RGB & Spectrum project status

by October 21st, 2019

Project development streams

- **Core RGB** based on client-side validation paradigm

  security-critical part
  core reviewers: Peter Todd, Andrew Poelstra, may be Peter Wuille

- **Lightning Network RGB & Spectrum** based on gossip & onion routing

  scalability-critical part
  core reviewers: Christian Decker, may be Alex Buthworth

- **Confidential Assets** interoperability

  privacy- and interoperability-critical part
  core reviewers: Andrew Poelstra, Adam Back

RGB+CA: four dimensions of the privacy

- **Amounts:**

  Pedersen commitments + Borromean signatures range-proofs in CT/CA today

  Pedersen commitments + bulletproofs range-proofs in CT2/RGB tomorrow

- **The fact of transaction (non-public tx):**

  Client-size validation with elliptic curve commitments in RGB

- **Addresses:**

  Merklezation of asset owning addresses inside the proofs

- **Assets:**

  Pedersen commitments and encapsulation for multi-asset transfers

We have to launch all streams together
after thorough review and verification

Project development risks

- **Core RGB:**

  Peter Todd input is critical in security assessment;
  multiple re-designs of protocol structure

- **Lightning Network:**

  immature LN functionality will require constant re-integration
  need to implement & support own LN (sub)node

- **Confidential Assets:**

  complexity of zero-knowledge components
  risks of hidden inflation

Detected & addressed challenges so far

- **Fitting RGB into LN in standards-compliant way**
  - ➙ final parts of the problem were solved with the latest BOLT changes discussed and accepted during the Lightning Conference

- **LN instability:** the protocol will undergo drastic unpredictable changes at least each quarter
  - ➙ Modularise and create proper abstractions within RGB and Spectrum, increasing its flexibility and decreasing risks: if something will become incompatible it can be replaced as a module
  - ➙ Work with Christian Decker & community on a more strict & fixed LN standards and rust node (see also next point)
  - ➙ Change a number of previously-defined RGB parts (like commitments)

Detected & addressed challenges so far

- **RGB can’t be implemented as just an extension to Lightning Nodes due to architecture limitations**
  - ➞ Found a way with Christian Decker how we can avoid the need for a complete maintained c-lightning fork and build rust-node by stages attached to a mainstream c-lightning implementation
  - ➞ Chaincode may co-operate on this development
  - ➞ It will take months to do the necessary changes even outside of RGB/Spectrum functionality :(

Detected & addressed challenges so far

- **Confidential assets interoperability**
  - ➞ Adam Back is willing to explore the potential for a joint asset standard for Bitcoin, Liquid and LN with RGB+CA joint spec
  - ➞ Client-side validation paradigm can be very useful for Liquid, since it delivers both better privacy and scalability
  - ➞ With Andrew Poelstra we need to explore the space of possible options
  - ➞ I have prepared a draft, but much more work is still required

Project development status

- **Core RGB:**

  33% of final specs
  33% of final code
  100% of spec drafts
  2 previous specs and 3 implementations were discarded

- **Lightning Network:**

  50% of final specs, 100% of spec drafts
  all issues and problems are analysed and we know how to address them
  design for a c-lightning fork and further maintenance

- **Confidential Assets:**

  generic design for embedding zero-knowledge & building CA compatibility
  Blockstream awareness and openness to work on generic industry standard for assets

What is currently required

- **Core RGB:**

  Peter Todd: write single-use seals specs & code

- **Lightning Network:**

  me & Christian Decker: refactor c-lightning architecture & specs
  get full-time developer
  me & Chaincode: work on rust implementation of rgb-specific daemons for c-lightning

- **Confidential Assets:**

  work with Andrew Poelstra on their design

Potential deadlines

- **Core RGB:**

  Specs: expect to finish new iteration in early **Nov** (depends on Peter Todd)

  Daemon+cli: end of November, but w/o CA and LN functionality

- **Lightning Network:**

  end of this year: first very early prototypes with c-lightning RGB/ Spectrum-specific daemons

  mid-next year: beta for asset over lightning

- **Confidential Assets:**

  end of November: initial design with Andrew Poelstra

Project modules / libraries

RGB: assets

client-side validated state history graphs

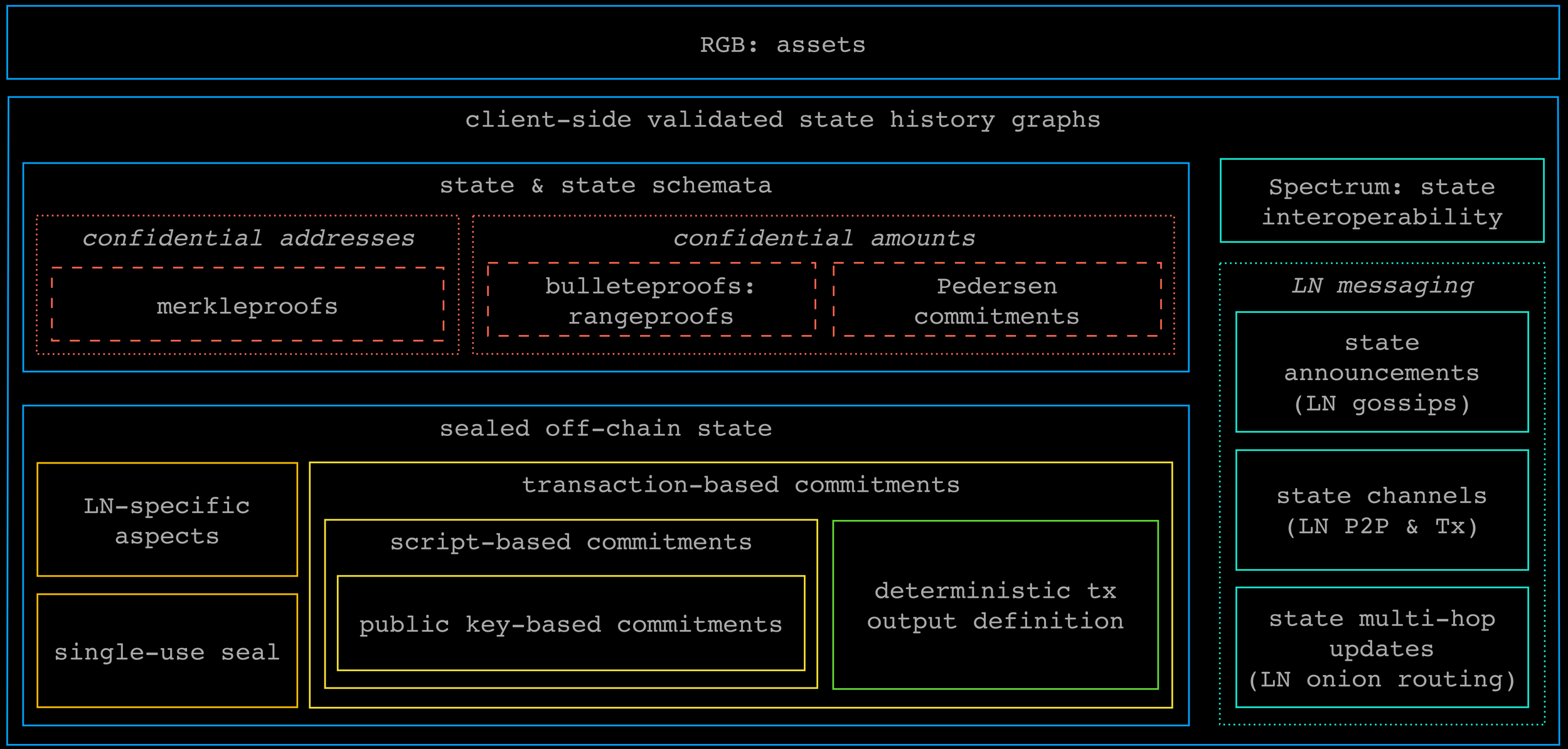
confidential state

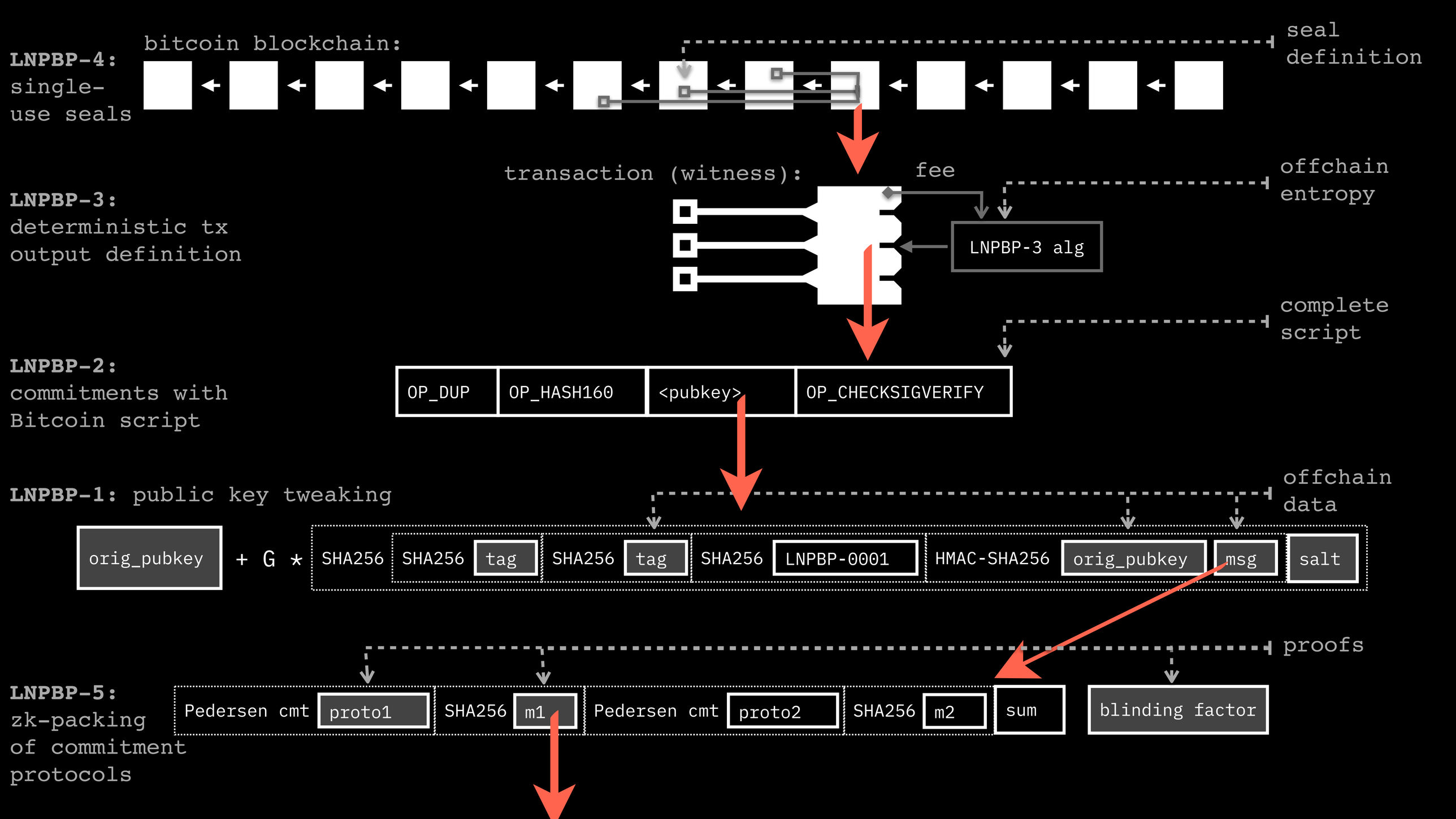
Spectrum: state interoperability

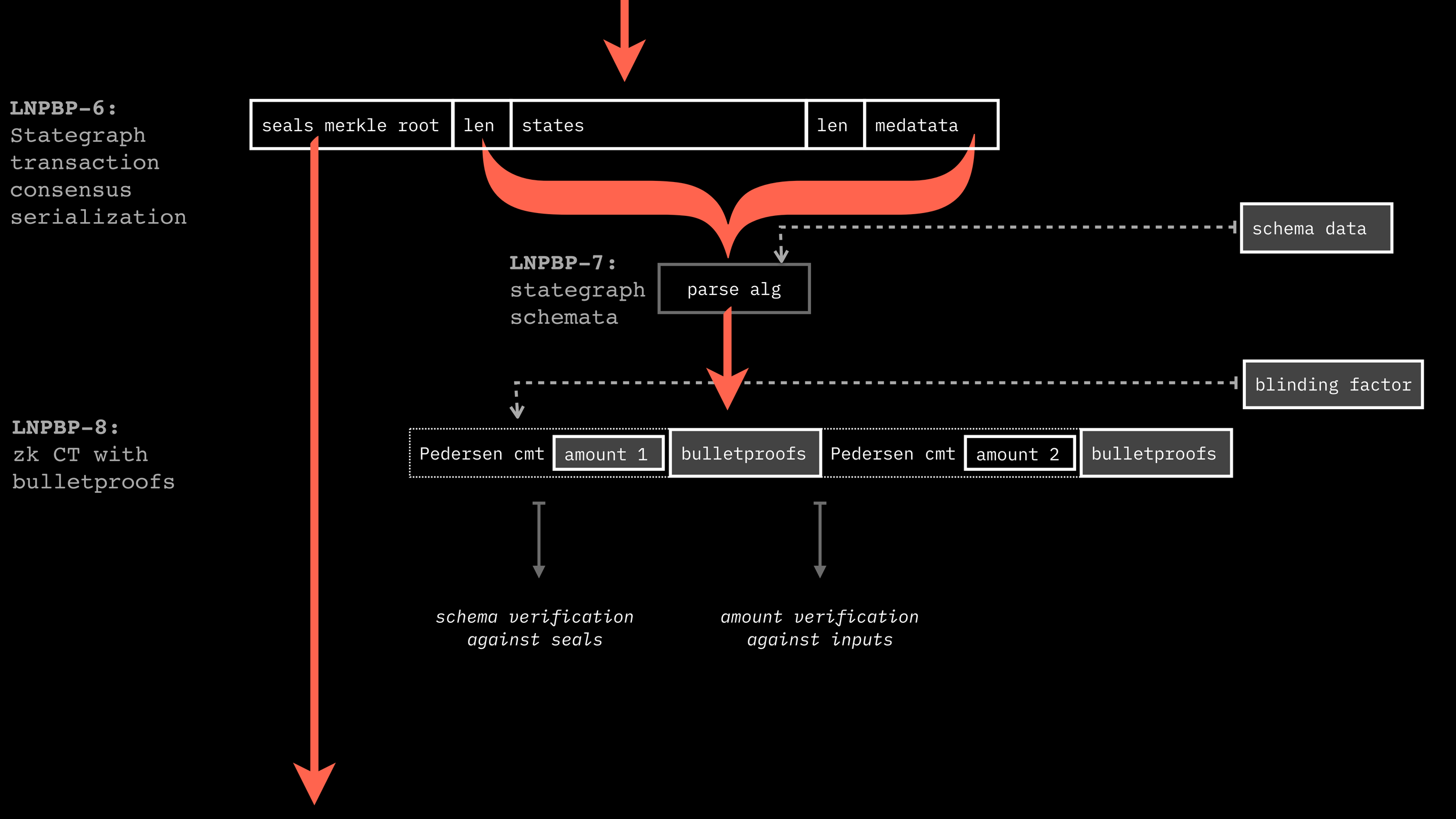
state commitment history

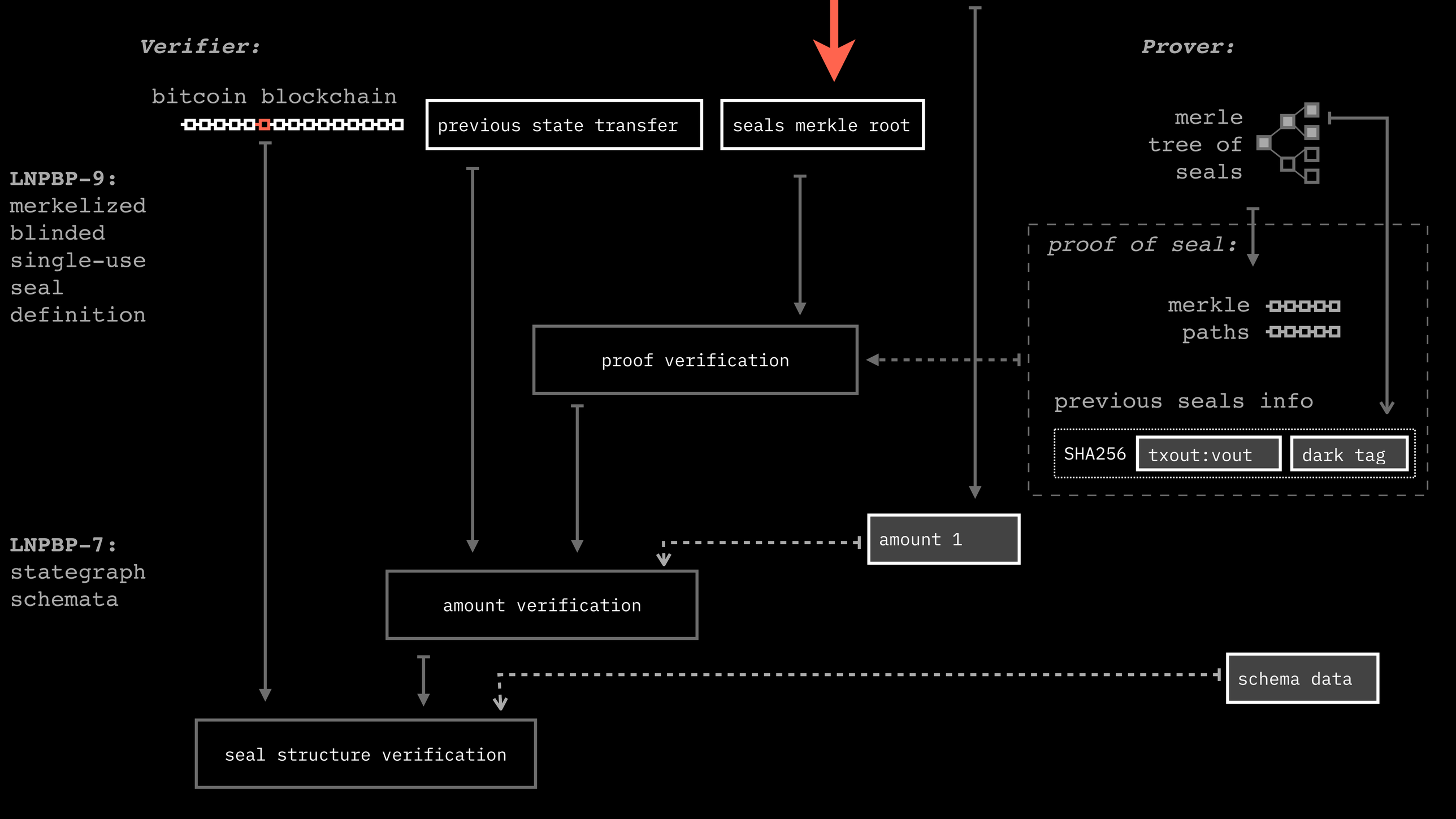
LN in-channel state information

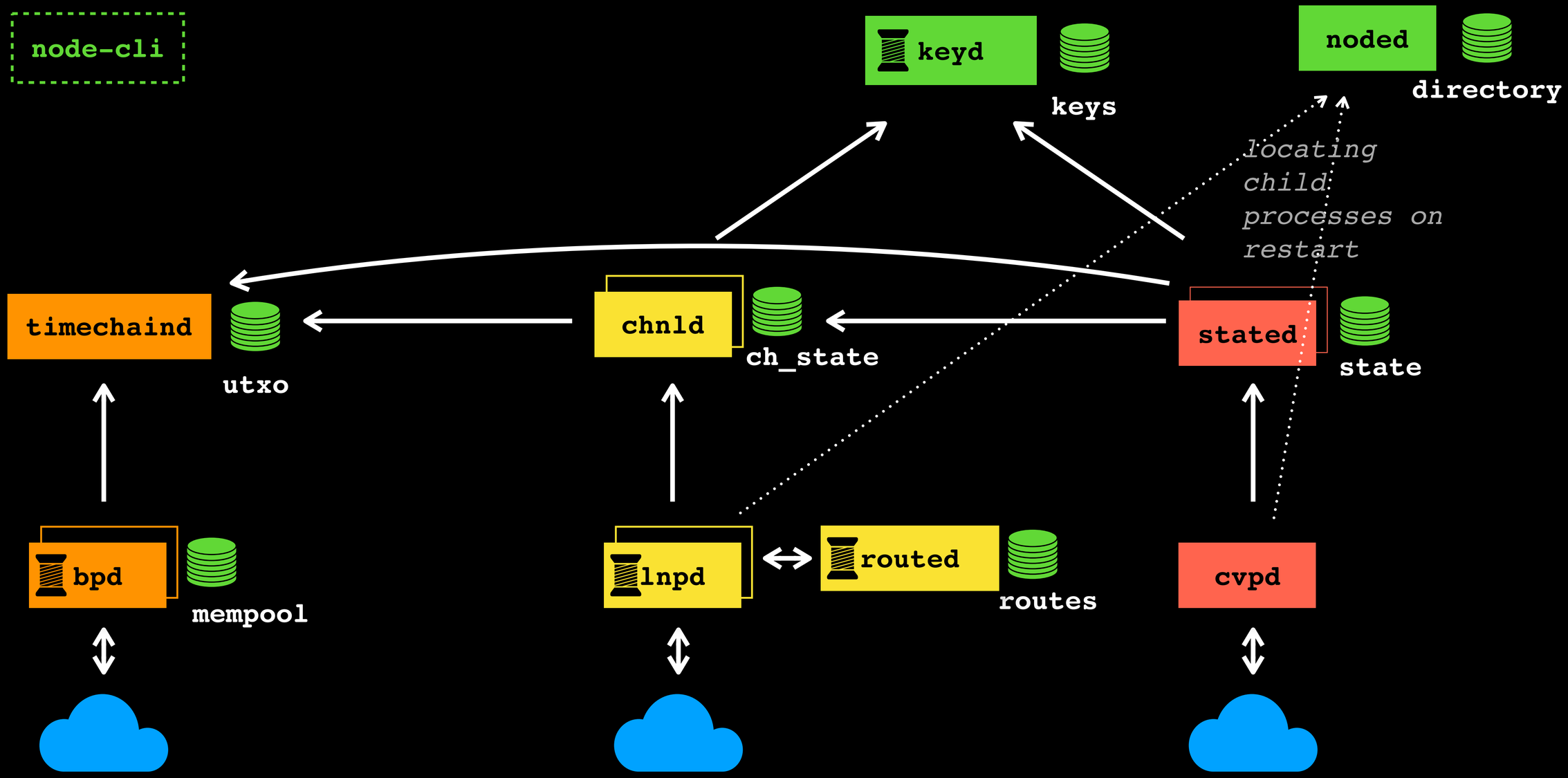
Project modules / libraries




LNPBP-4:
single-
use seals
bitcoin blockchain:
seal
definition
LNPBP-3:
deterministic tx
output definition
transaction (witness):
fee
offchain
entropy
LNPBP-3 alg
complete
script
LNPBP-2:
commitments with
Bitcoin script
OP_DUP
OP_HASH160
<pubkey>
OP_CHECKSIGVERIFY
LNPBP-1: public key tweaking
offchain
data
orig_pubkey
+ G *
SHA256
SHA256
tag
SHA256
tag
SHA256
LNPBP-0001
HMAC-SHA256
orig_pubkey
msg
salt
proofs
LNPBP-5:
zk-packing
of commitment
protocols
Pedersen cmt
proto1
SHA256
m1
Pedersen cmt
proto2
SHA256
m2
sum
blinding factor


LNPBP-6:
Stategraph
transaction
consensus
serialization
seals merkle root
len
states
len
medatata
schema data
LNPBP-7:
stategraph
schemata
parse alg
blinding factor
LNPBP-8:
zk CT with
bulletproofs
Pedersen cmt
amount 1
bulletproofs
Pedersen cmt
amount 2
bulletproofs
schema verification
against seals
amount verification
against inputs


Verifier:
bitcoin blockchain
previous state transfer
seals merkle root
Prover:
merle tree of seals
LNPBP-9: merkelized blinded single-use seal definition
proof of seal:
merkle paths
proof verification
previous seals info
SHA256
txout:vout
dark tag
amount 1
LNPBP-7: stategraph schemata
amount verification
schema data
seal structure verification


node-cli
keyd
keys
noded
directory
locating child processes on restart
timechaind
utxo
chnld
ch_state
stated
state
bpd
mempool
lnpd
routed
routes
cvpd

TL;DR

- **Positive things:**
  - Industry-wide involvement: Blockstream, Chaincode, multiple independent reviewers
  - High potential for common standard with Confidential Assets
  - Better privacy features
  - Many security risks and potential vulnerabilities were mitigated
  - Better modularisation simplifies future flexibility
  - Found a way to implement LN-specific parts; conflicts with standards are mitigated
- **Negative things:**
  - A lot of stuff to do and develop
  - Many external risks with LN internal instability
  - Much more work on LN than was expected (need to re-implement part of LN node functionality)
  - We will be very restricted in what we can upgrade after the first release, so we need to deliver 100% working and tested product from the day 1 in production