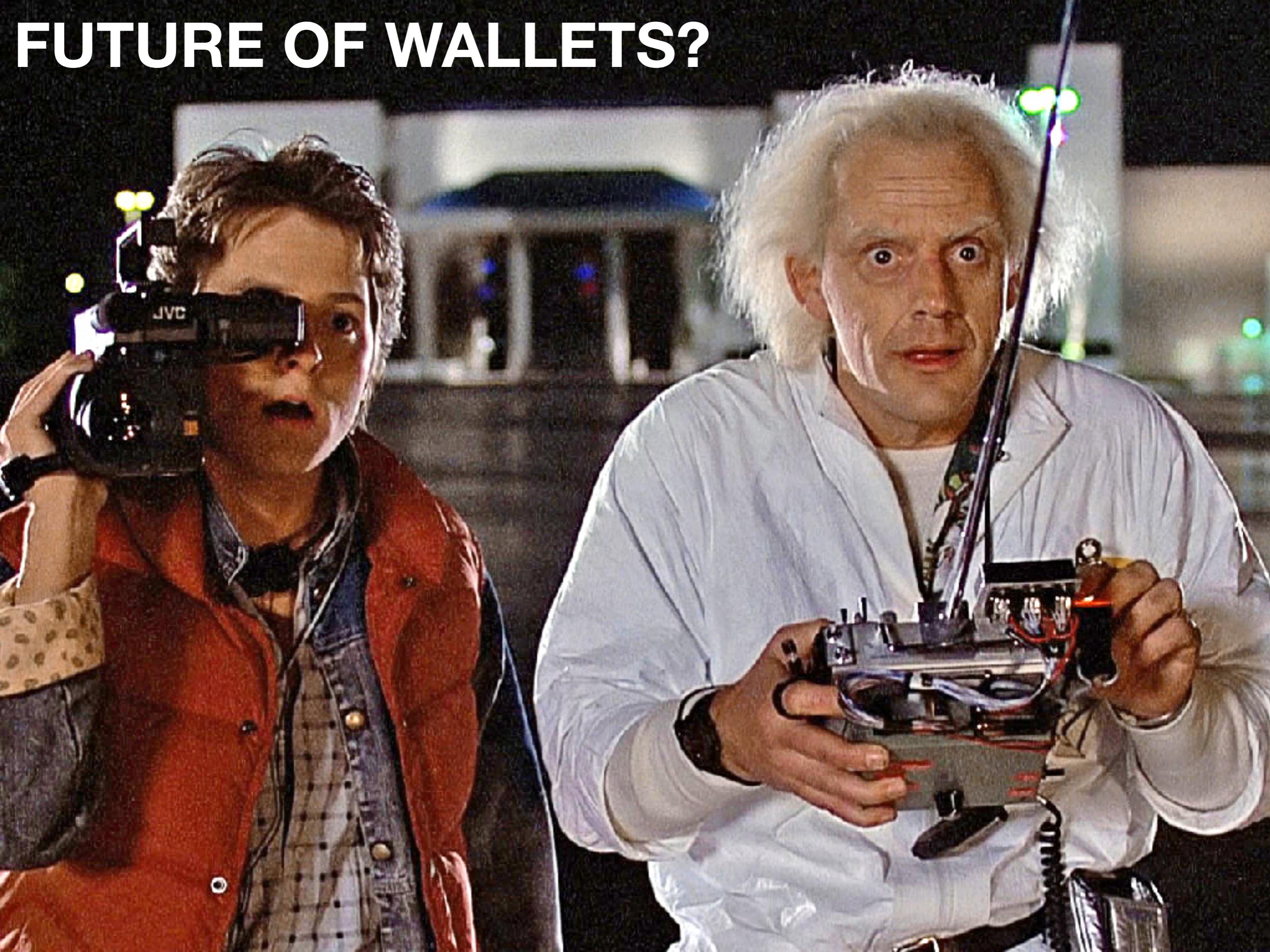


Core / full node

Decentralization

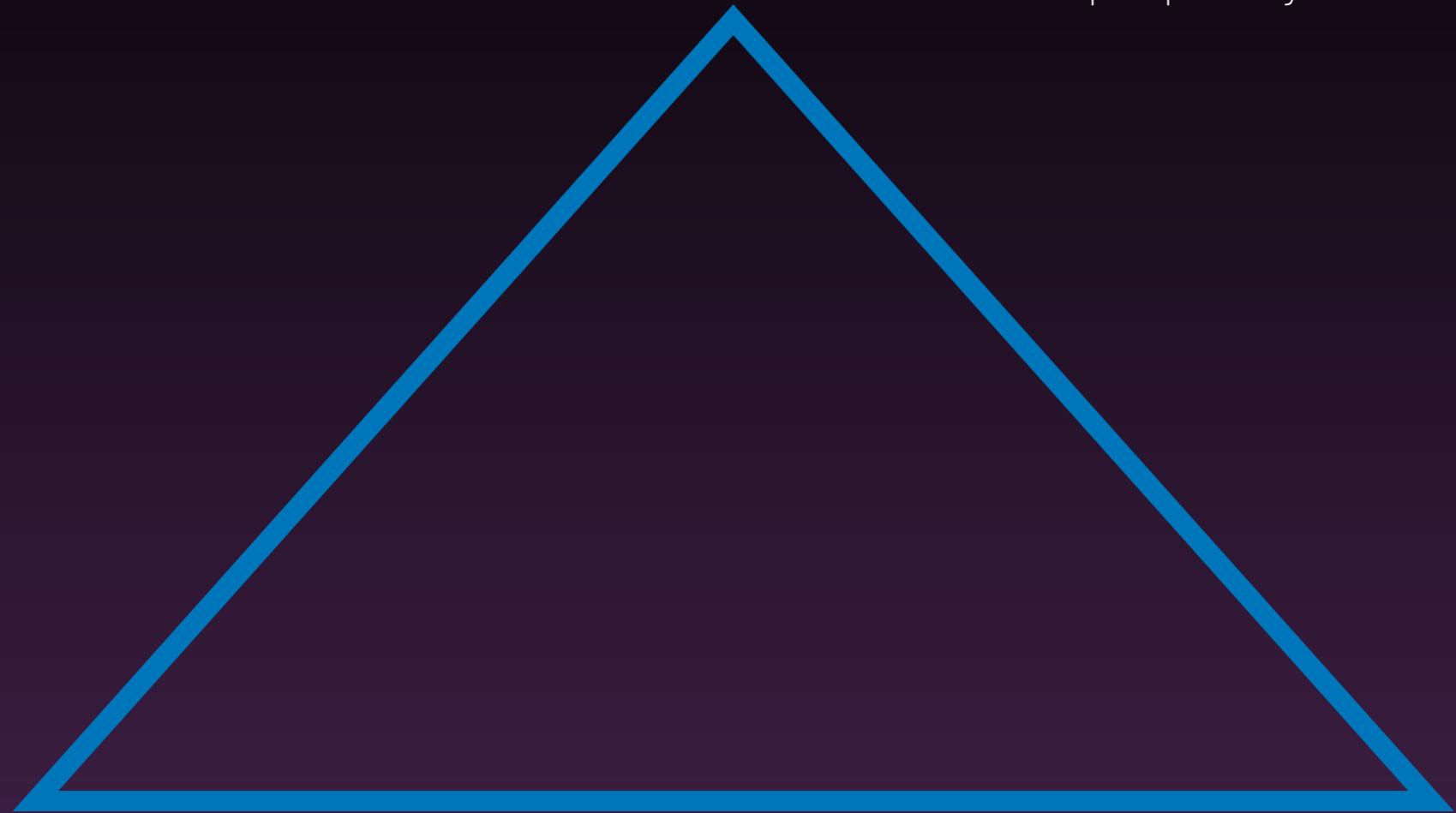


# FUTURE OF WALLETS?



# ✓ Privacy

Transaction / scripts privacy



# ✓ Security

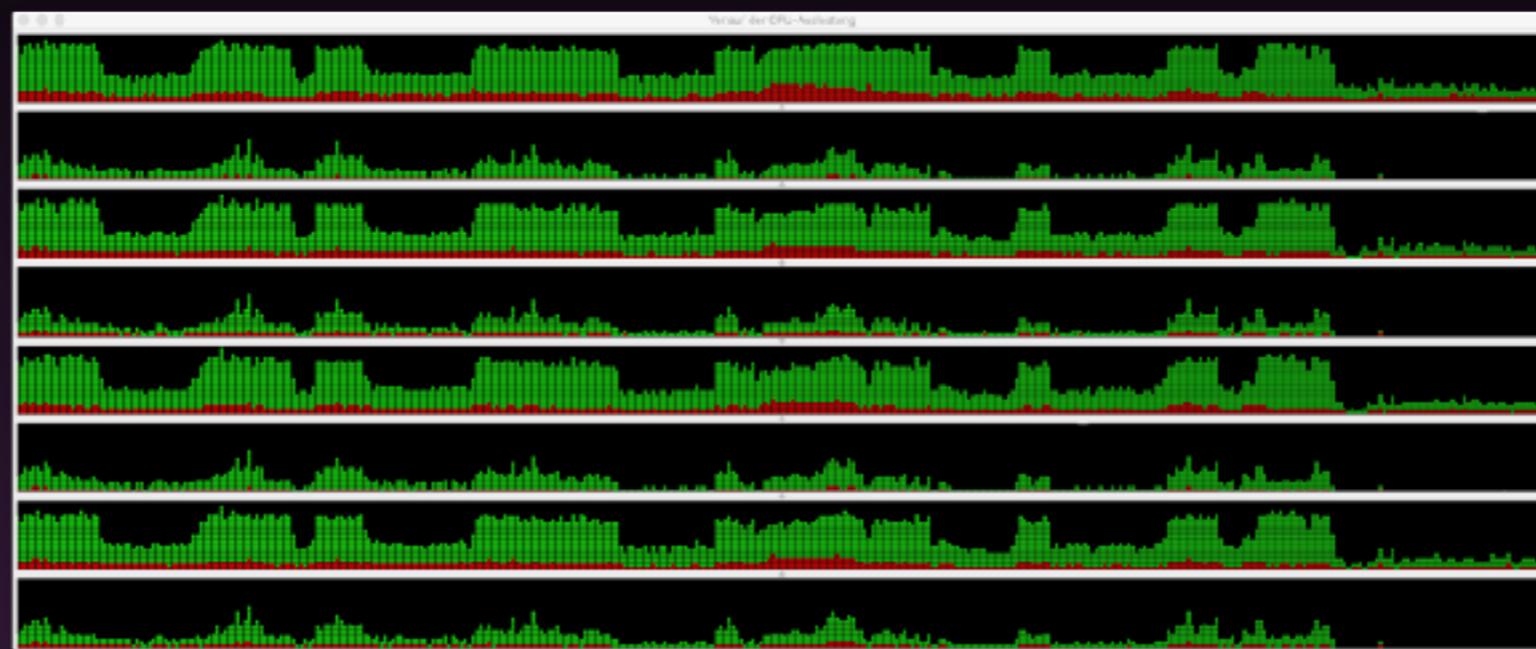
Keystorage  
Cold-Storage

# ✓ Trust

No-trust required  
Chain-Validation

**Consensus / p2p**

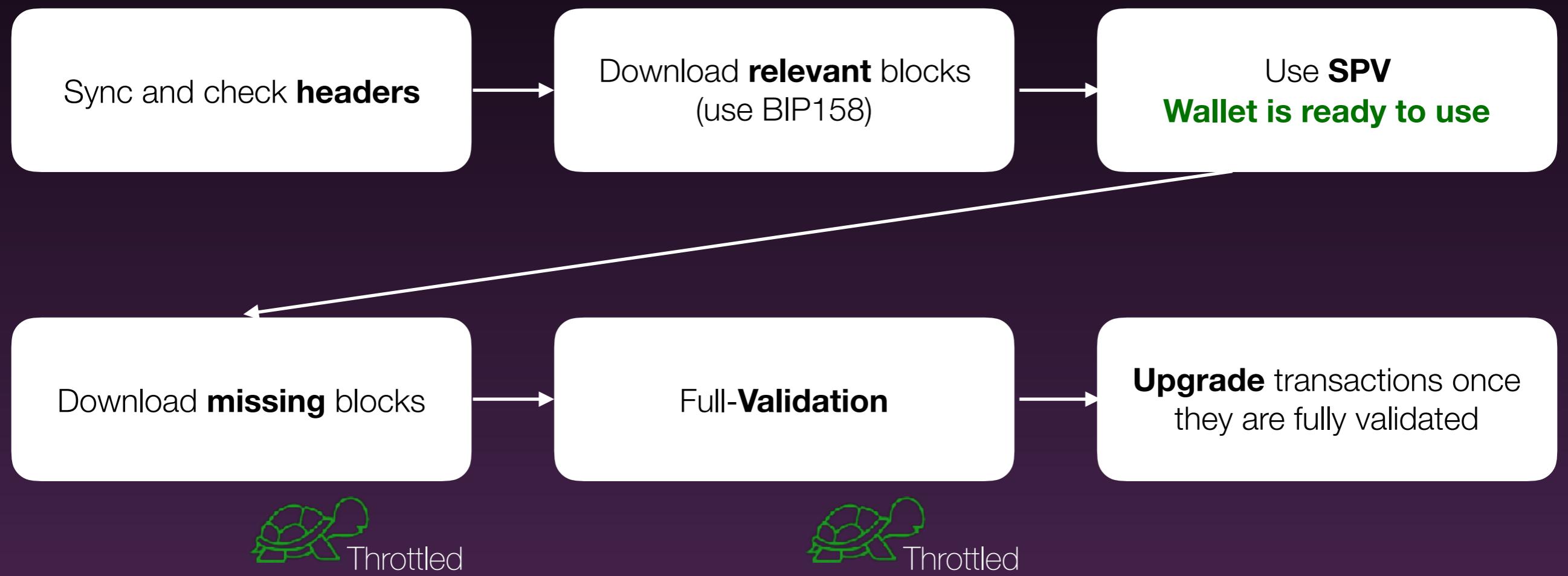
Catching up a **month** of blocks  
**(45min; consumer system)**



Acceptable CPU / memory rates once in-sync

Prozessname	% CPU	Physikal. Speicher	CPU-Zeit	Threads
Slack Helper	0.4	156.9 MB	2:11.66	5
Bitcoin Core	0.4	1.13 GB	1:10:15.74	18
Slack Helper	0.3	315.3 MB	9:22.91	20
Slack Helper	0.2	155.2 MB	25.60	20
Slack	0.2	86.4 MB	4:32.27	30
Slack Helper	0.1	74.4 MB	24.13	20
Slack Helper	0.0	61.2 MB	7.04	19

# Hybrid SPV



**Privacy and self-  
verification (no trust)**  
is not an opt-in model

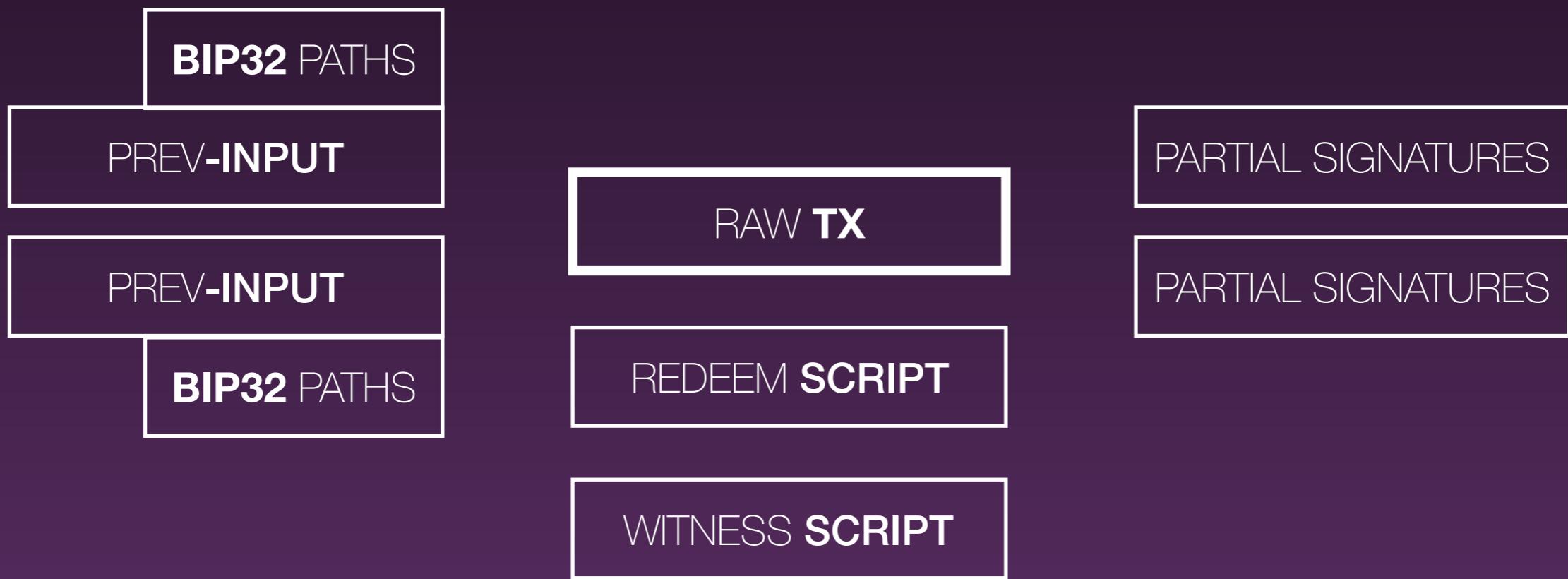
Keep users **away** from  
**trusted third parties**

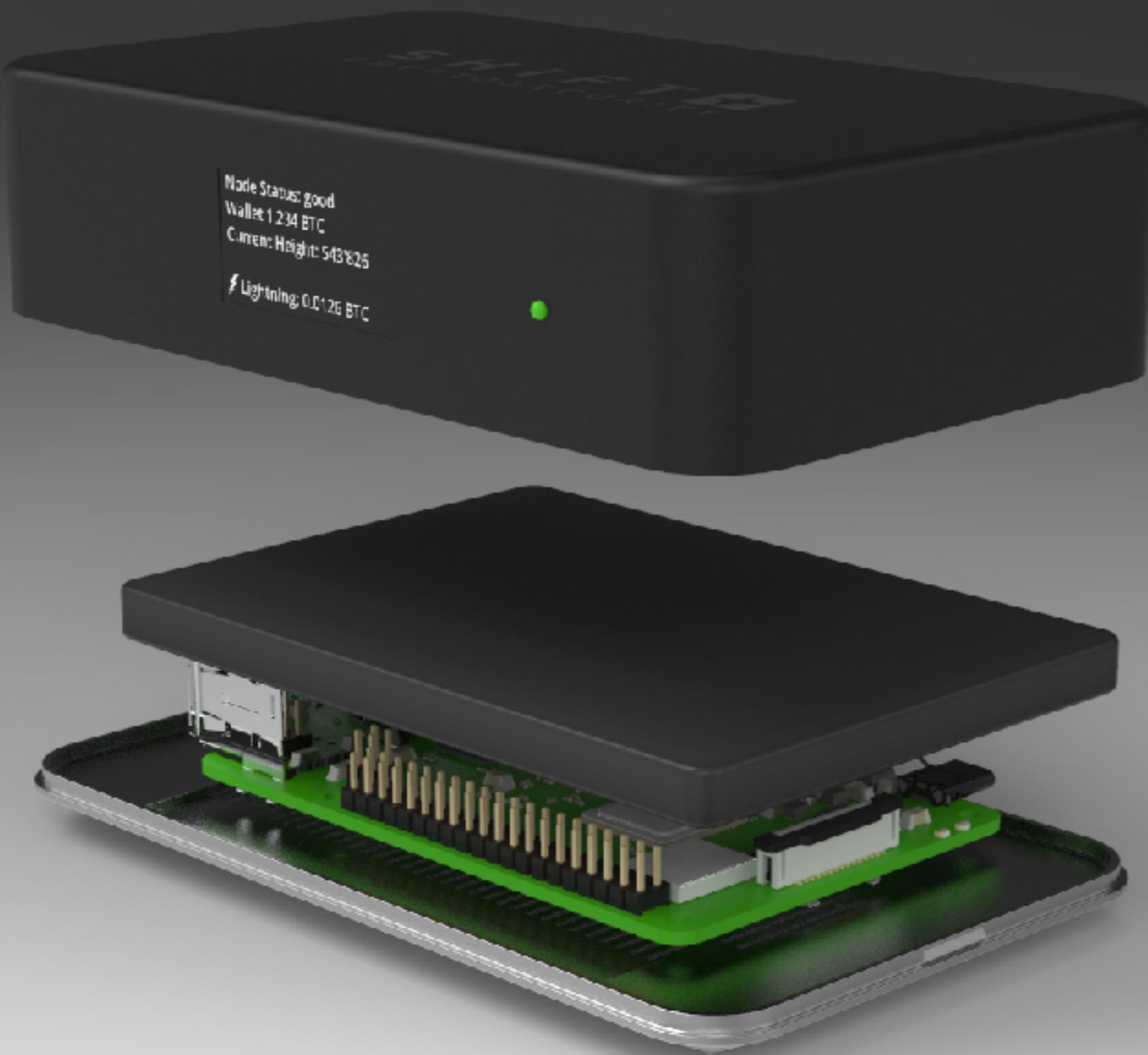
# UTXO set commitments

```
{  
  "height": 530075,  
  "bestblock": "000000000000000000002fe10af166937d506ece7fad4381fda6cb86e9e1404be2",  
  "transactions": 24567998,  
  "txouts": 50460119,  
  "bogosize": 3798659787,  
  "hash_serialized_2":  
    "090c1276fe42f98246840fabac42dfa0e8b89b428f81ab16d53d69ae669bec4b",  
  "disk_size": 2921681465,  
  "total_amount": 17125767.33401612  
}
```

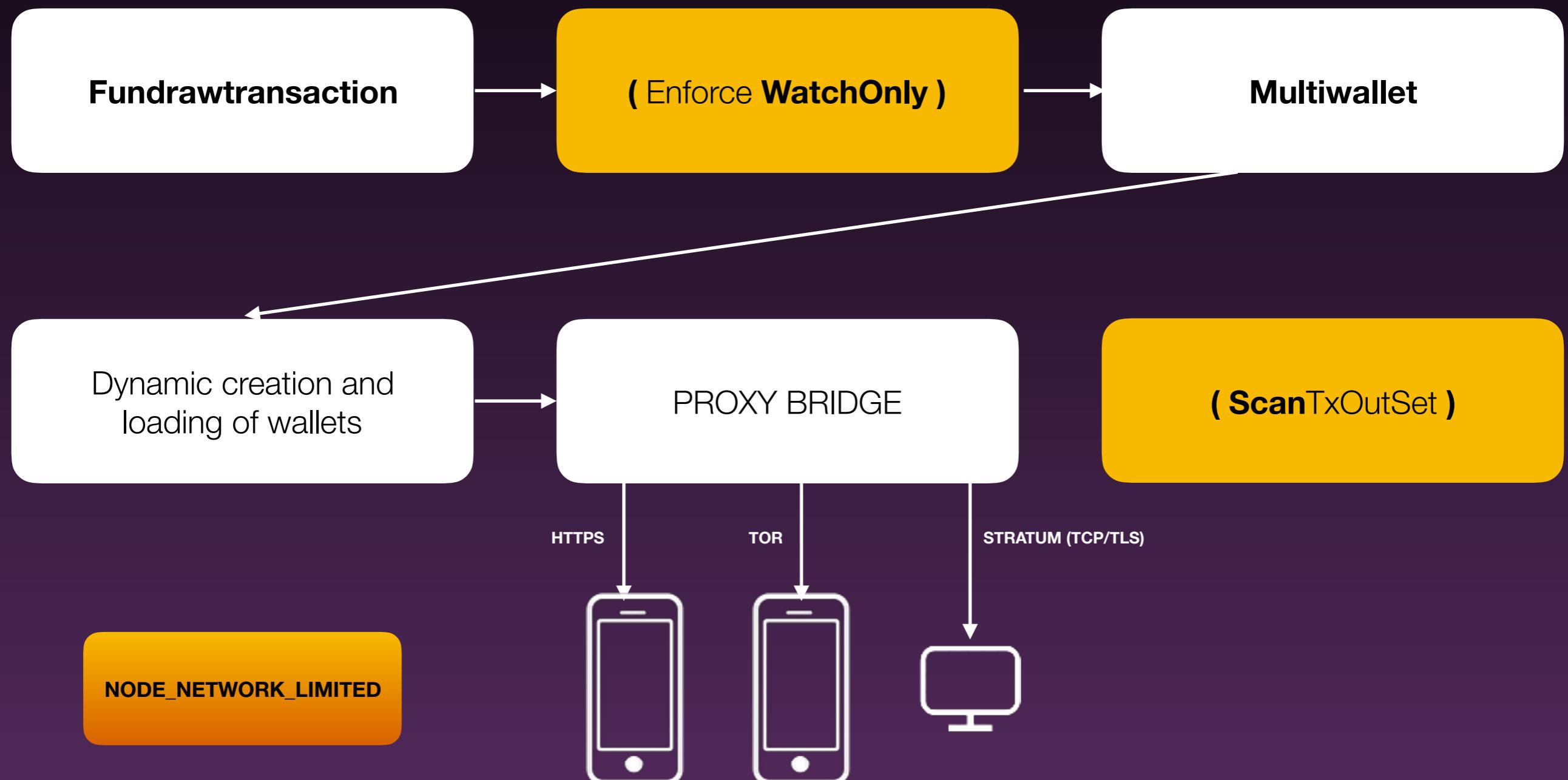
# BIP174

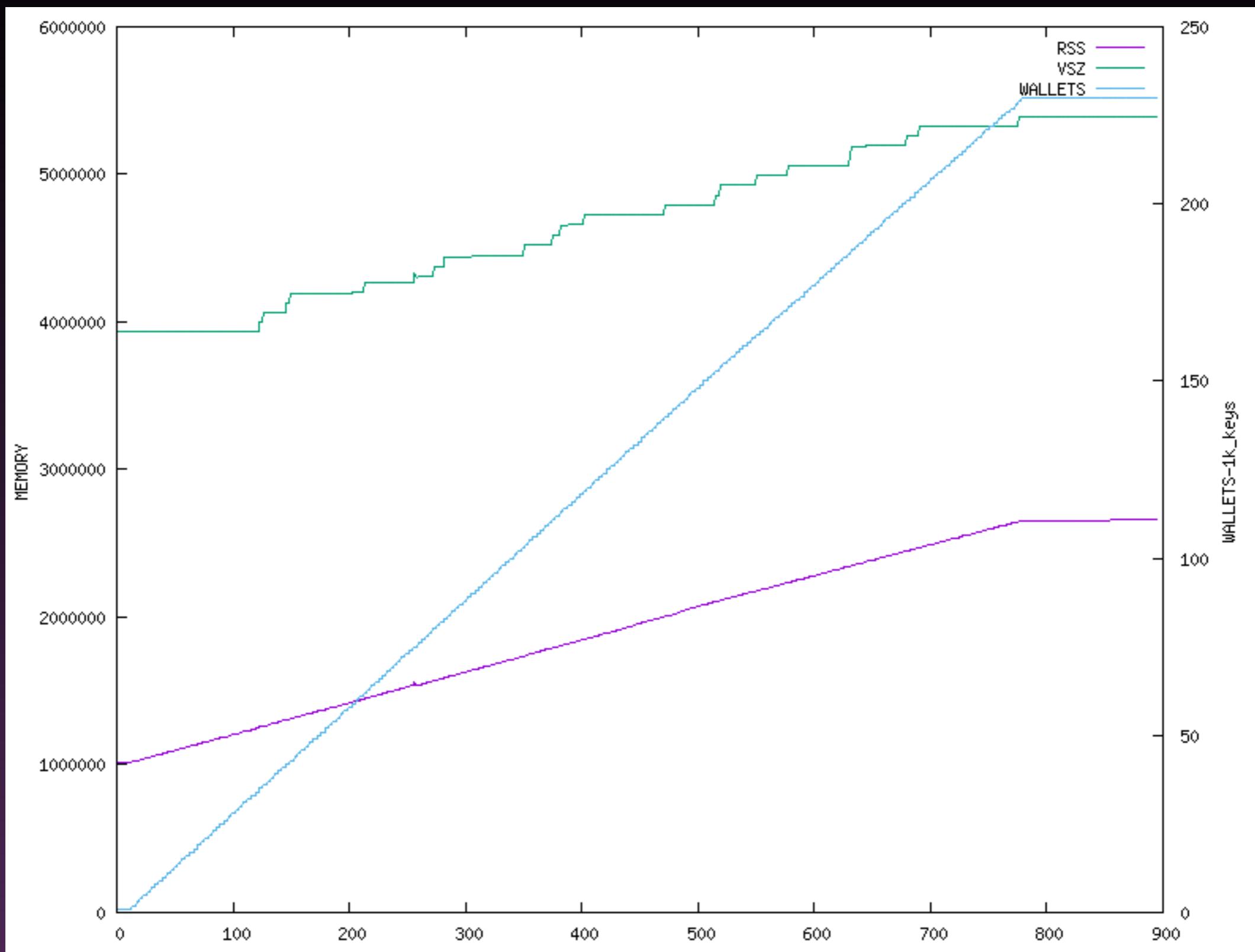
Partially **S**igned **B**itcoin **T**ransaction Format  
(PSBT)





# BitcoinCore 0.17 PRUNED





 BitcoinCore +  =

Chris Belcher's

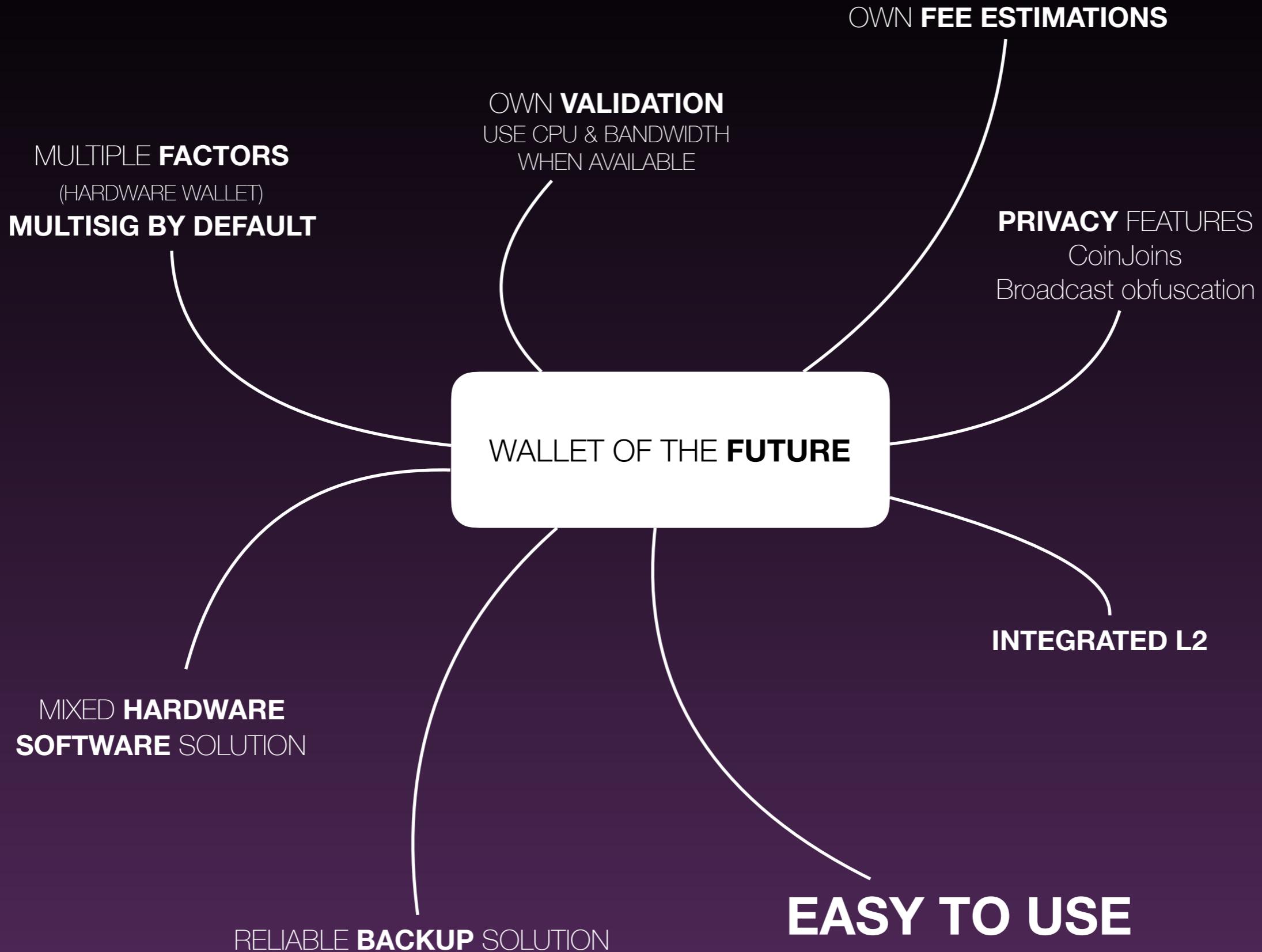
# Personal **Electrum Server**

# JoinMarket Orderbook

18 orders found by 11 counterparties

BTC %

Type	Counterparty	Order ID	Fee	Miner Fee Contribution / BTC	Minimum Size / BTC	Maximum Size / BTC
Absolute Fee	J5CawiUMyqBc77Gk	0	0.00000202	0.00000000	0.00100000	0.38545544
Absolute Fee	J5A5GMk9VwzmJc9L	0	0.00000945	0.00000000	0.00027300	0.03788056
Absolute Fee	J57RQ7Mfr3MSXq2g	1	0.00000954	0.00000000	0.07358091	0.17257353
Absolute Fee	J5A5GMk9VwzmJc9L	1	0.00001351	0.00000000	0.03788057	0.30446864
Absolute Fee	J592CTWwWhKm2YFR	0	0.00005000	0.00000000	0.10000000	0.22497598
Absolute Fee	J57RQ7Mfr3MSXq2g	0	0.00005420	0.00000000	0.00027300	0.07358090
Relative Fee	J5A5GMk9VwzmJc9L	2	0.00010963%	0.00000000	0.30446865	1.46144594
Relative Fee	J57RQ7Mfr3MSXq2g	2	0.00026723%	0.00000000	0.17257354	0.60258490
Relative Fee	J59SSNZRM7NnoWgp	0	0.0007%	0.00001000	2.14285714	4.16541954
Relative Fee	J5EwDvIkb9WdLvkn	0	0.0045%	0.00000000	0.01000000	1.35909720
Relative Fee	J59a2ajX6X1Dc8u7	1	0.005%	0.00000000	0.01000000	9.99999999
Relative Fee	J59eV89JyUXFxad1	0	0.009%	0.00001000	0.16666666	17.16377123
Relative Fee	J5Dq4MHrd7vJupRw	0	0.018%	0.00000000	0.00300000	1.88143549
Relative Fee	J5EopHErZ6G6XqFs	0	0.02%	0.00001000	0.07500000	6.82622258
Relative Fee	J5GcWFMkgsue2HR6	0	0.02%	0.00001000	0.07500000	0.20283752
Relative Fee	J59a2ajX6X1Dc8u7	2	0.02%	0.00000000	10.00000000	29.99999999



# Thanks, Q&A?

---

dev@jonasschnelli.ch

PGP: CA1A2908DCE2F13074C62CDE1EB776BB03C7922D



\_jonasschnelli\_



**github.com/jonasschnelli**