Descriptor Wallet: library and command-line

Dr Maxim Orlovsky,
Pandora Core AG

Call agenda

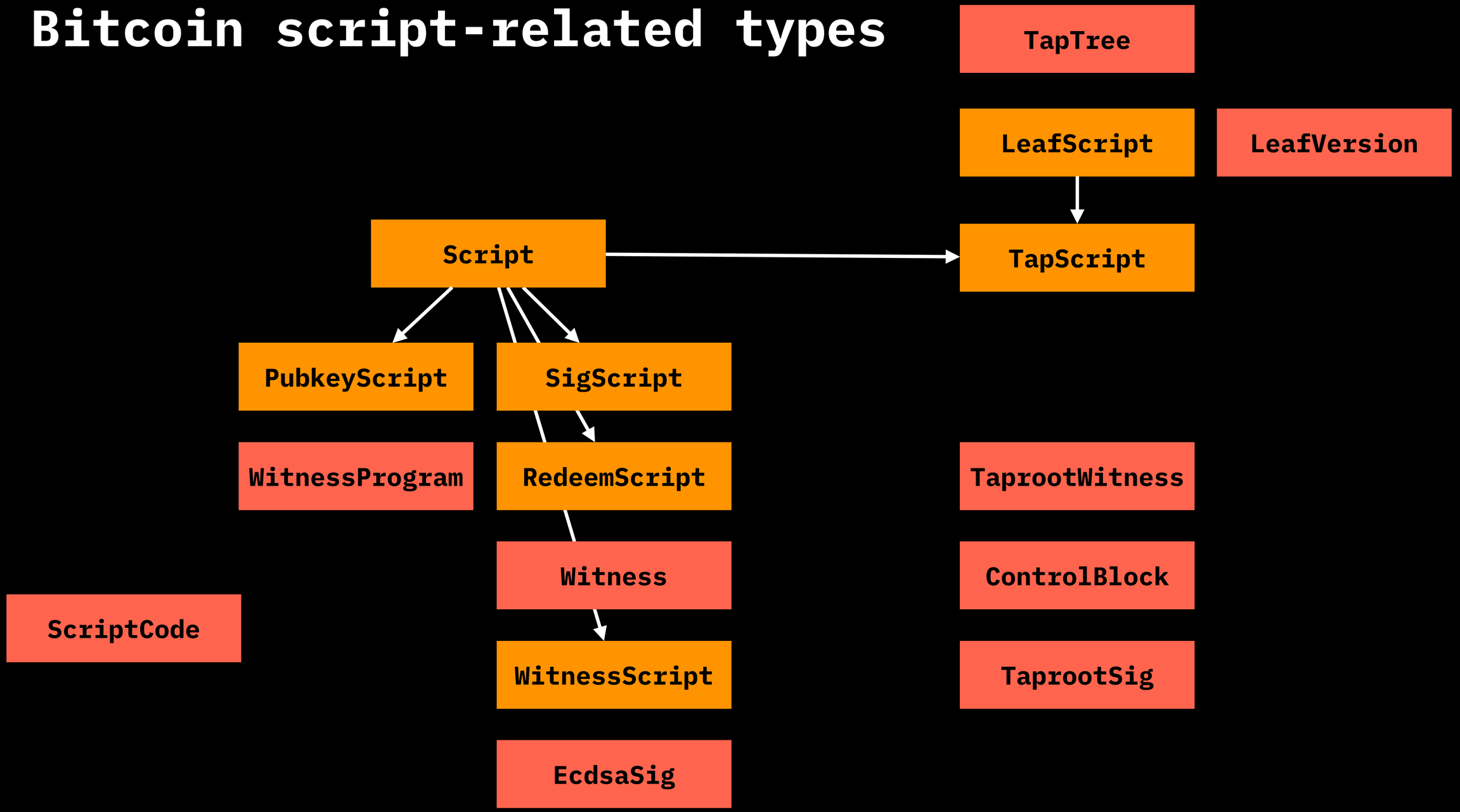
- Descriptor wallet intro
- Descriptor wallet demo
- Wallet challenges related to RGB & single-use-seals
- P2C & S2C commitment discussion

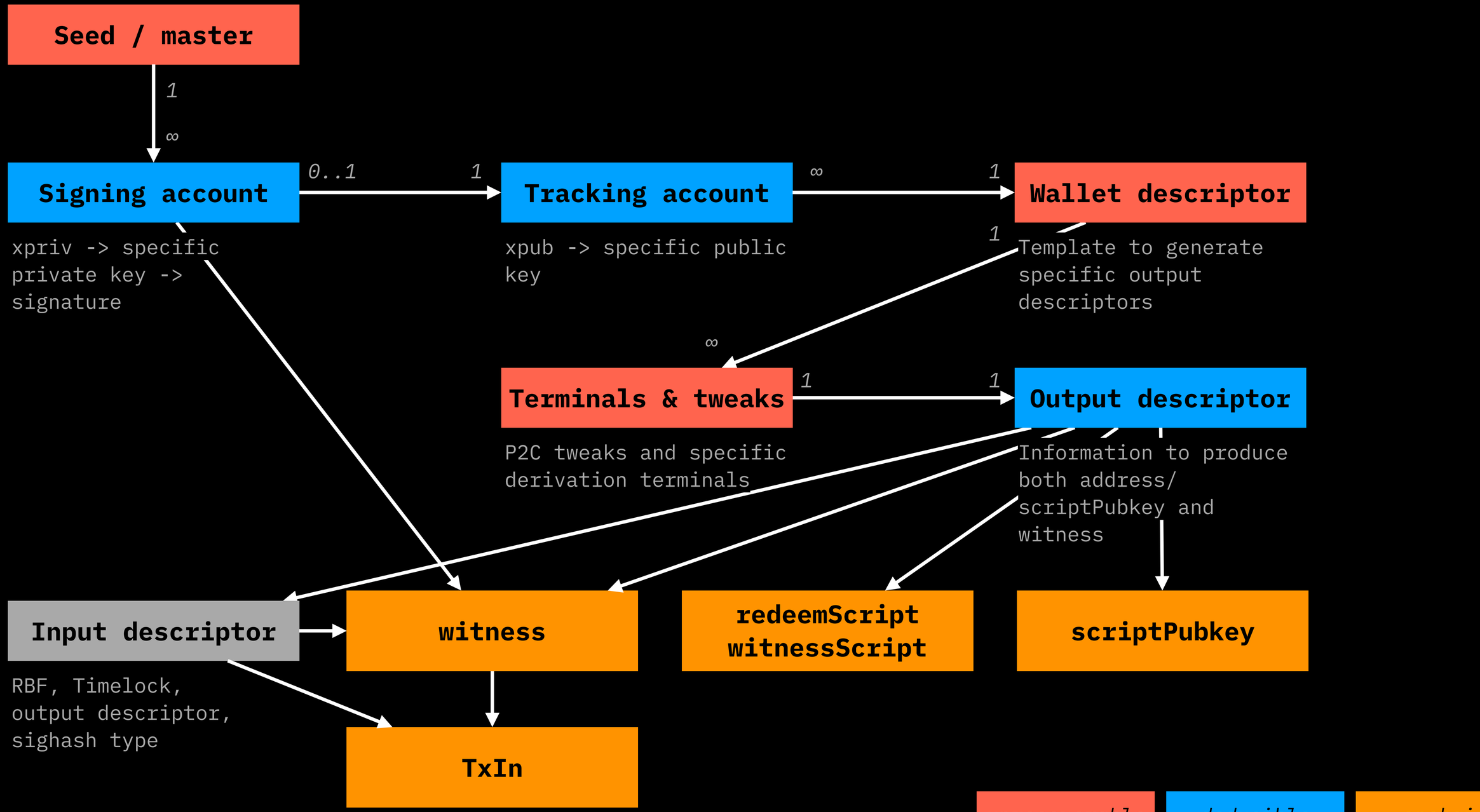
Descriptor wallet

- Separating cold and hot parts
- Allowing P2C / S2C commitments & single-use-seals
- Foundation for LNP Node, RGB Node, MyCitadel Wallet
- Pro command-line tool to do arbitrary-complex magic with bitcoin

	Descriptor Wallet	BitcoinDevKit	Green dev kit (GDK)	Libwally	All other wallet libraries
Full hot/cold wallet isolation	✅	🚫	🚫	🚫	🚫
Output descriptors	✅	✅	🚫	🚫	🚫
Input descriptors	✅	🚫	🚫	🚫	🚫
Miniscript	✅	✅	🚫	🚫	🚫
Taproot base	✅	✅	🚫	🚫	🚫
Taproot MuSig2	🚫	🚫	🚫	🚫	🚫
TapScript	✅	✅	🚫	🚫	🚫
P2C tweaks	✅	🚫	🚫	🚫	🚫
Custom sighashes	✅	🚫	🚫	🚫	🚫
Lightning compatibility	✅	🚫	🚫	🚫	🚫

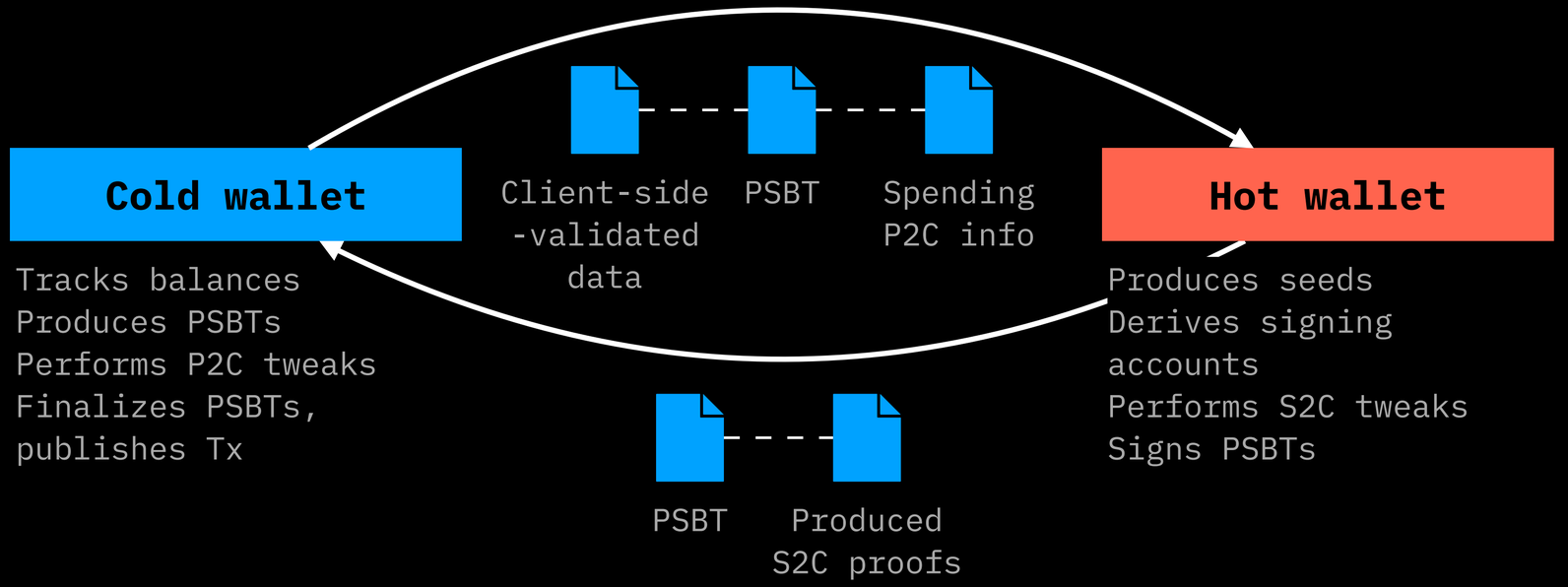
Bitcoin script-related types




Seed / master
1
∞
Signing account
0..1
1
Tracking account
∞
1
Wallet descriptor
xpriv -> specific private key -> signature
xpub -> specific public key
1
Template to generate specific output descriptors
∞
Terminals & tweaks
1
1
Output descriptor
P2C tweaks and specific derivation terminals
Information to produce both address/ scriptPubkey and witness
Input descriptor
witness
redeemScript witnessScript
scriptPubkey
RBF, Timelock, output descriptor, sighash type
TxIn
unrecoverable
deducible
onchain

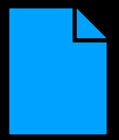
Demo

Cold-hot wallet distinction



Contact identities

Tracking accounts



Wallet descriptor

Tweak Information

Client-side-validation state commitments

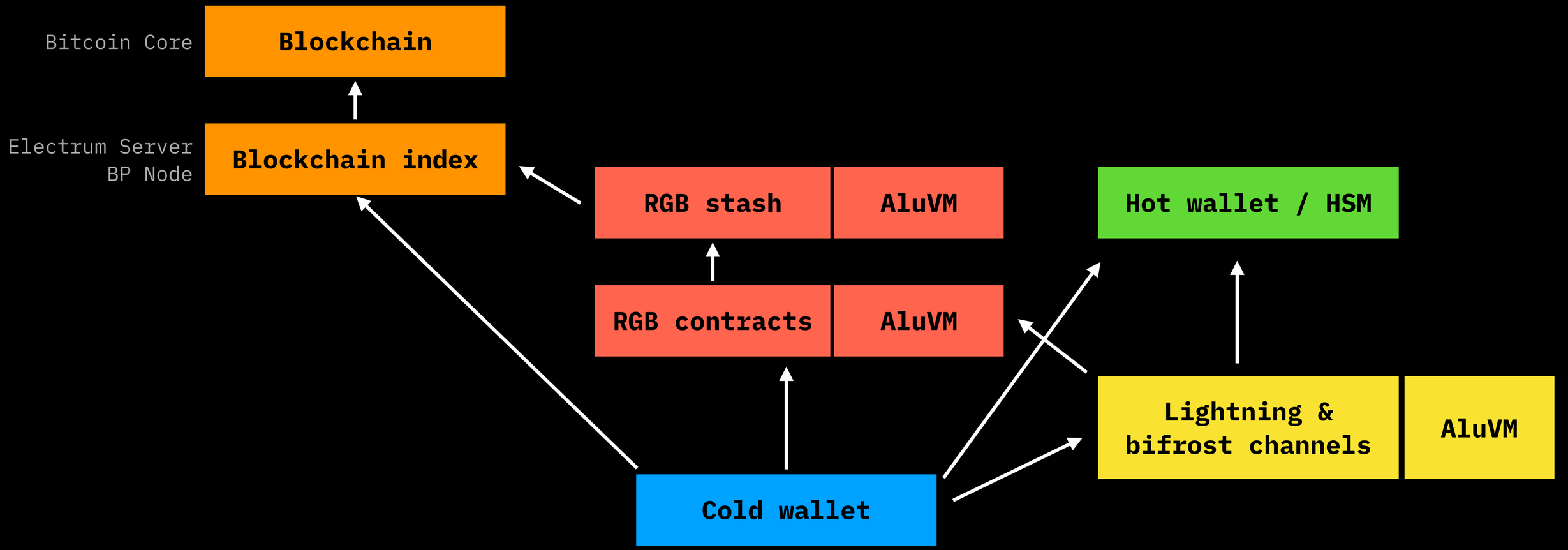
Signing accounts

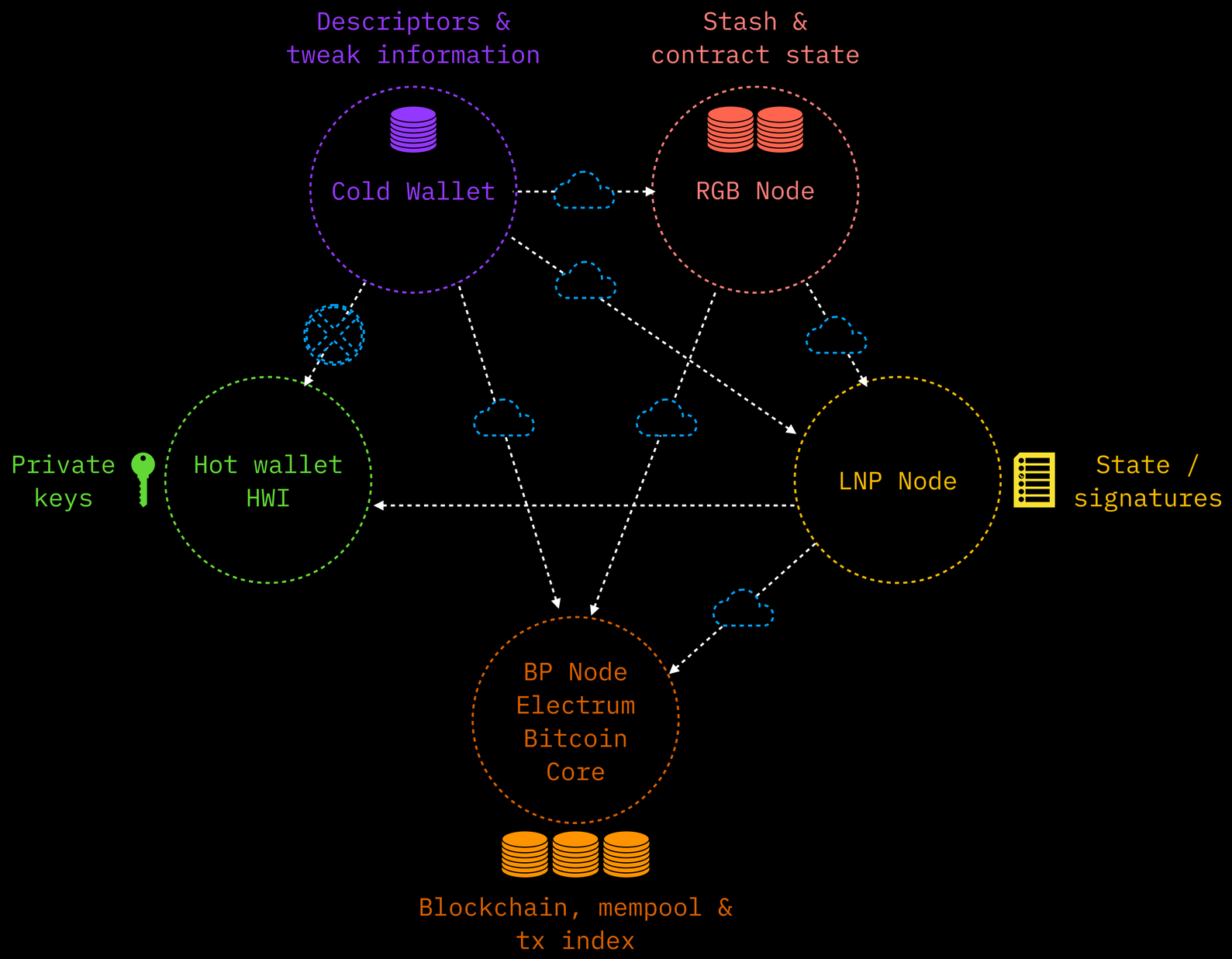
Wallet data structures

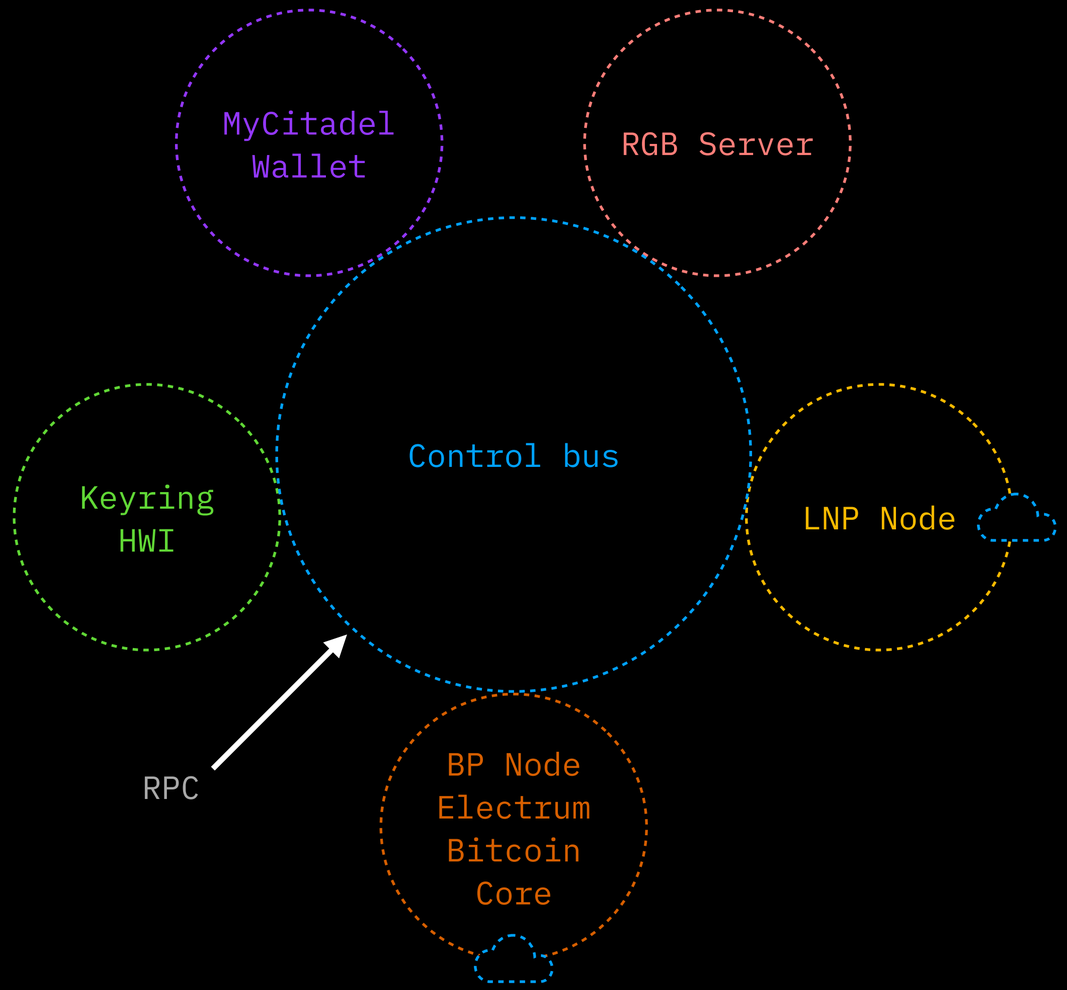
- Seed:
  mnemonic (12..24 words)
- Signing account:
  m=[f1493701]/44’/0’=[xpriv…]
- Tracking account (TA):
  m=[f1493701]/44’/0’=[xpub…]//**
- Wallet descriptor:
  tr(musig(TA1, TA2), {
  and_v(older(5),TA1),
  and_v(older(5),agg(TA1, TA1)
  })
- Terminals & tweaks:
  1/1665~[key_fingerprint]:tweak_hash
- Output descriptor:
  wsh(
  sortedmulti([fp1]~hash, [fp2])
)@1/1665
- Input descriptor:
  wsh(
  sortedmulti([fp1]~hash, [fp2])
)@1/1665@rbf#SINGLE

Input descriptor example

- txid:vout /1/167 [deadbeef]+hash rbf SIGHASH_ALL

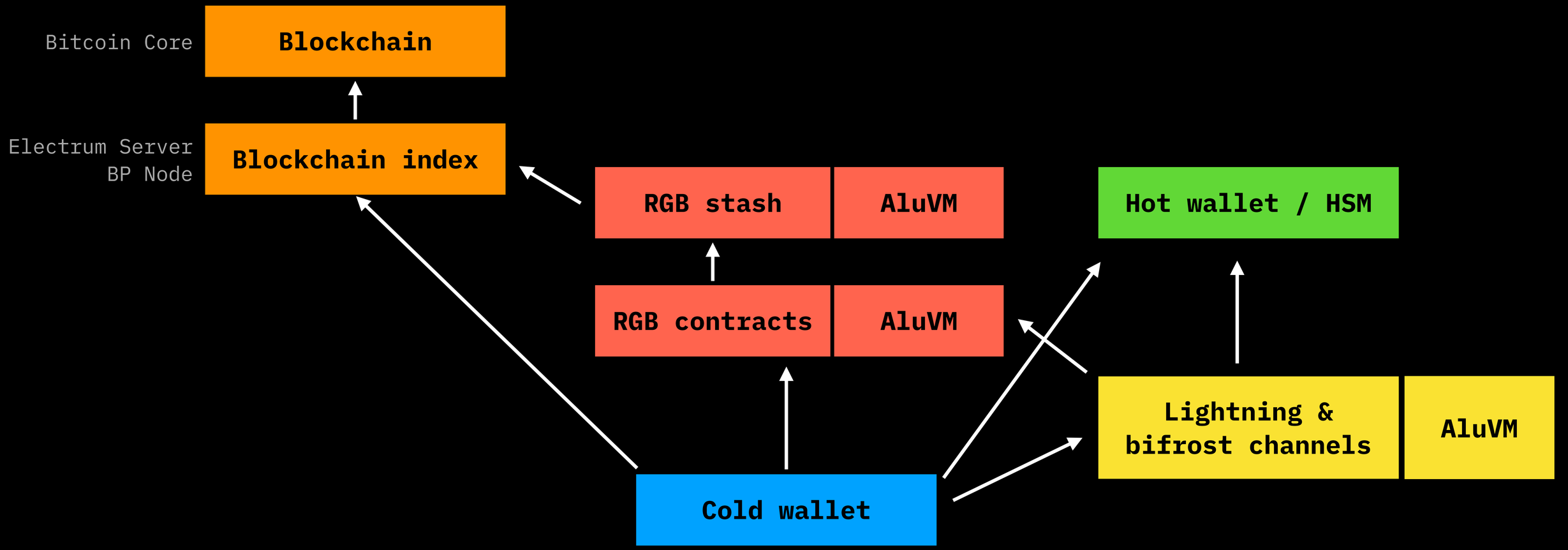

Bitcoin Core
Blockchain
Electrum Server
BP Node
Blockchain index
RGB stash
AluVM
RGB contracts
AluVM
Hot wallet / HSM
Lightning &
bifrost channels
AluVM
Cold wallet

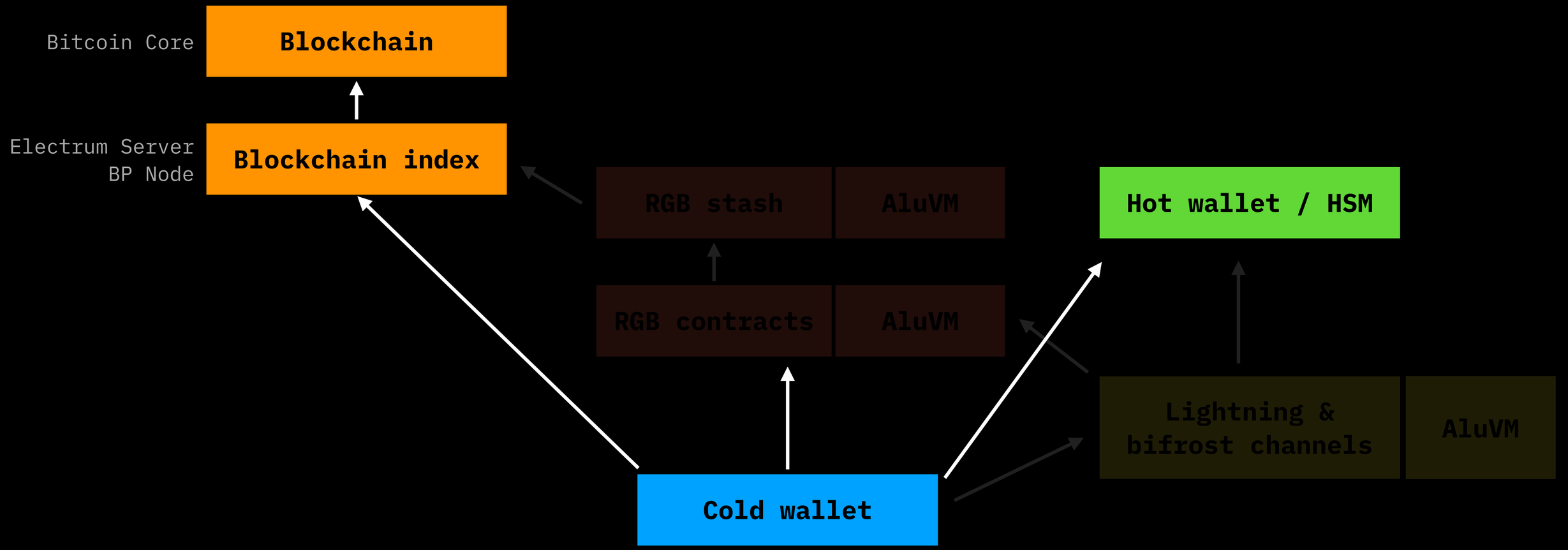

Descriptors &
tweak information
Stash &
contract state
Cold Wallet
RGB Node
Private
keys
Hot wallet
HWI
LNP Node
State /
signatures
BP Node
Electrum
Bitcoin
Core
Blockchain, mempool &
tx index

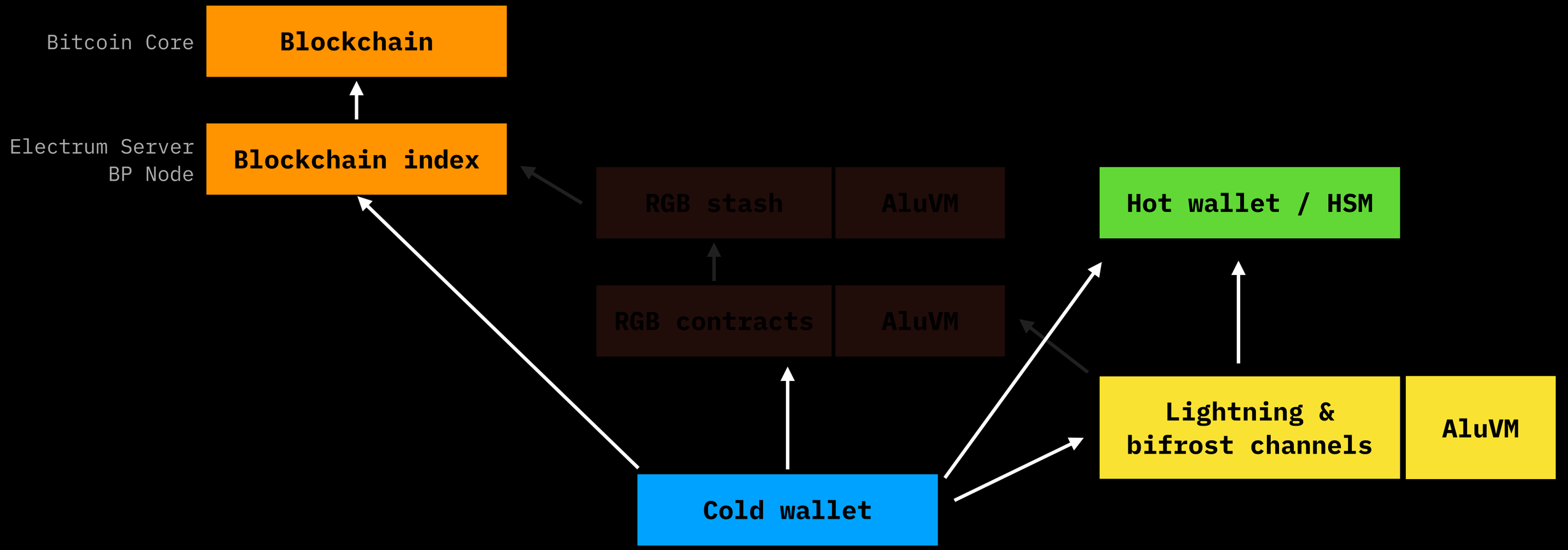

MyCitadel
Wallet
RGB Server
Control bus
Keyring
HWI
LNP Node
RPC
BP Node
Electrum
Bitcoin
Core

LNP: Protocols and Node Demo

Dr Maxim Orlovsky,
Pandora Core AG


Bitcoin Core
Blockchain
Electrum Server
BP Node
Blockchain index
RGB stash
AluVM
RGB contracts
AluVM
Hot wallet / HSM
Lightning &
bifrost channels
AluVM
Cold wallet


Bitcoin Core
Blockchain
Electrum Server
BP Node
Blockchain index
RGB stash
AluVM
RGB contracts
AluVM
Hot wallet / HSM
Lightning &
bifrost channels
AluVM
Cold wallet


Bitcoin Core
Blockchain
Electrum Server
BP Node
Blockchain index
RGB stash
AluVM
RGB contracts
AluVM
Hot wallet / HSM
Lightning &
bifrost channels
AluVM
Cold wallet

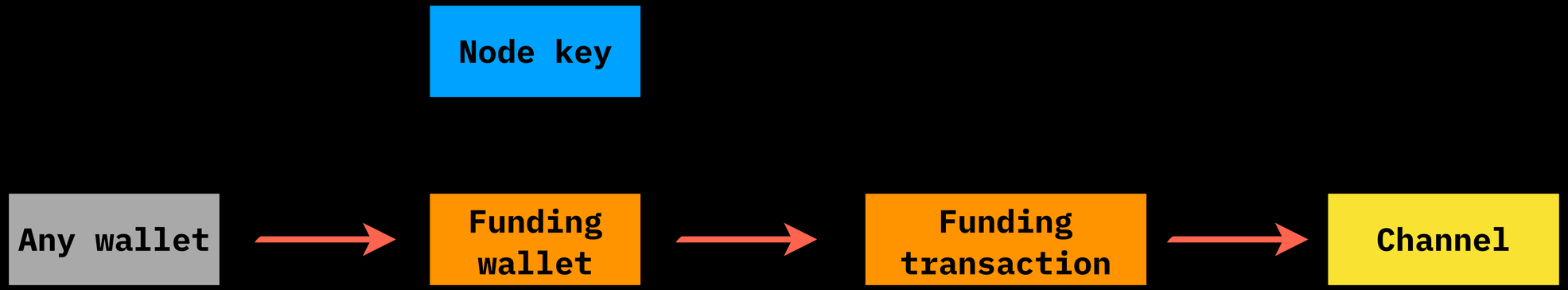
LNP, Bifrost & LNP Node roadmap

- **By the end 2021:** get LNP Node full operational with the **legacy Lightning network**
  (BOLT-standard compliant)

- **Jan-Feb 2022: Bifrost** implementation in LNP Node
  (completion of LNPBP standards)

Demo of lightning node initialization

Lightning funding wallet



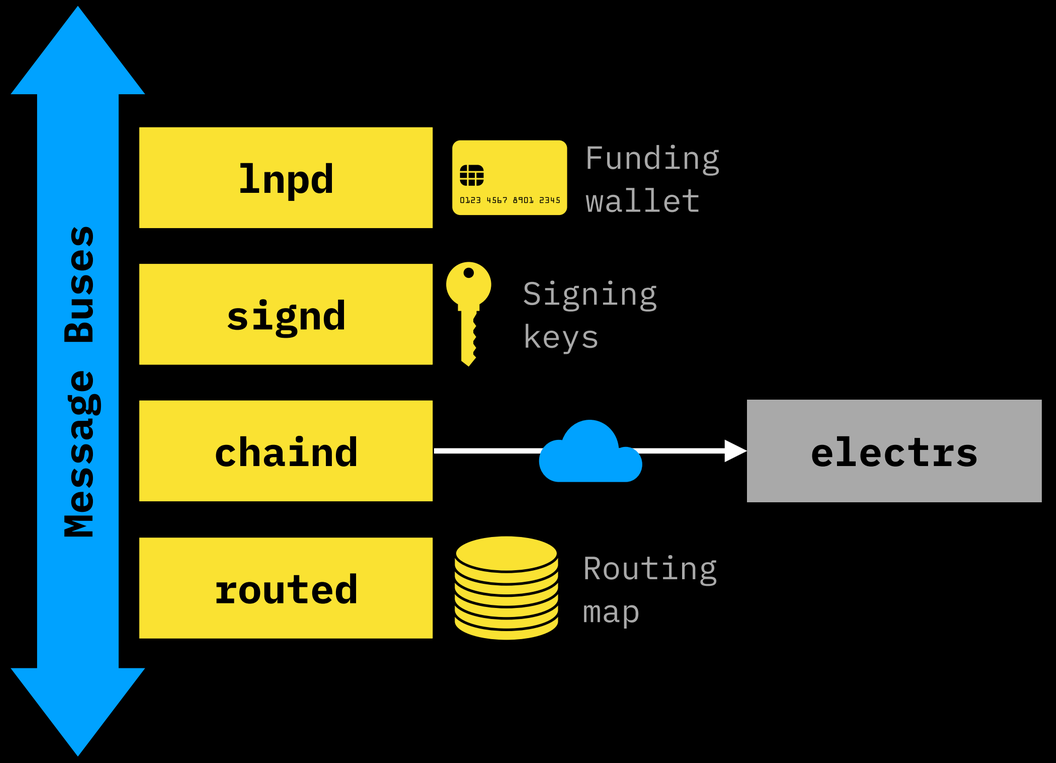
LNPBP-46: HD wallets for lightning

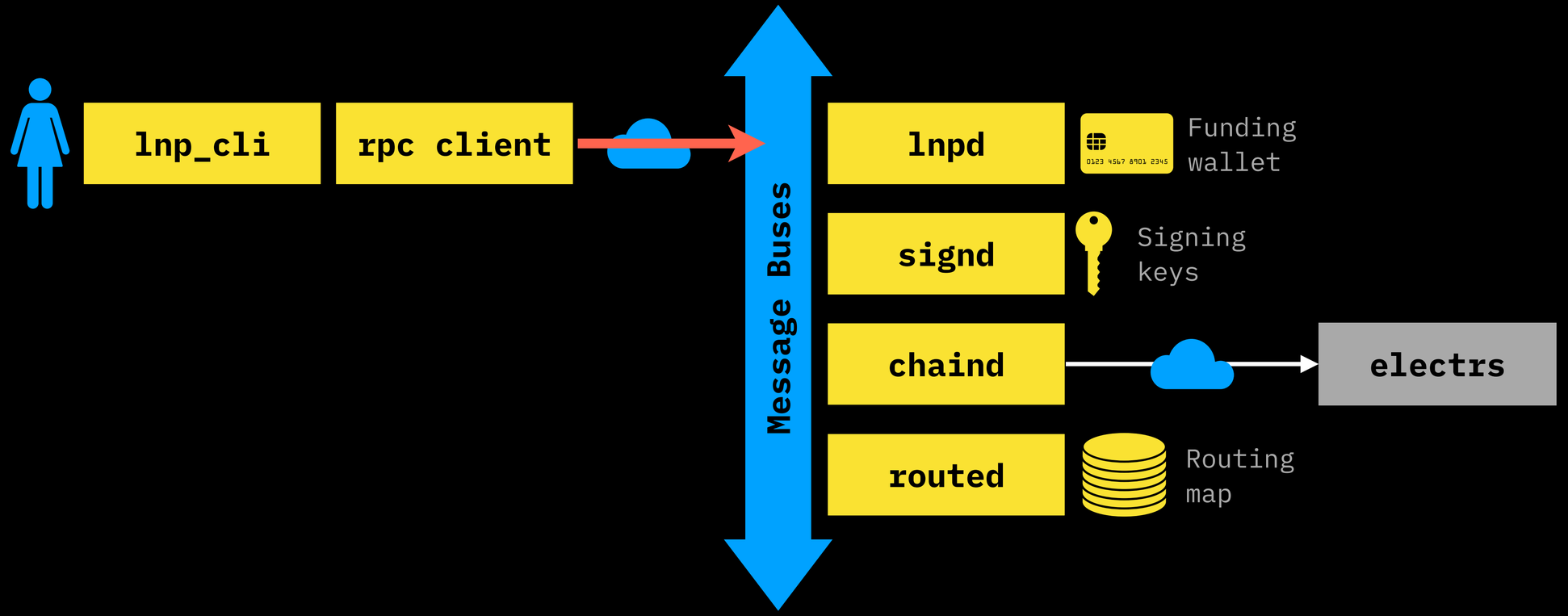
https://github.com/LNP-BP/LNPBPs/pull/120

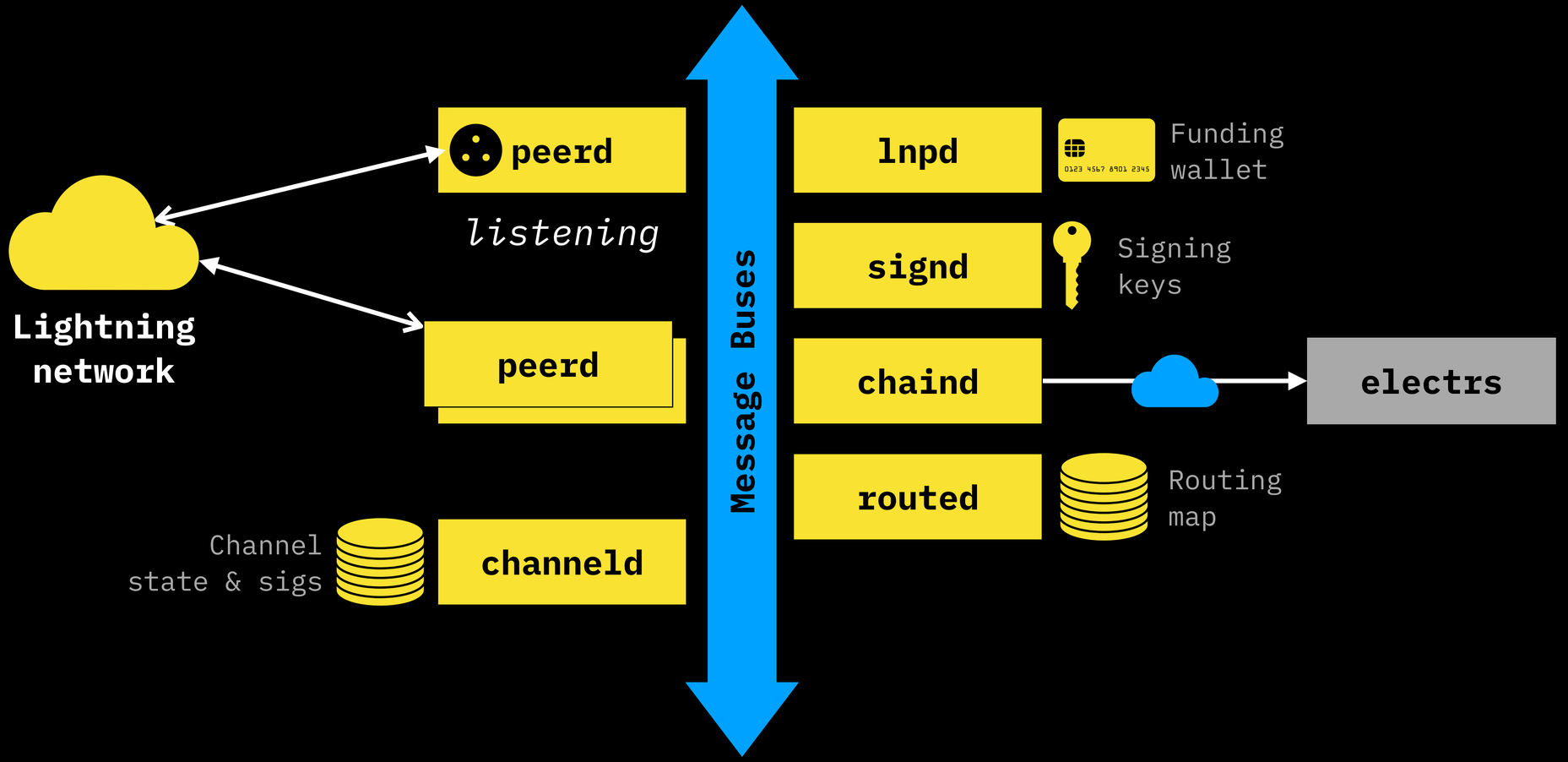
Concept: restore most of node from a single seed

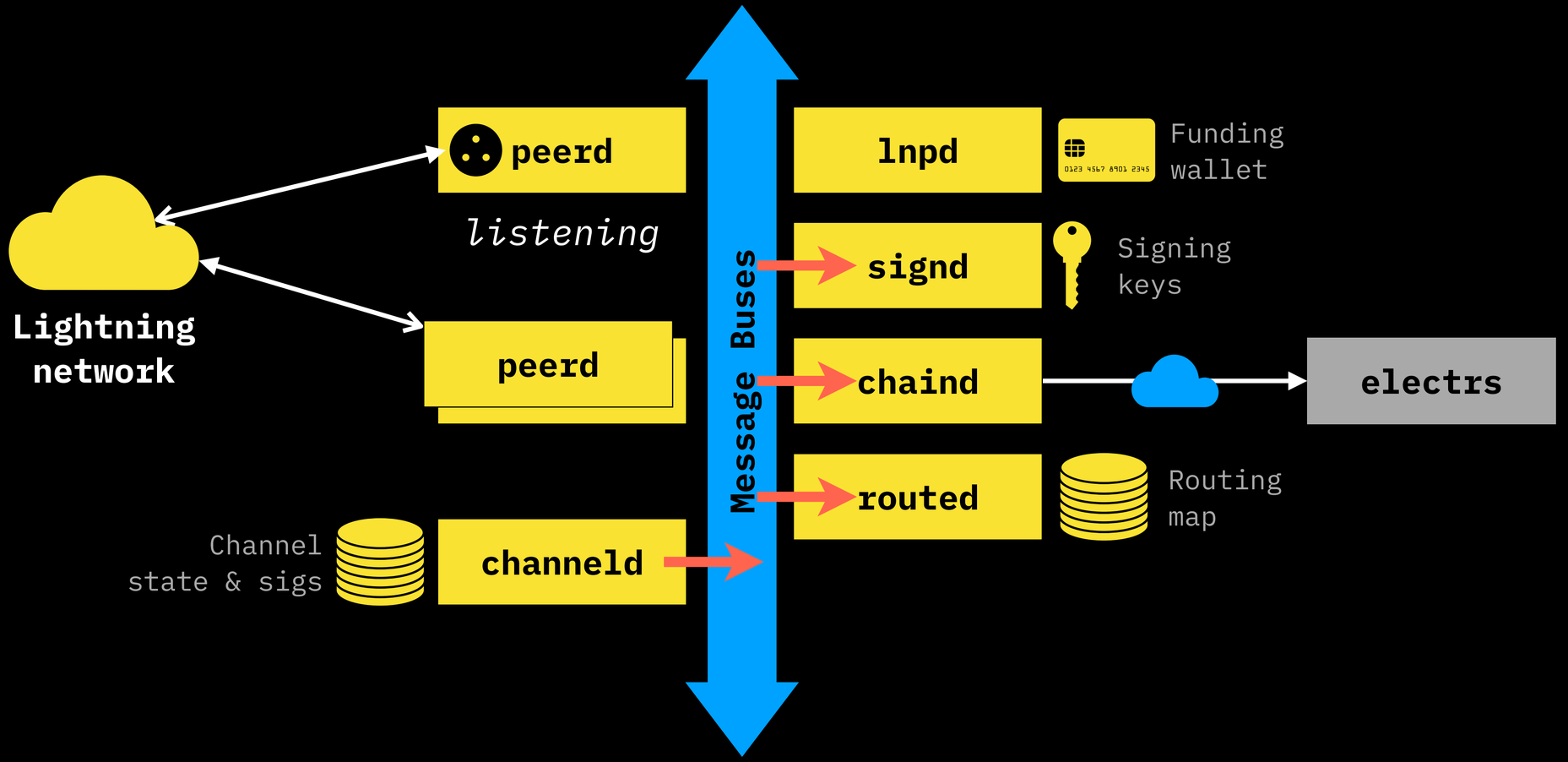
- Funding wallet
- Node key
- Per-channel set of basepoints

Demo of lightning basics

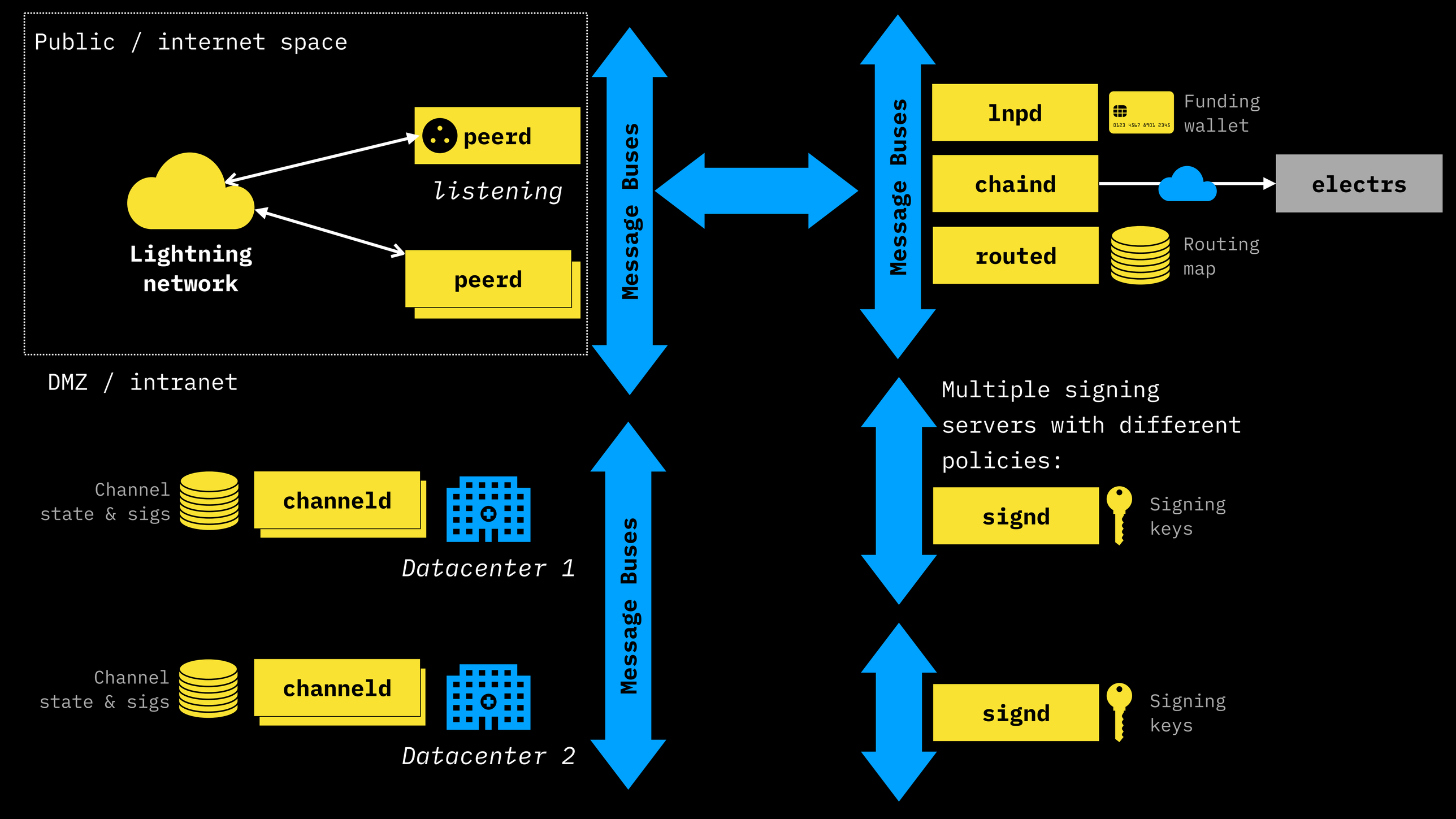

Message Buses
lnpd
signd
chaind
routed
0123 4567 8901 2345
Funding wallet
Signing keys
electrs
Routing map

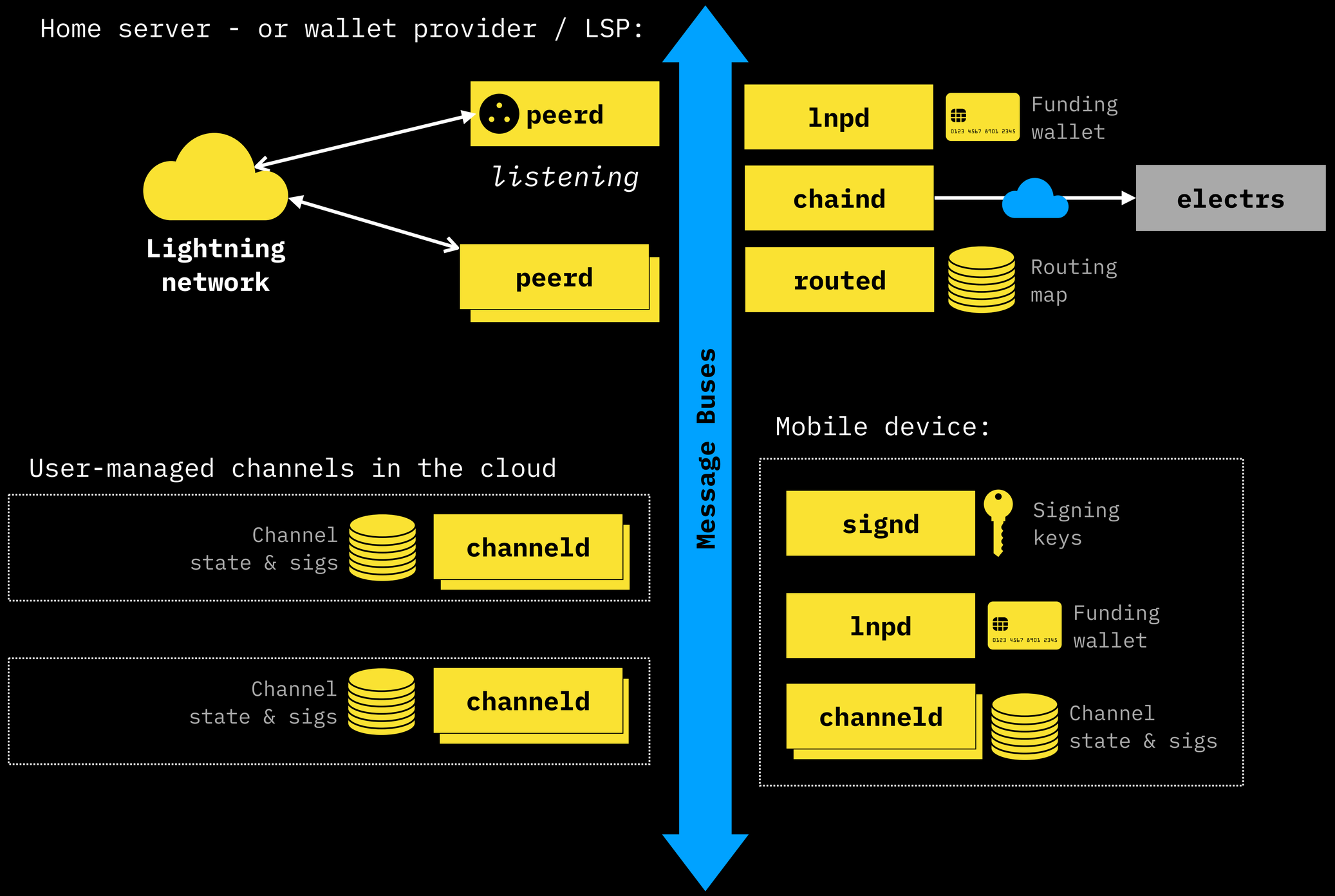

lnp_cli
rpc client
Message Buses
lnpd
signd
chaind
routed
0123 4567 8901 2345
Funding wallet
Signing keys
electrs
Routing map


peerd
listening
Lightning network
peerd
Channel state & sigs
channeld
Message Buses
lnpd
Funding wallet
signd
Signing keys
chaind
electrs
routed
Routing map

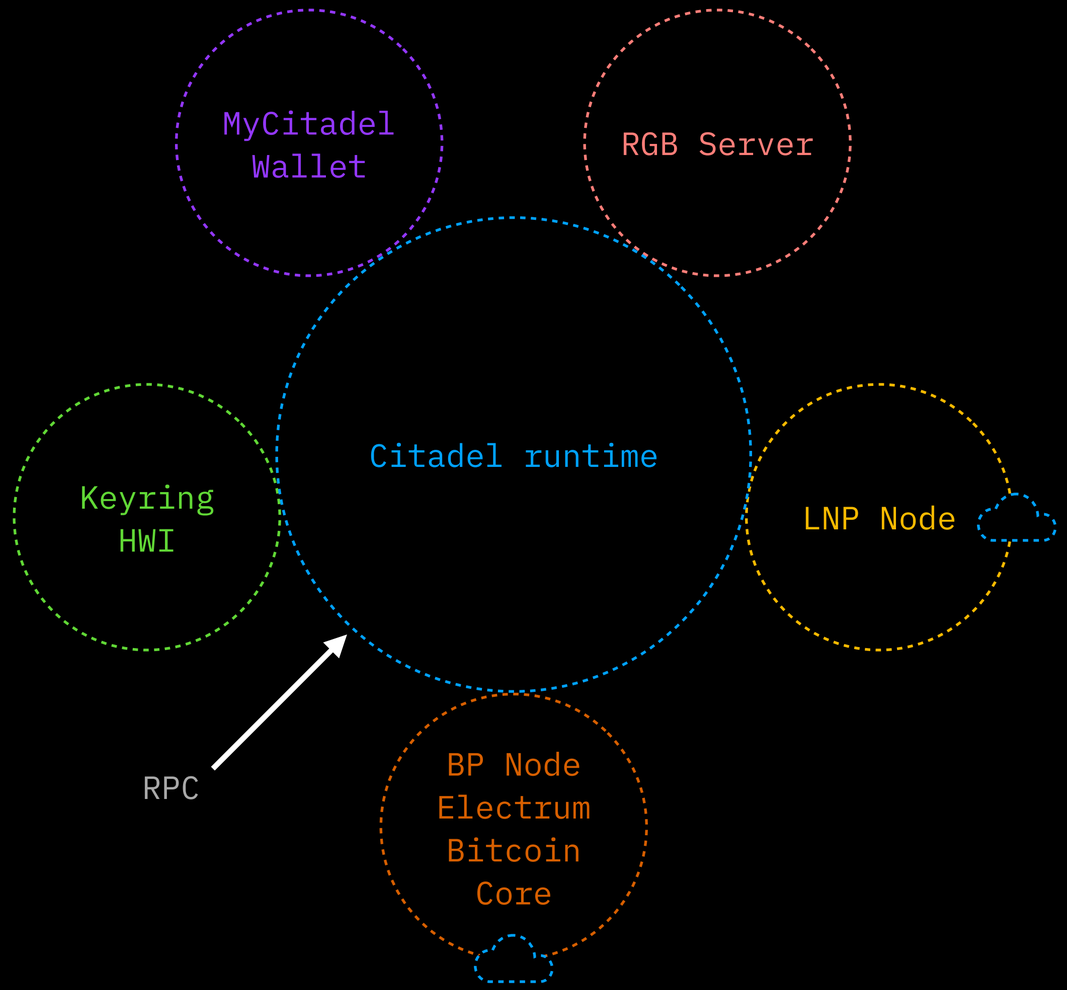

peerd
listening
peerd
Lightning network
Channel state & sigs
channeld
Message Buses
lnpd
Funding wallet
signd
Signing keys
chaind
electrs
routed
Routing map

LNP Node and infrastructure / service providers (LSP)

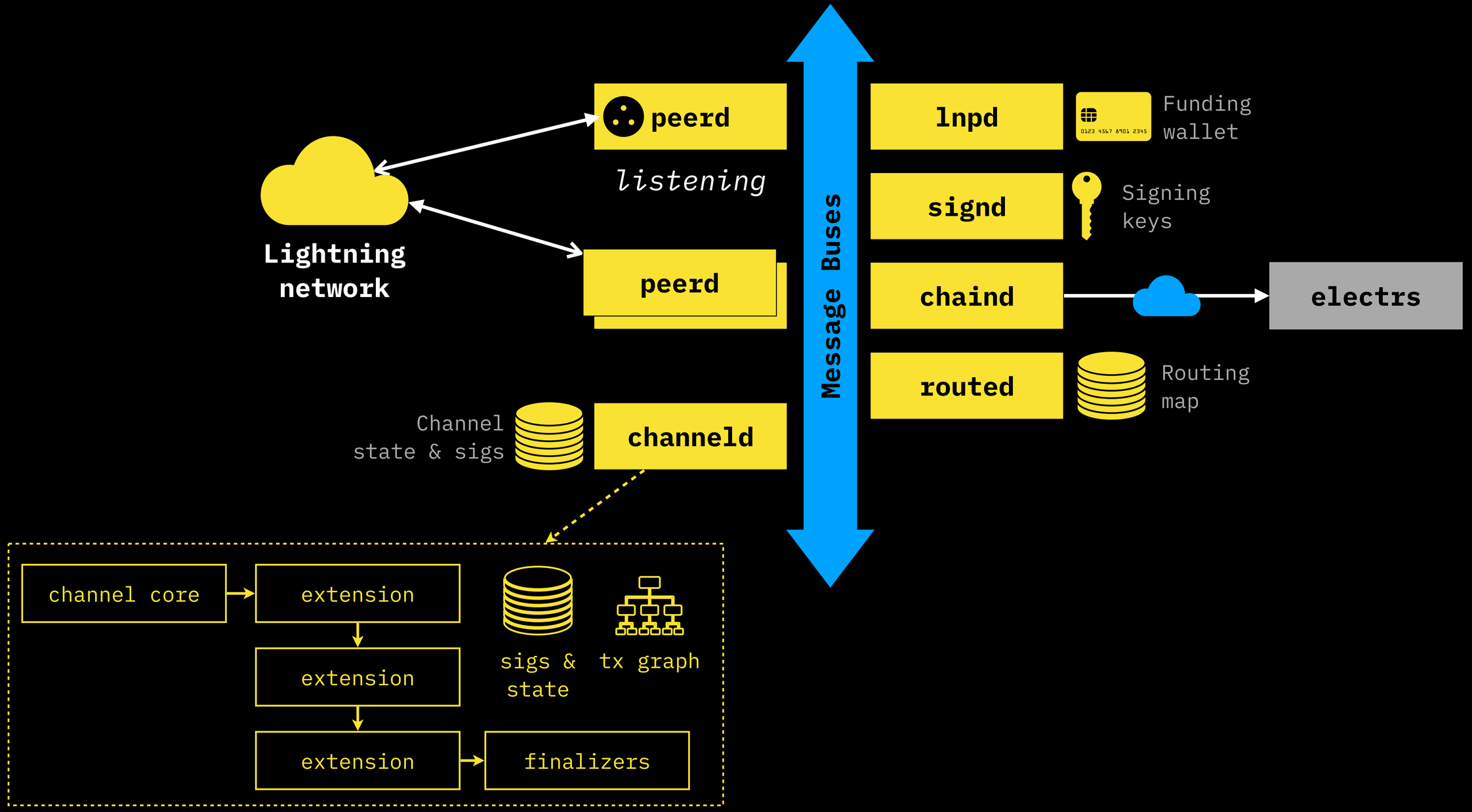

Public / internet space
peerd
listening
Lightning network
peerd
DMZ / intranet
Message Buses
Message Buses
lnpd
Funding wallet
chaind
electrs
routed
Routing map
Multiple signing servers with different policies:
signd
Signing keys
Channel state & sigs
channeld
Datacenter 1
Message Buses
Channel state & sigs
channeld
Datacenter 2
signd
Signing keys

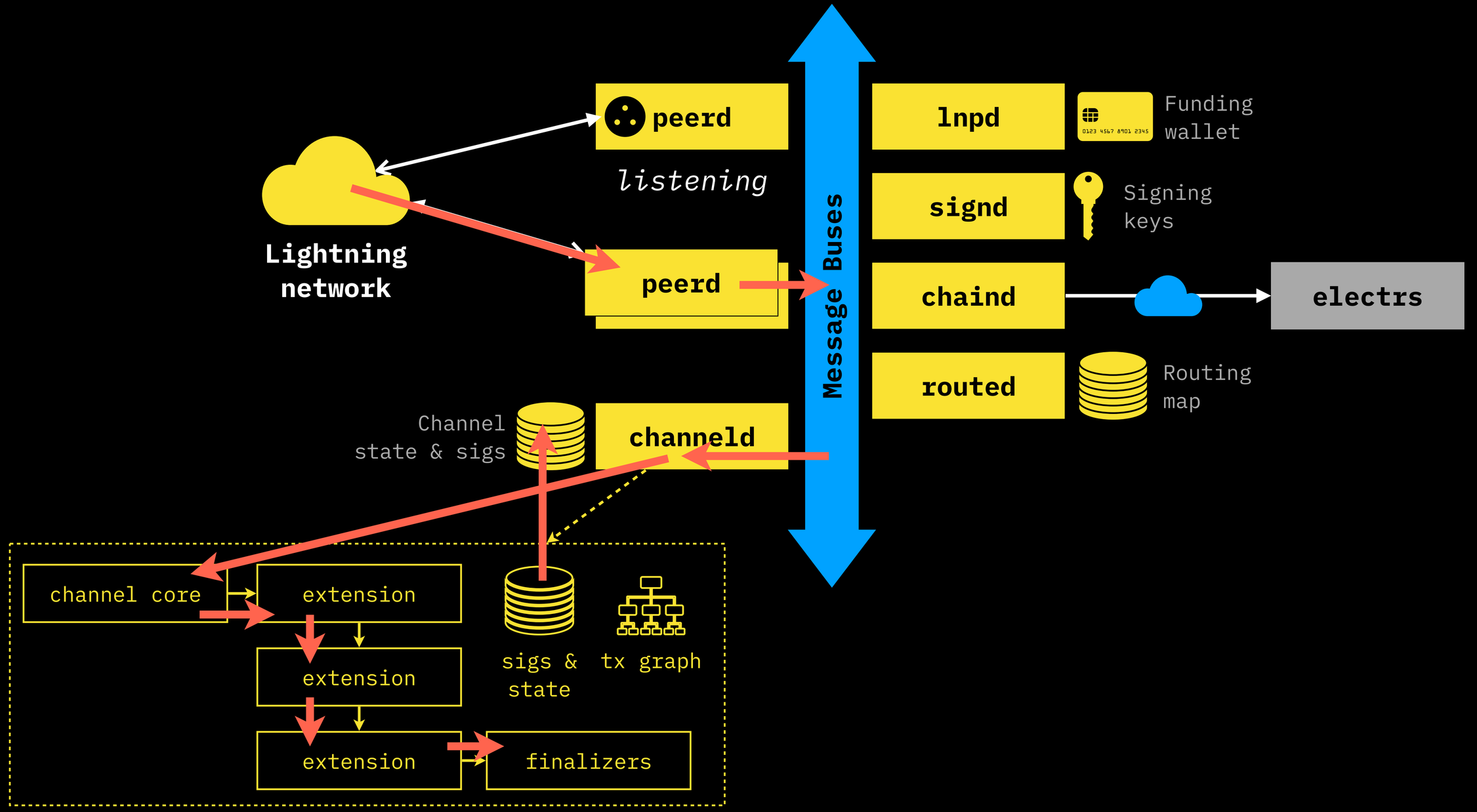

Home server - or wallet provider / LSP:
peerd
listening
Lightning network
peerd
Message Buses
lnpd
Funding wallet
chaind
electrs
routed
Routing map
Mobile device:
signd
Signing keys
lnpd
Funding wallet
channeld
Channel state & sigs
User-managed channels in the cloud
Channel state & sigs
channeld
Channel state & sigs
channeld

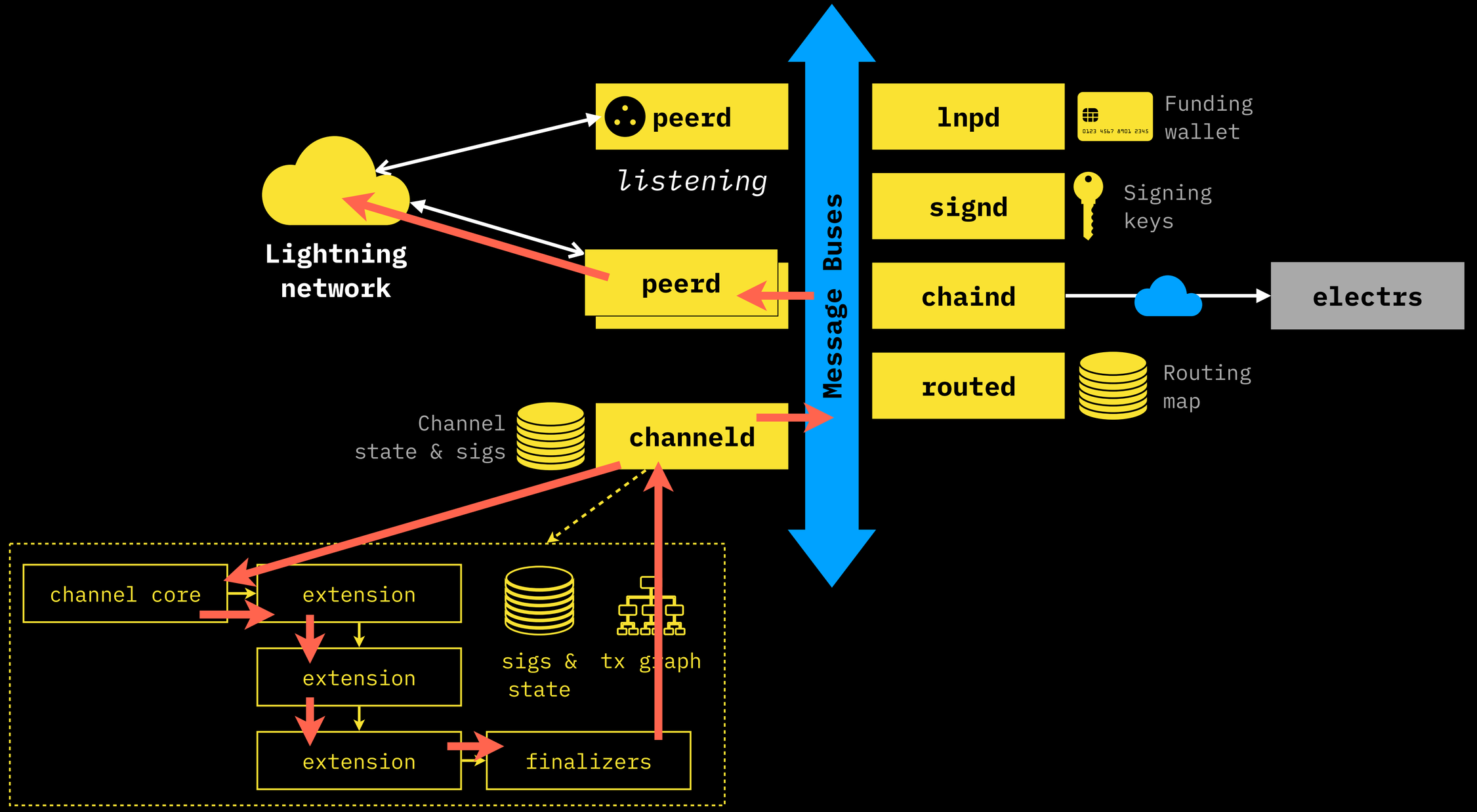
Citadel runtime for wallet devs



Extensions, extensions, extensions!


peerd
listening
Lightning network
peerd
Message Buses
lnpd
Funding wallet
signd
Signing keys
chaind
electrs
routed
Routing map
Channel state & sigs
channeld
channel core
extension
extension
extension
finalizers
sigs & state
tx graph


peerd
listening
Lightning network
peerd
Message Buses
lnpd
Funding wallet
signd
Signing keys
chaind
electrs
routed
Routing map
Channel state & sigs
channeld
channel core
extension
extension
extension
finalizers
sigs & state
tx graph


peerd
listening
Lightning network
peerd
Message Buses
lnpd
Funding wallet
signd
Signing keys
chaind
electrs
routed
Routing map
Channel state & sigs
channeld
channel core
extension
extension
extension
finalizers
sigs & state
tx graph

Everything in channel is an plugin!

- HTLC
- Anchor outputs
- Lexicographic output ordering

Extensions

- Read state from incoming peer messages
- Can contribute to composing messages to other peers
- Add / remove transactions to the graph
- Add / remove transaction outputs
- Manage their state
  channeld orchestrates state saving, restoration & backups across all extensions

Constructing channels like that not only allow us to quickly adopt new BOLT changes –

but also start with the Bifrost concept of generalized channels with no additional work required

Everything in channel is an plugin!

- HTLC or PTLC (or both!)
- RGB
- DLCs
- Storm
- Prometheus
- Lexicographic output ordering

Extensions unlock channel composability

Not only channels: extensions work for

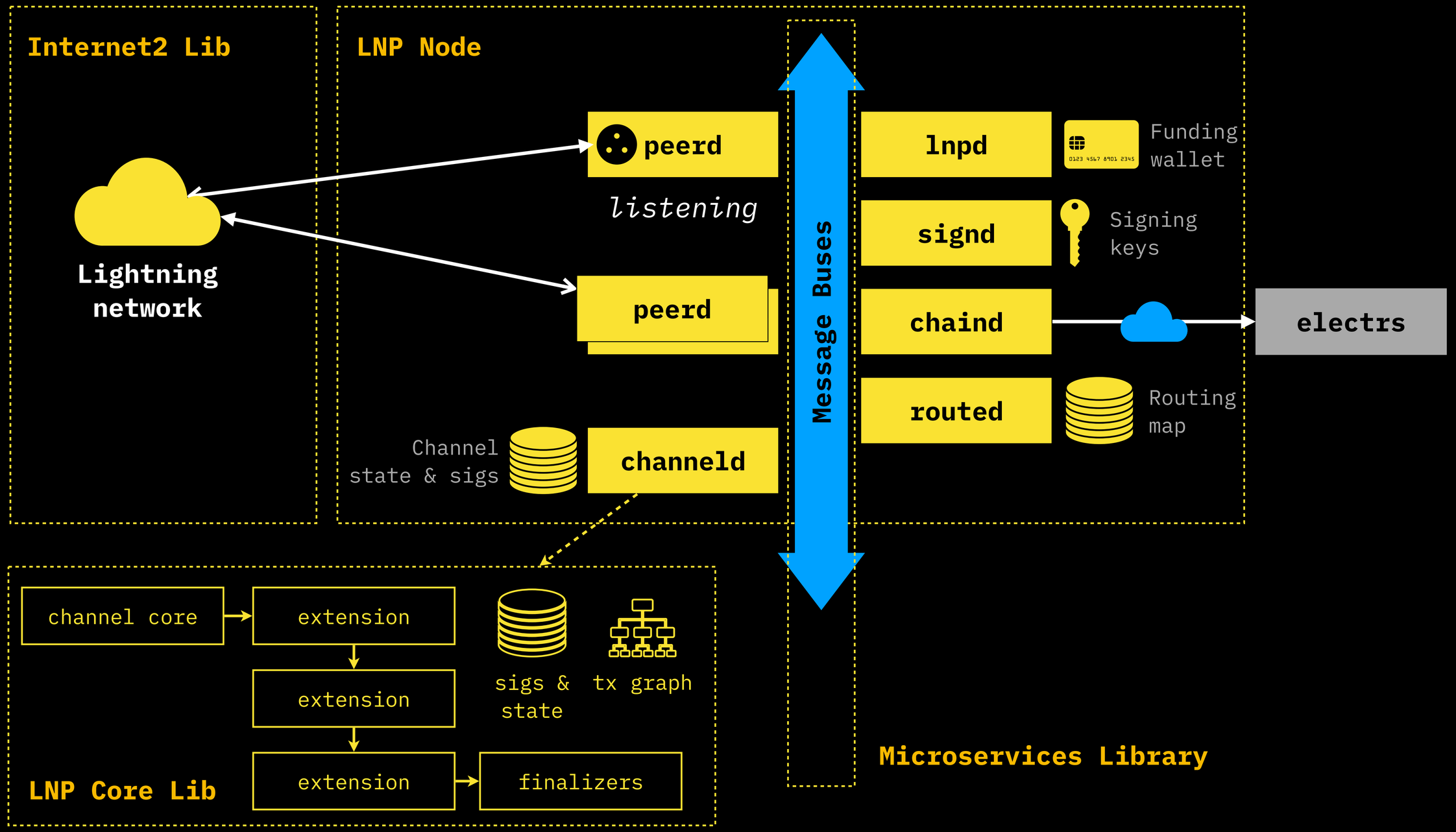
- Gossip protocol: currently BOLT-7
- Routing:
  - trampoline
  - ant routing
  - … more to come?

The ultimate goal* is to have all extensions to run on **AluVM** – such that external devs can create new channel types, routing algos etc, distribute to users, who can plug them in hot into wallets

** probably not that soon*

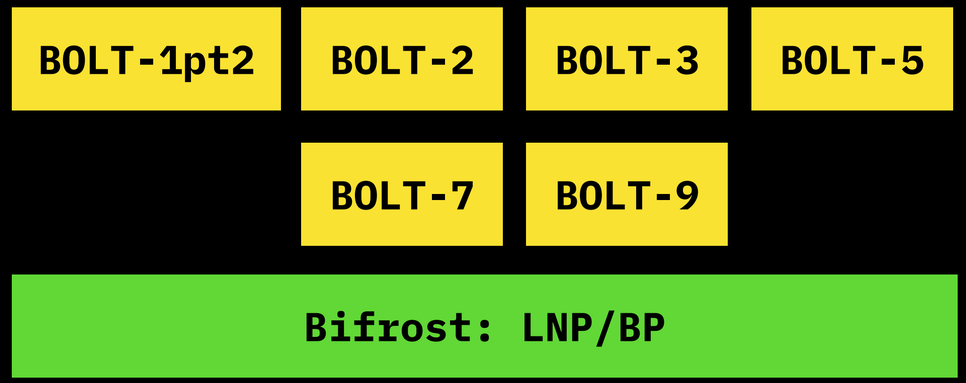
Dealing with the codebase

Abstraction is crucial for extensibility


Internet2 Lib
LNP Node
peerd
listening
peerd
Lightning network
Message Buses
lnpd
Funding wallet
signd
Signing keys
chaind
electrs
routed
Routing map
Channel state & sigs
channeld
channel core
extension
extension
extension
finalizers
sigs & state
tx graph
LNP Core Lib
Microservices Library

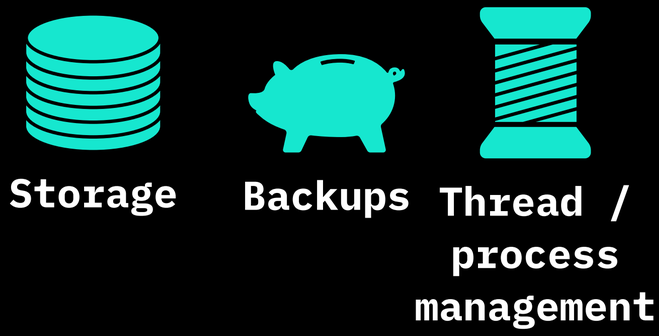
LNP Core

- Everything related to BOLT - or LNPBP standards



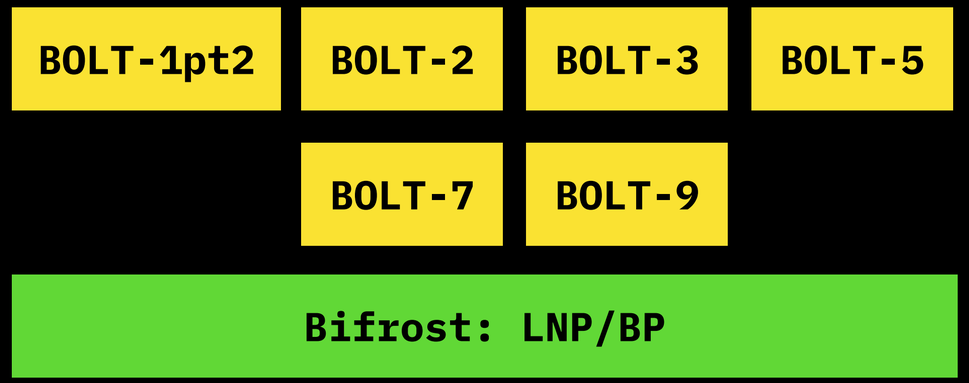
LNP Node

- Everything aside from BOLT/LNPBP standards



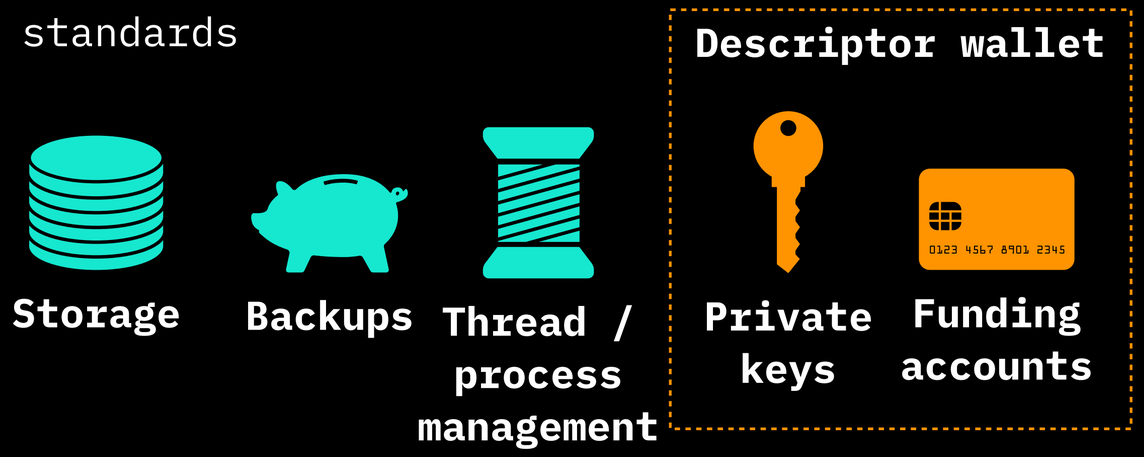
LNP Core

- Everything related to BOLT - or LNPBP standards



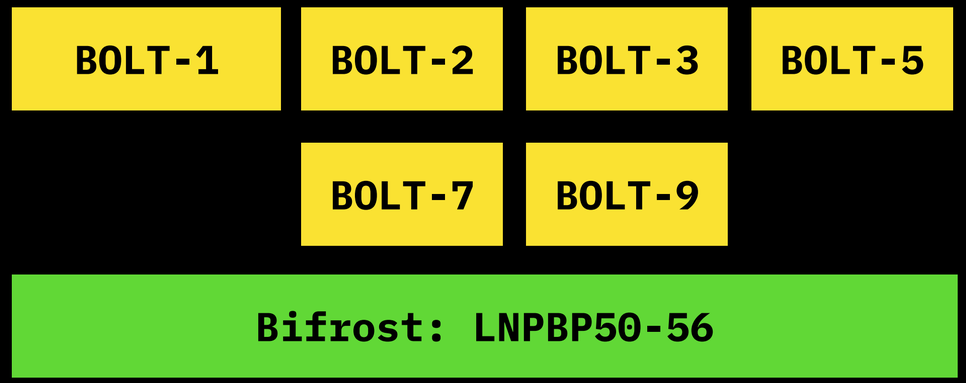
LNP Node

- Everything aside from BOLT/LNPBP standards



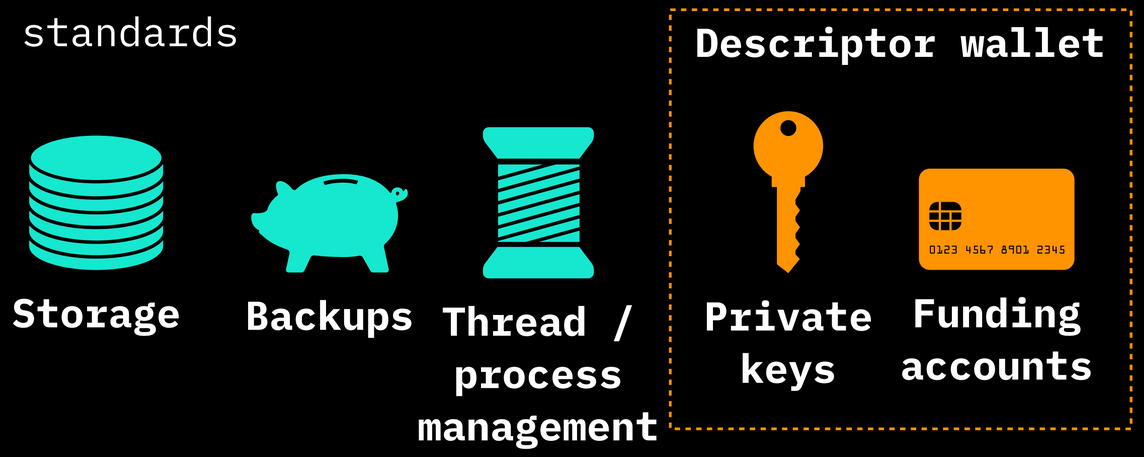
LNP Core

- Everything related to BOLT - or LNPBP standards



LNP Node

- Everything aside from BOLT/LNPBP standards



Internet2

Microservices

Libraries

- github.com/LNP-BP/LNP-Core — **BOLTs & Bifrost**
- github.com/LNP-BP/LNP-Node — **Node, RPC, client**
- github.com/LNP-BP/descriptor-wallet — **Wallet lib**
- github.com/internet2-org/rust-internet2 — **Networking**
- github.com/internet2-org/rust-microservices — **Infrastructure**
- github.com/LNP-BP/LNPBPs — **Standards**

Channel operations

Demo of lightning channel operations