State of the Ark

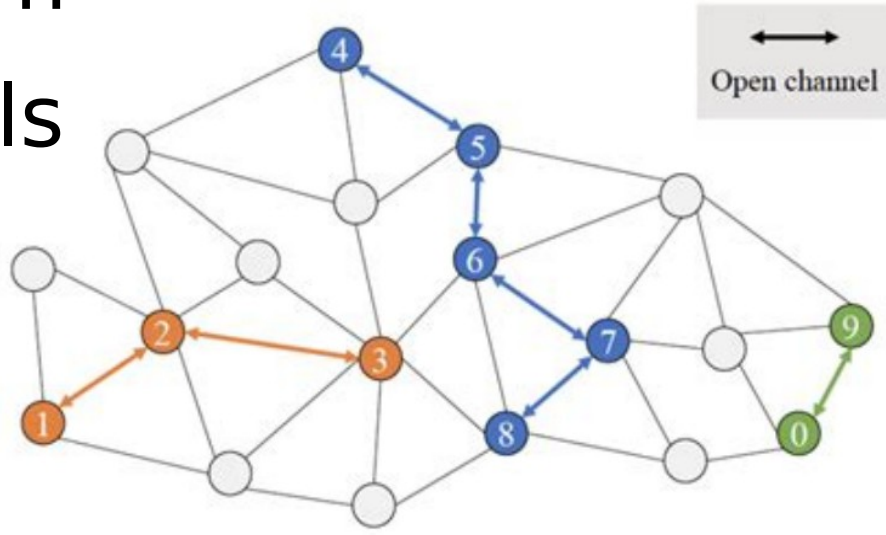
progress on a new Bitcoin layer 2 protocol

Who am I?

- Steven Roose
- Bitcoin dev >10 years
- formerly Liquid team @ Blockstream
- rust-bitcoin

Lightning Network

- off-chain payment protocol
- connected graph of two-party channels
- inbound liquidity problem
- on-chain cost of channels



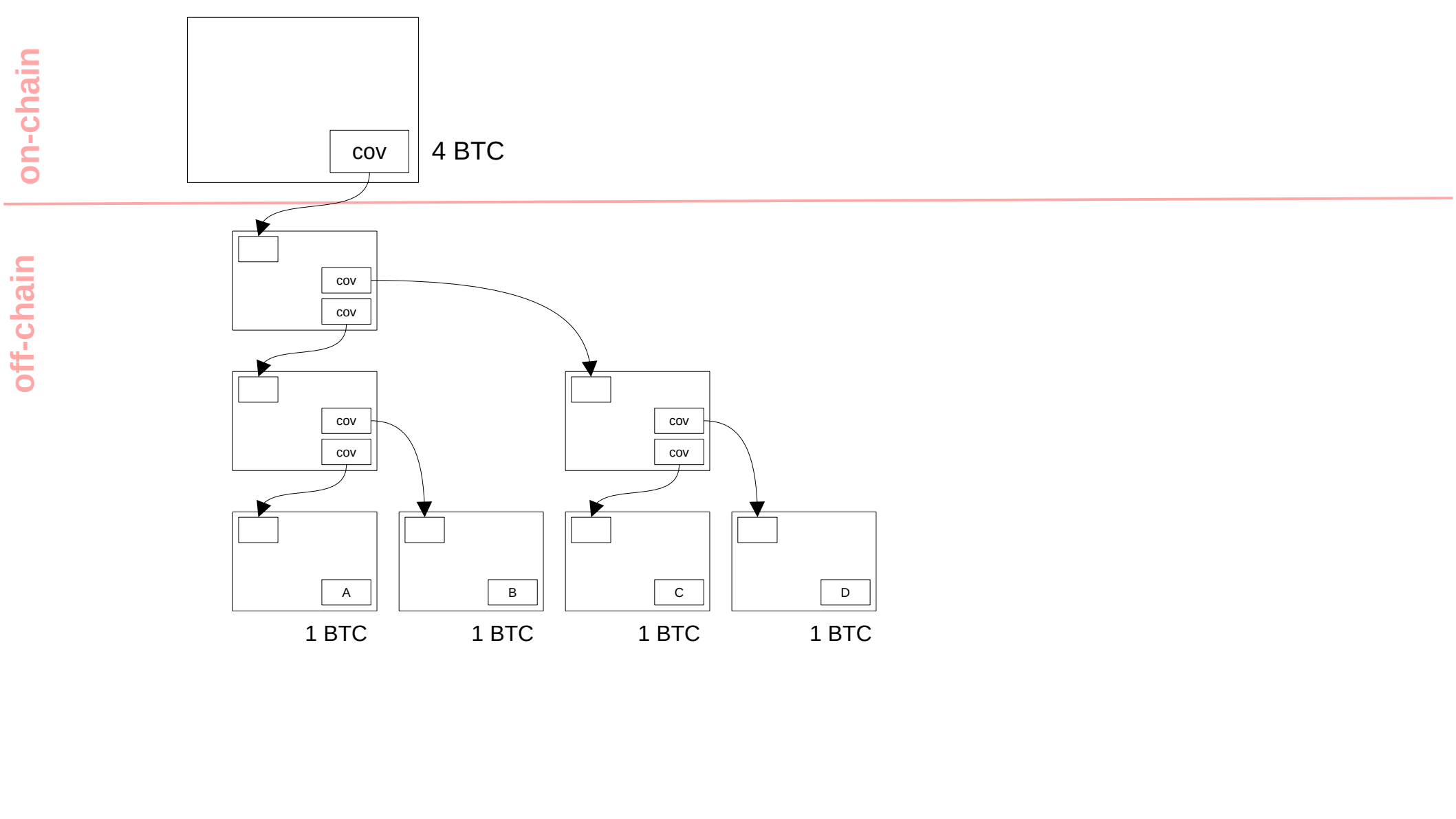
Introducing Ark

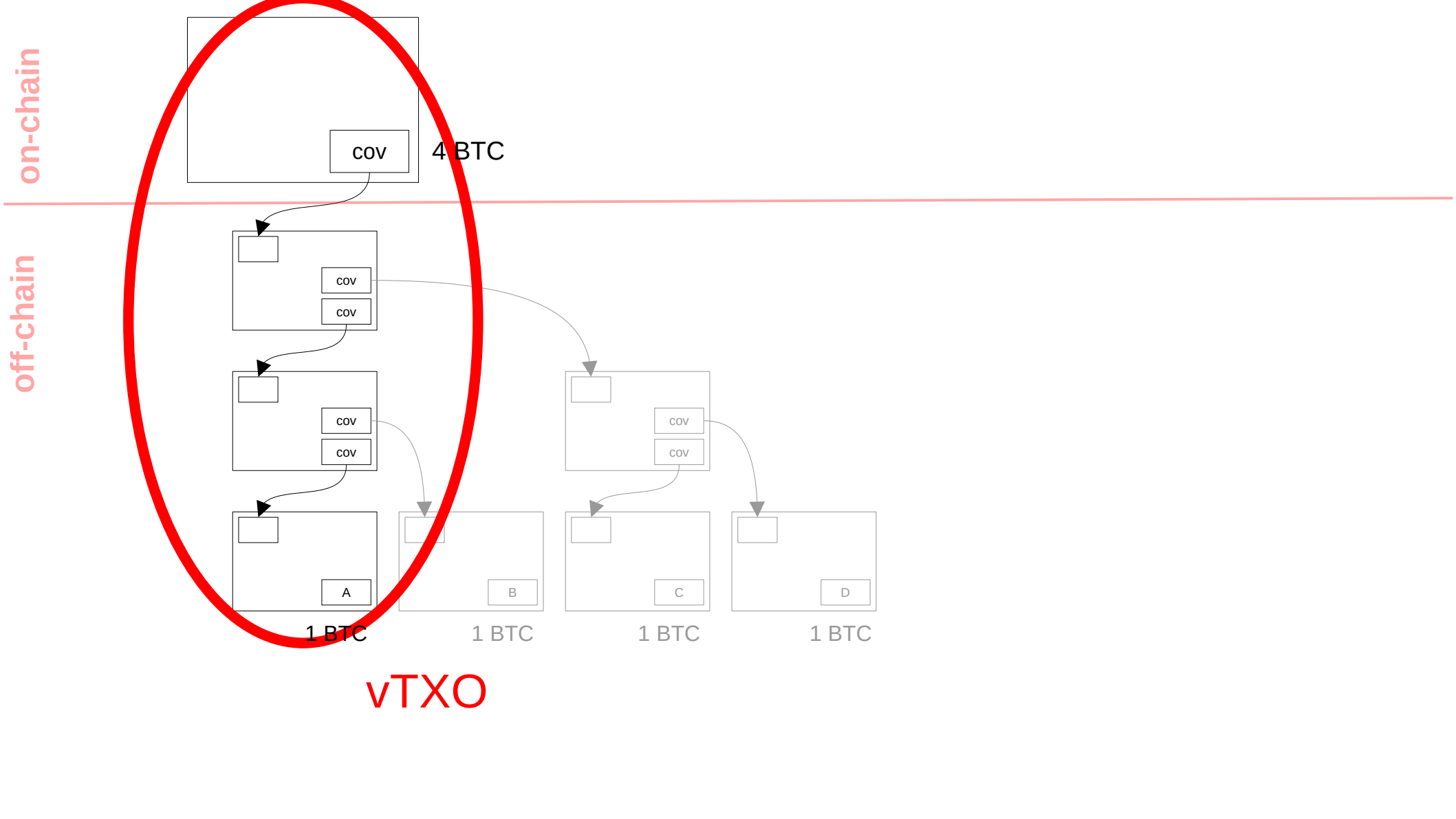
- new layer 2 protocol for Bitcoin
  - → interoperable with Lightning
- shared UTXO model: VTXOs
  - → exchanging VTXOs for new VTXOs

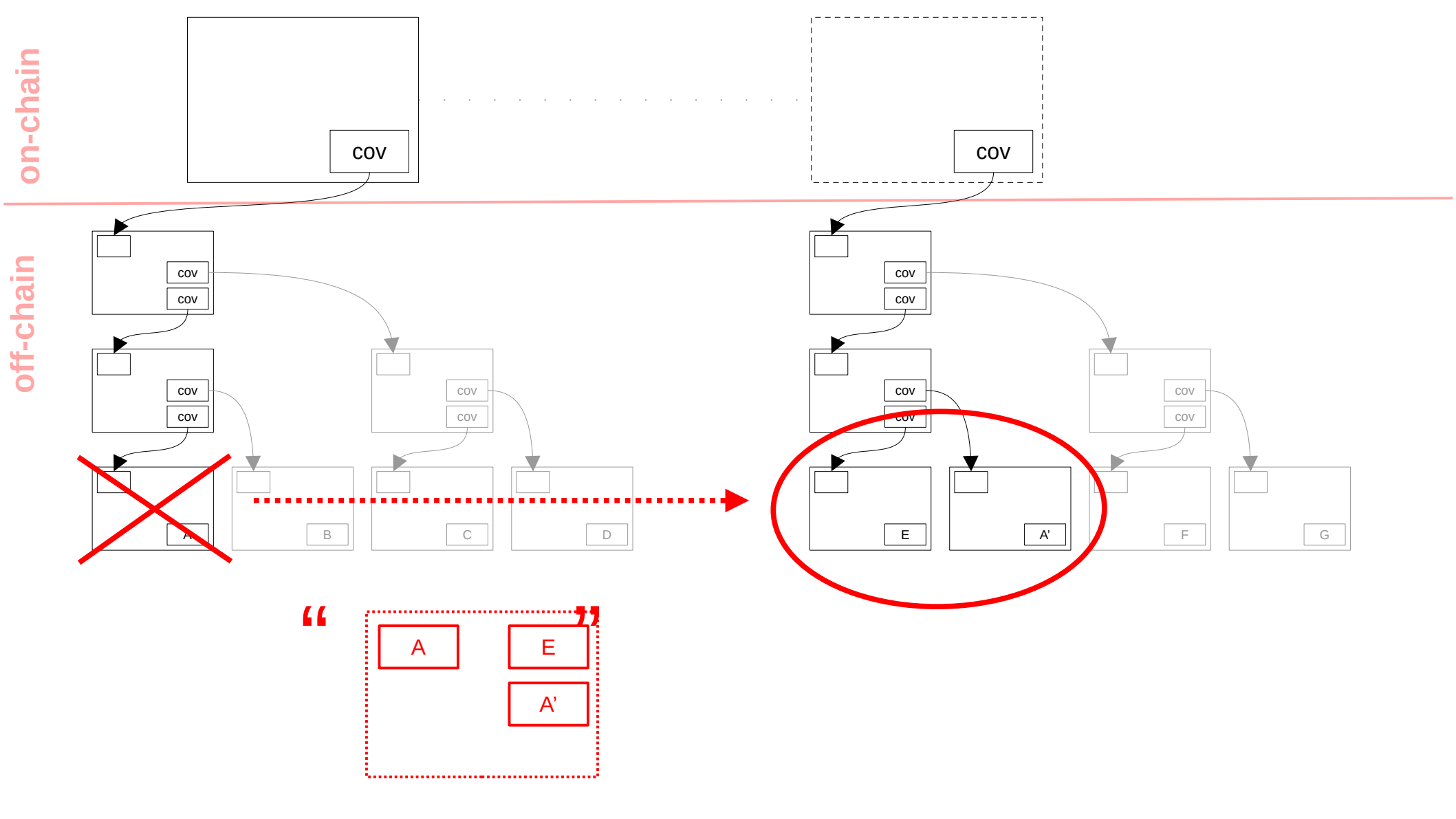
Covenants

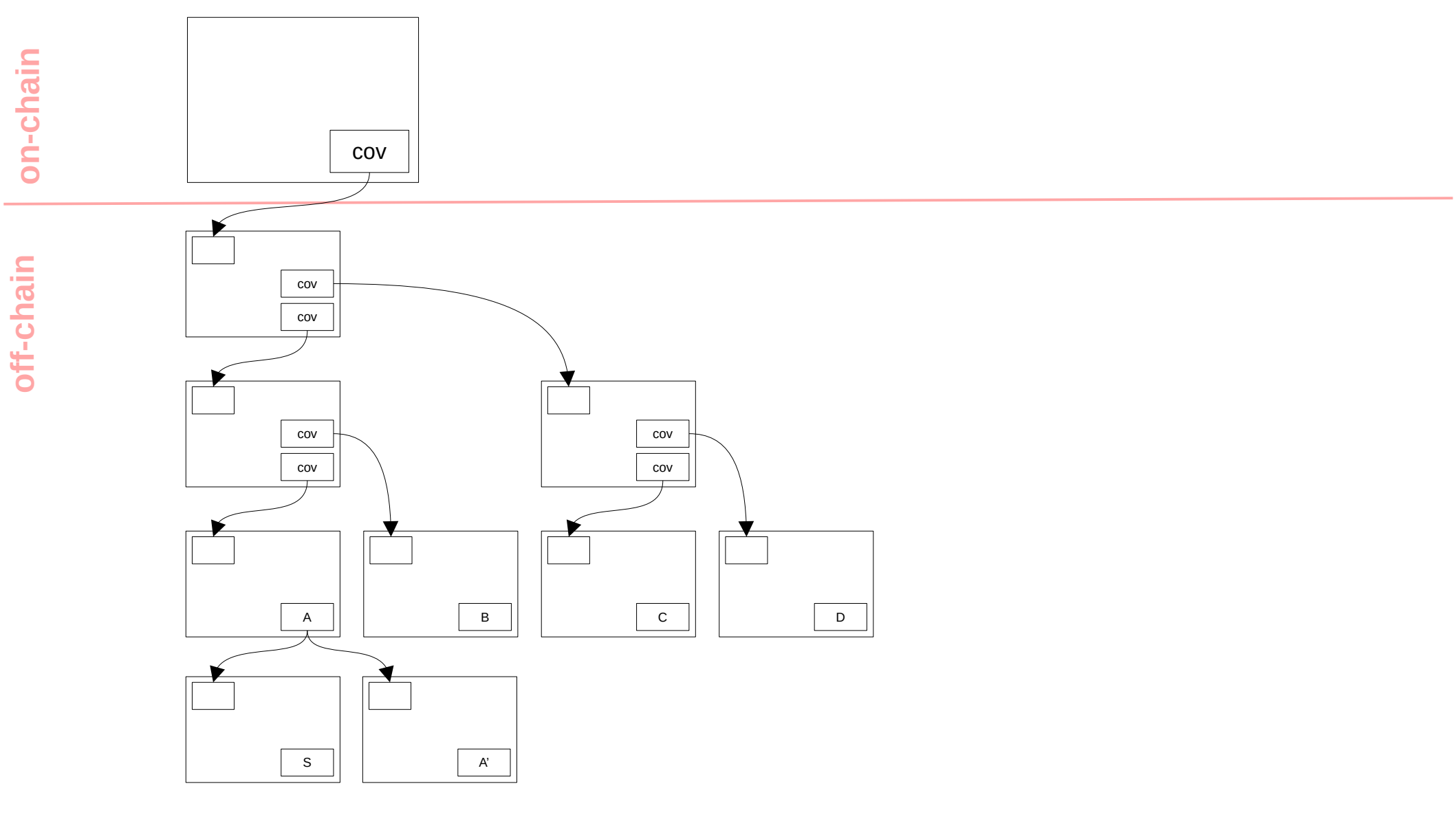
- restriction on where the money in a UTXO can go
- can be emulated by pre-signed transactions


on-chain
cov
4 BTC
off-chain
A
1 BTC
B
1 BTC
C
1 BTC
D
1 BTC


on-chain
cov
4 BTC
off-chain
cov
cov
cov
cov
cov
cov
cov
cov
A
B
C
D
1 BTC
1 BTC
1 BTC
1 BTC

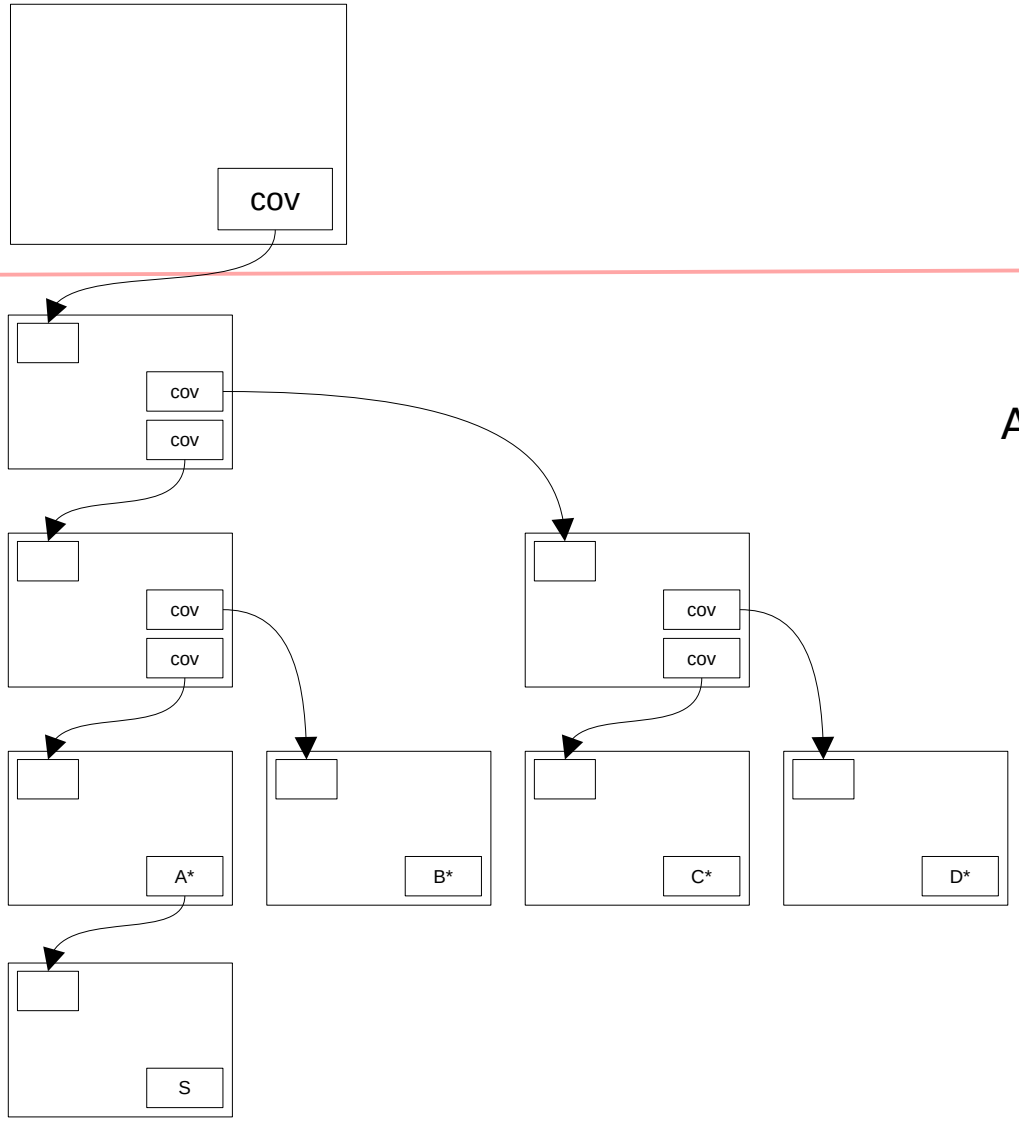

on-chain
off-chain
cov
4 BTC
cov
cov
cov
cov
cov
cov
A
B
C
D
1 BTC
1 BTC
1 BTC
1 BTC
vTXO


on-chain
off-chain
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
A
B
C
D
E
A'
F
G
“
”
A
E
A’


on-chain
off-chain
COV
cov
cov
cov
cov
cov
cov
A
B
C
D
S
A'

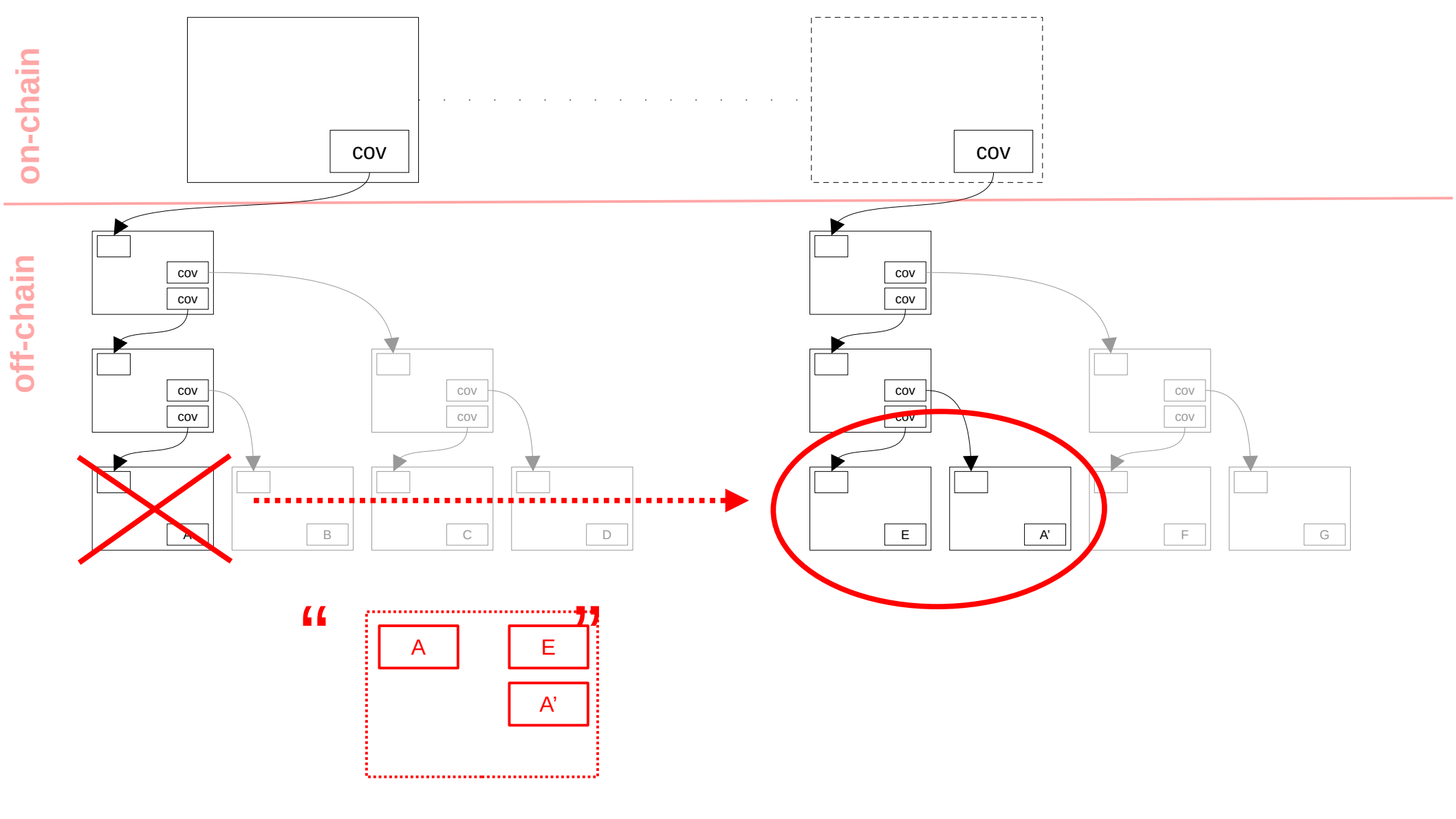
on-chain

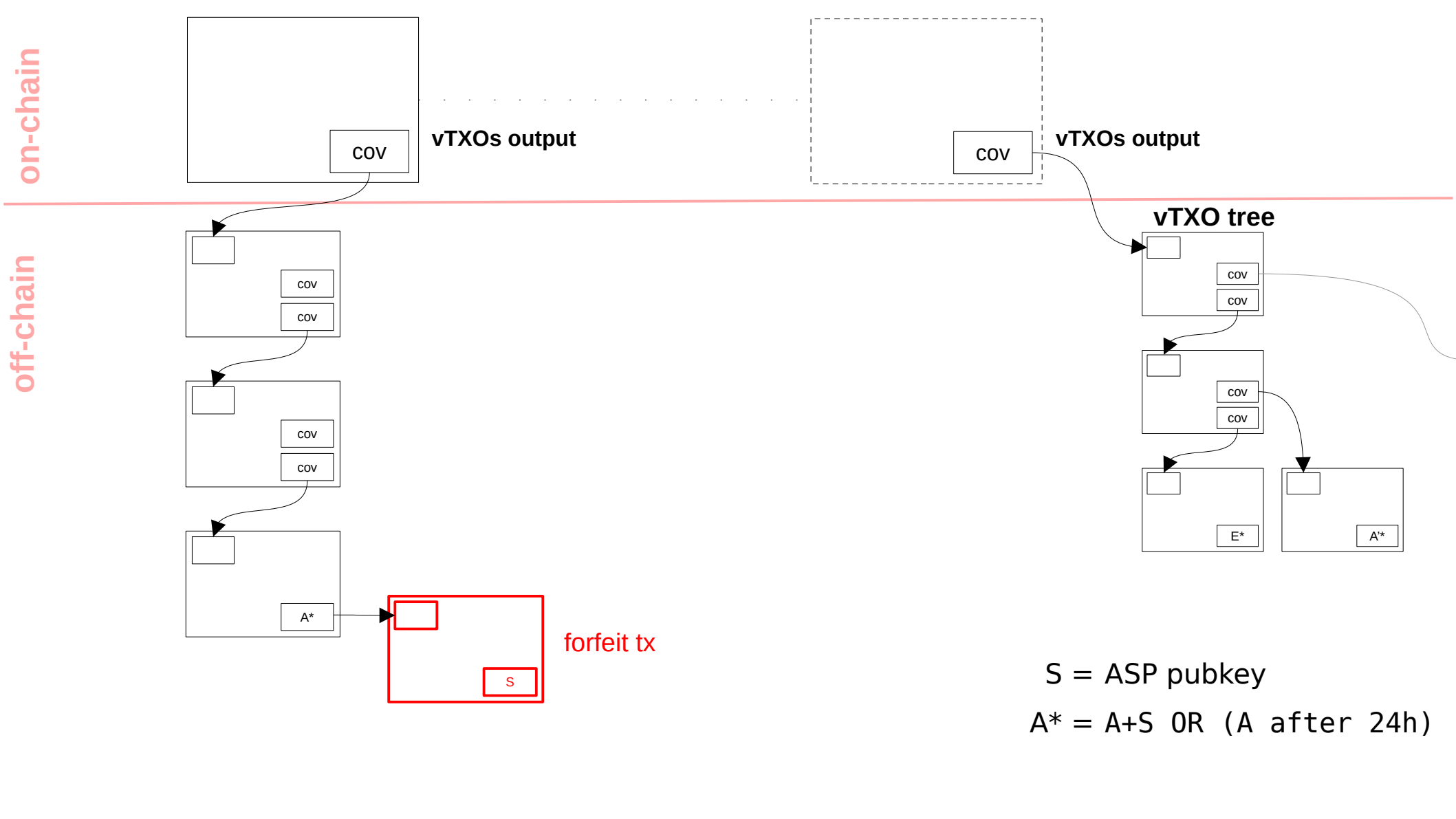
off-chain

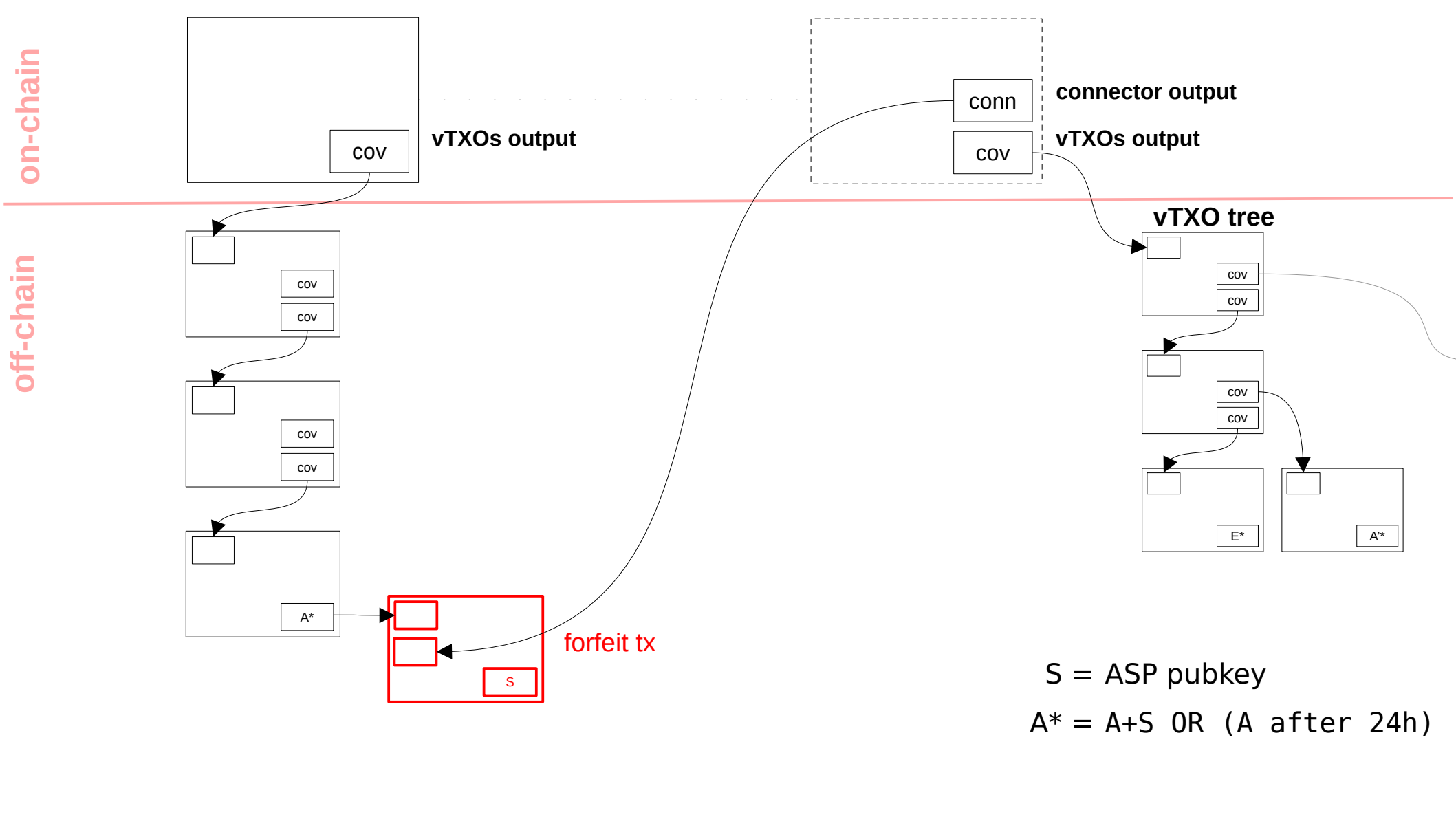


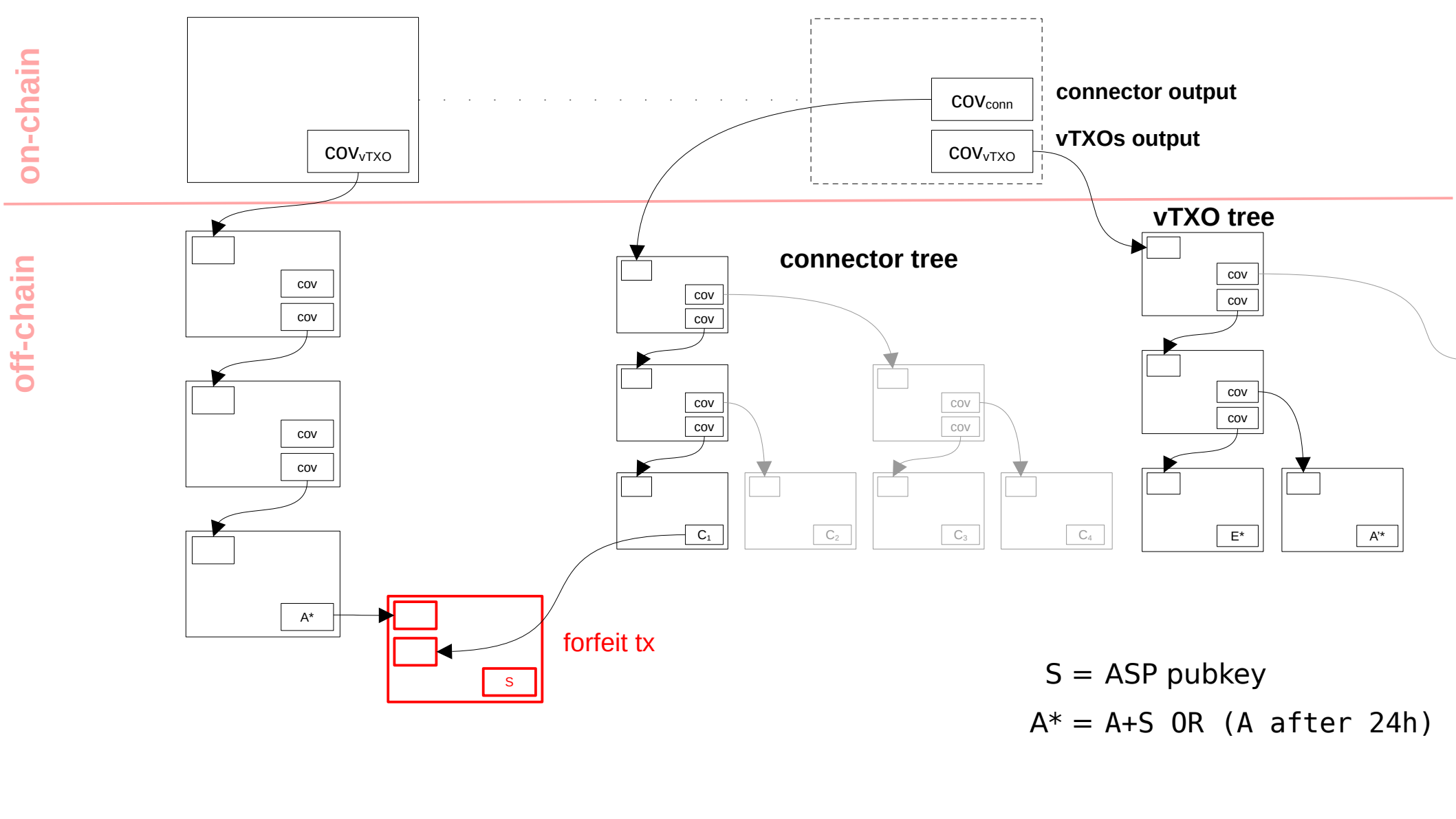
S = ASP pubkey

A* = A+S OR (A after 24h)


on-chain
off-chain
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
cov
A
B
C
D
E
A'
F
G
“
”
A
E
A’

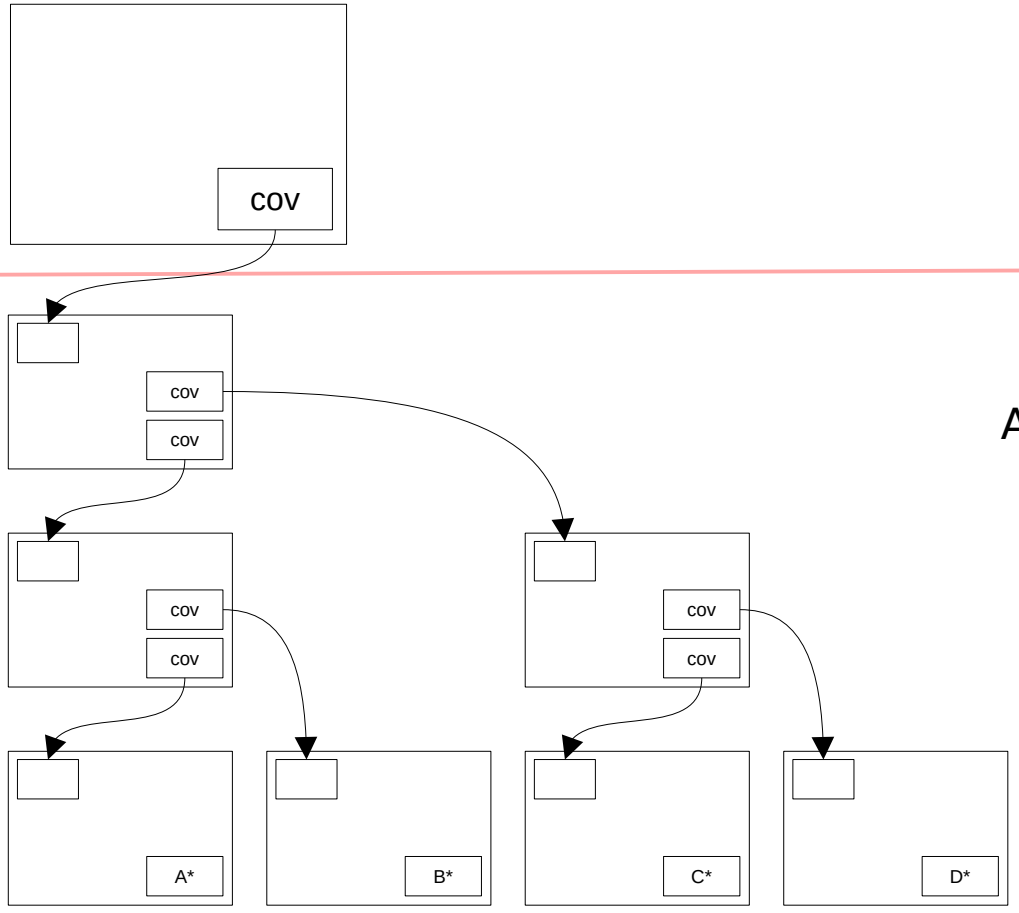

on-chain
off-chain
COV
vTXOs output
COV
vTXOs output
vTXO tree
COV
COV
COV
COV
COV
COV
COV
COV
A*
E*
A'*
forfeit tx
S
S = ASP pubkey
A* = A+S OR (A after 24h)


on-chain
off-chain
cov
vTXOs output
conn
connector output
cov
vTXOs output
vTXO tree
cov
cov
cov
cov
cov
cov
cov
cov
A*
E*
A'*
forfeit tx
S
S = ASP pubkey
A* = A+S OR (A after 24h)


on-chain
off-chain
COV_vTXO
COV_conn
connector output
vTXOs output
vTXO tree
connector tree
cov
A*
C1
C2
C3
C4
E*
A'*
forfeit tx
S
S = ASP pubkey
A* = A+S OR (A after 24h)

on-chain

off-chain

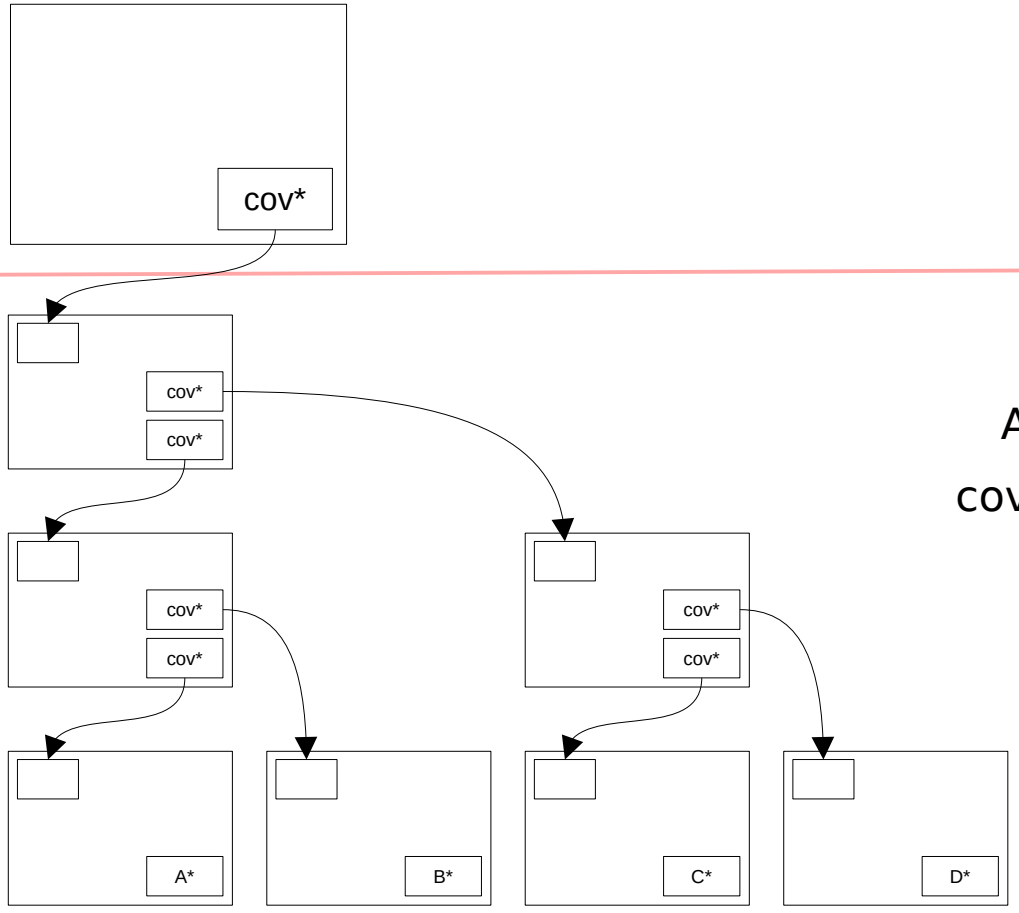


S = ASP pubkey

A* = A+S OR (A after 24h)

on-chain

off-chain

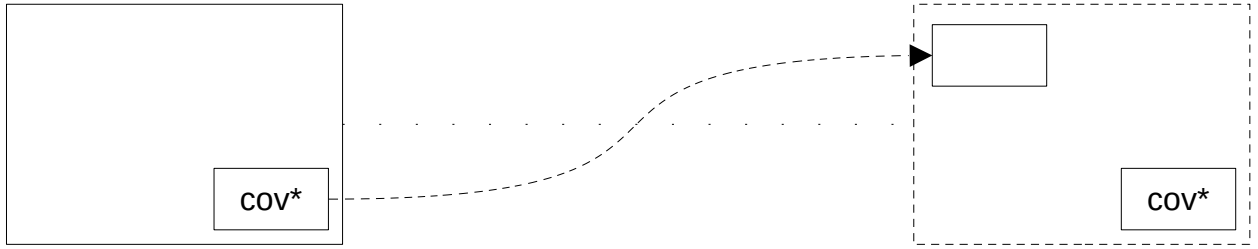


S = ASP pubkey

A* = A+S OR (A after 24h)

cov* = cov OR (S after 14d)

on-chain



off-chain

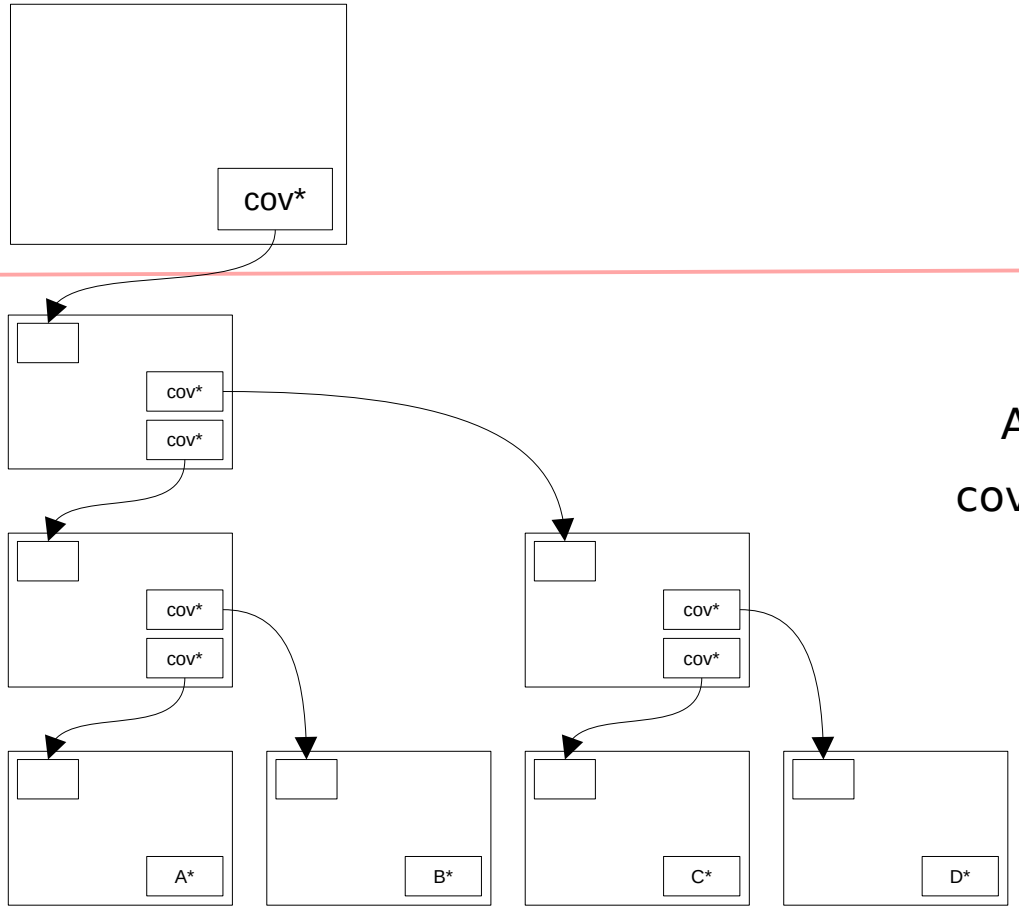
S = ASP pubkey

A* = A+S OR (A after 24h)

cov* = cov OR (S after 14d)

on-chain

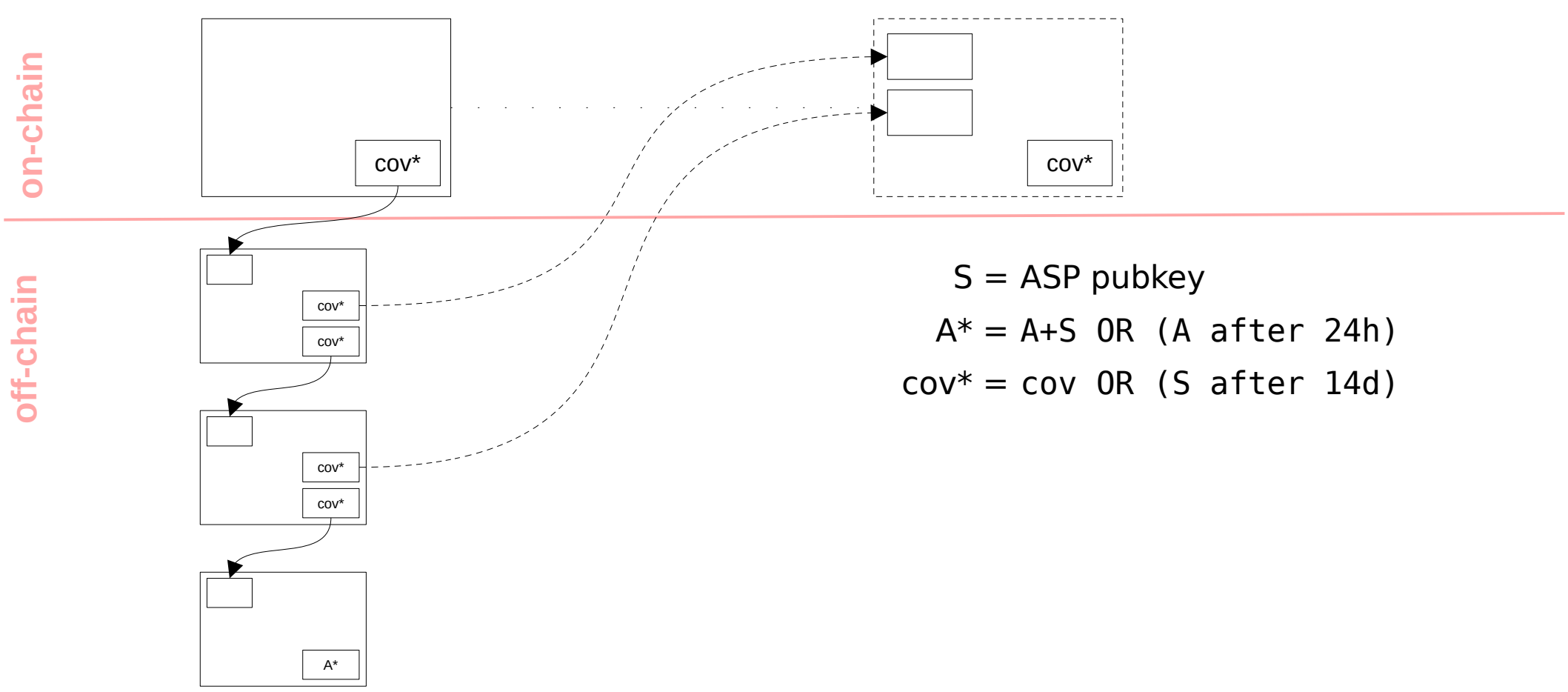
off-chain



S = ASP pubkey

A* = A+S OR (A after 24h)

cov* = cov OR (S after 14d)

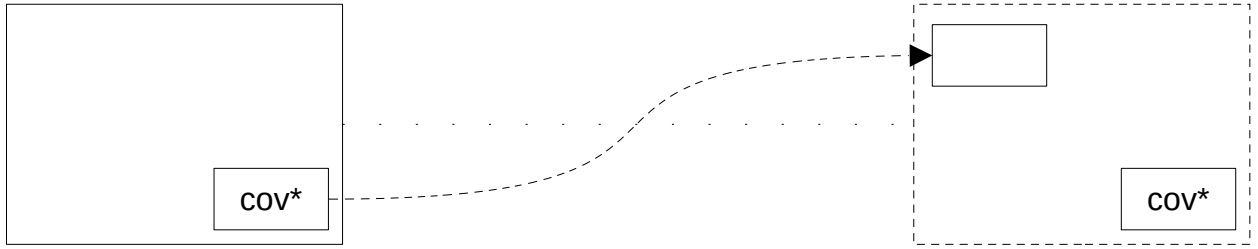


S = ASP pubkey

A* = A+S OR (A after 24h)

cov* = cov OR (S after 14d)

on-chain

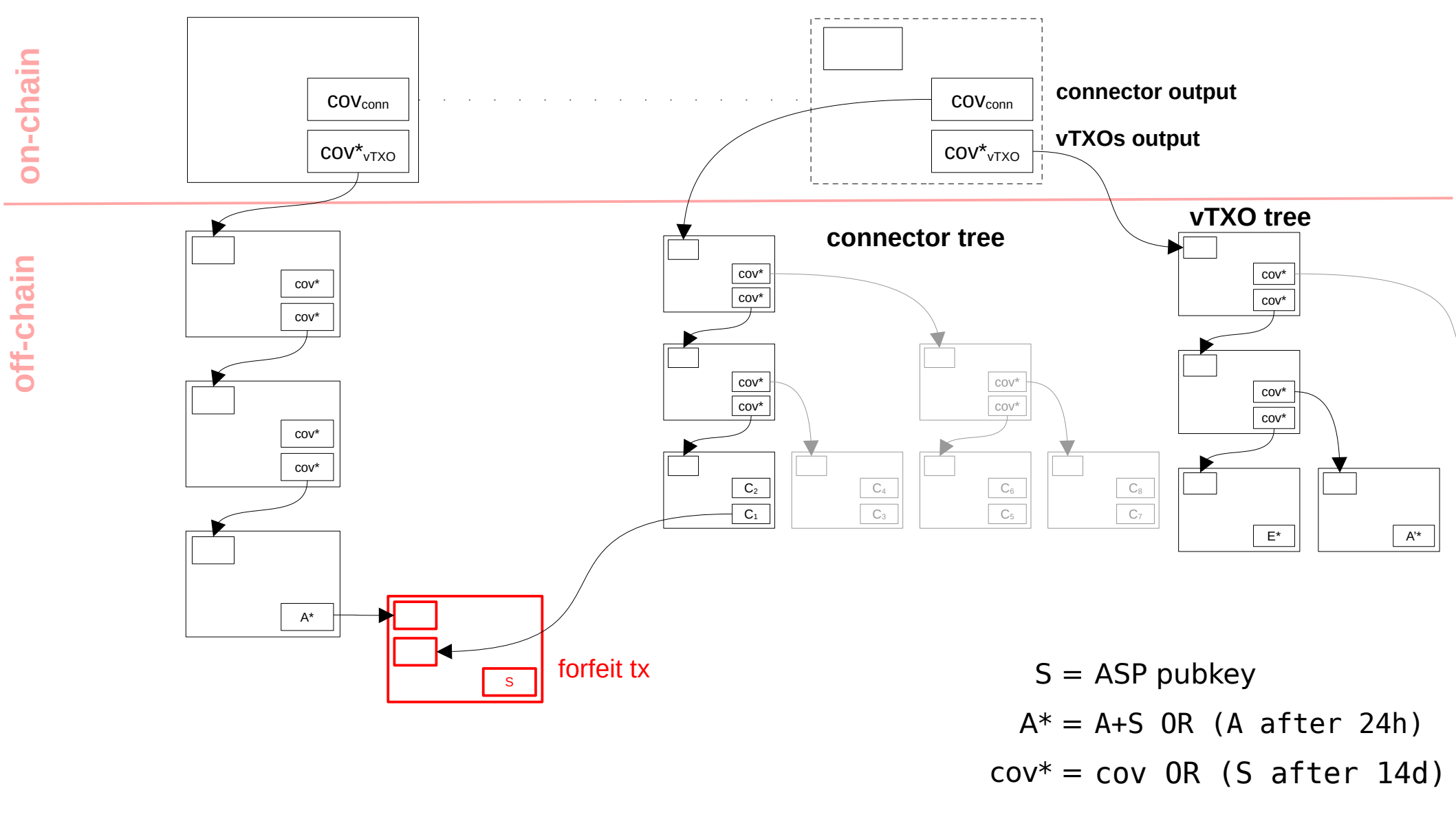


off-chain

S = ASP pubkey

A* = A+S OR (A after 24h)

cov* = cov OR (S after 14d)


on-chain
off-chain
COVconn
COV*vTXO
COVconn
COV*vTXO
connector output
vTXOs output
connector tree
vTXO tree
cov*
C2
C1
C4
C3
C6
C5
C8
C7
E*
A'*
A*
S
forfeit tx
S = ASP pubkey
A* = A+S OR (A after 24h)
cov* = cov OR (S after 14d)

What is (an) Ark?

- series of “Ark rounds”
  - → atomic spending VTXOs to create new ones
  - → one on-chain Bitcoin tx per round
- single service provider: “ASP”
  - → coordinates rounds & provides liquidity
  - → users always have unilateral exit

Why is this cool?

- UTXO-style off-chain txs

  → can make HTLCs for Lightning payments

- minimal on-chain footprint
- only client-server interactions

  → simpler than Lightning

- anyone can receive (no liquidity required!)

You said Lightning?

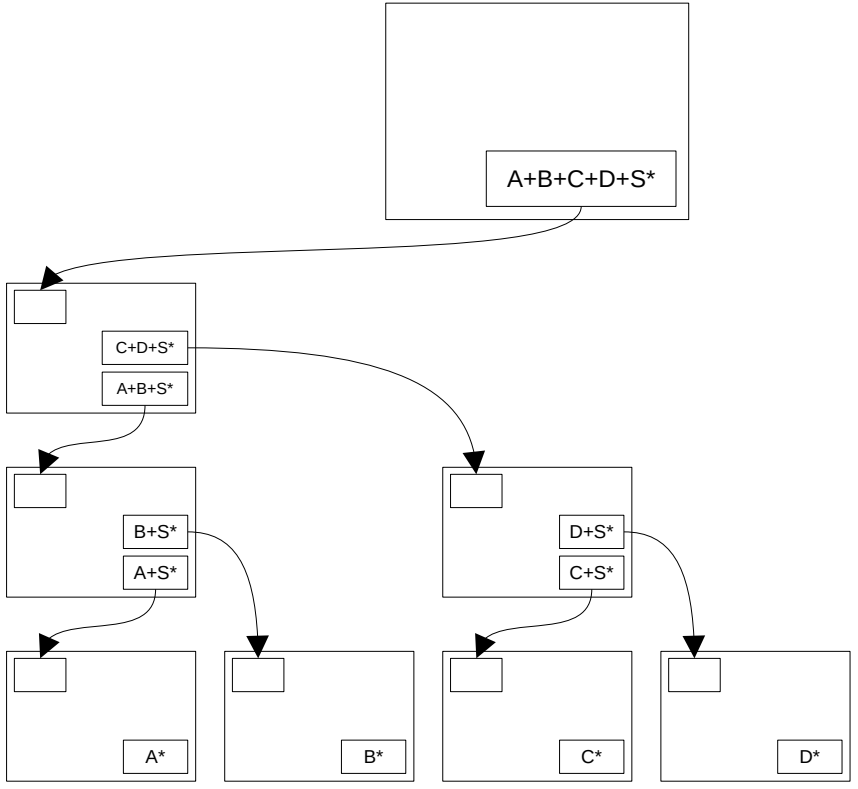
- make Lightning payments from Ark
  - → HTLC as VTXO
  - → ASP can function as LSP
- create Lightning channels inside Ark
  - → channel factory”: commitment VTXOs
  - → cheap channels with expiry

But what about?

- covenants
- liquidity requirement of ASP
- unilateral exit cost

clArk: covenant-less Ark

- pre-signed txs instead of covenants
  - → all receivers co-sign with ASP
  - → requires receivers online



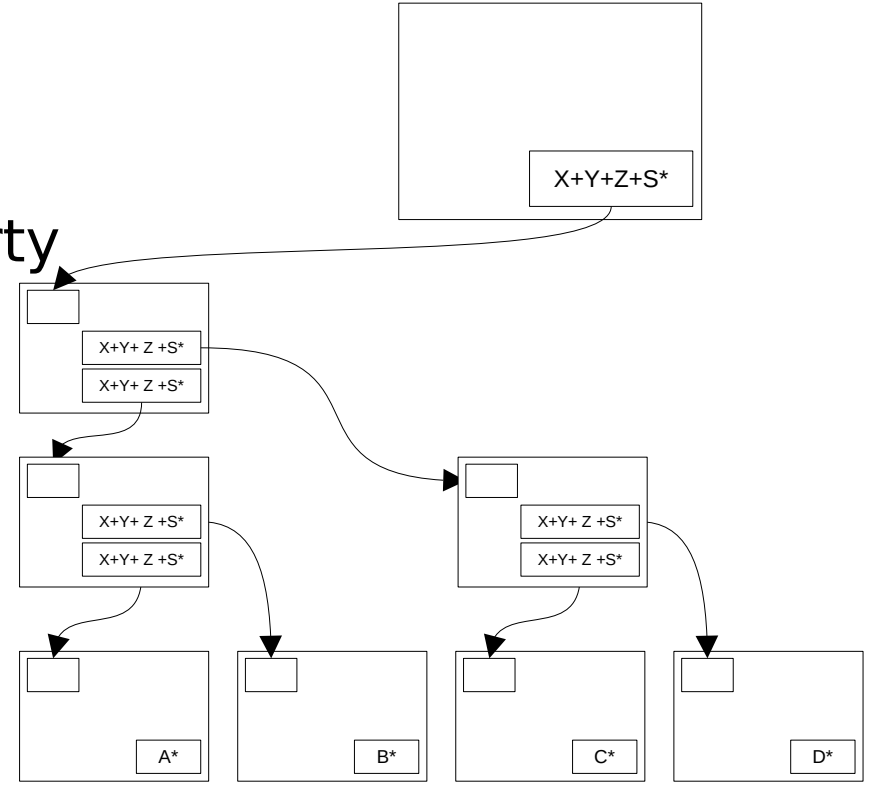
S = ASP pubkey

A+B+S* = A+B+S OR (S after 14d)

A* = A+S OR (A after 24h)

clArk: covenant-less Ark

- pre-signed txs instead of covenants
  - → all senders co-sign with ASP
  - → senders are already online
- secure as long as single honest party (N-of-N)
- possible on Bitcoin today



S = ASP pubkey

A+B+S* = A+B+S OR (S after 14d)

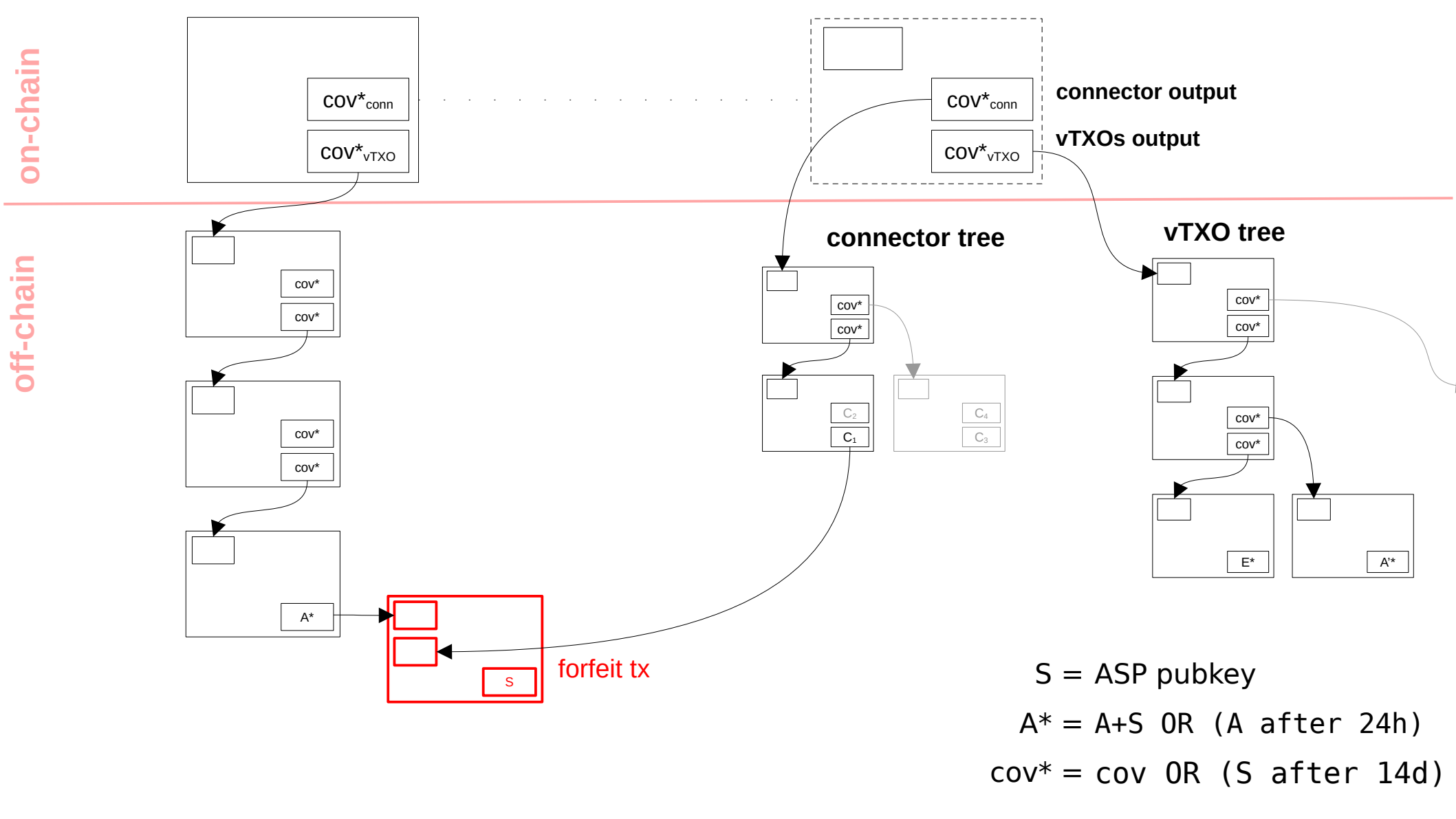
A* = A+S OR (A after 24h)

Liquidity

- function of user activity, not deposits
- when a VTXO gets spent, ASP provides liquidity for the value of the VTXO until it expires
- incentivize users to spend VTXOs closest to expiring

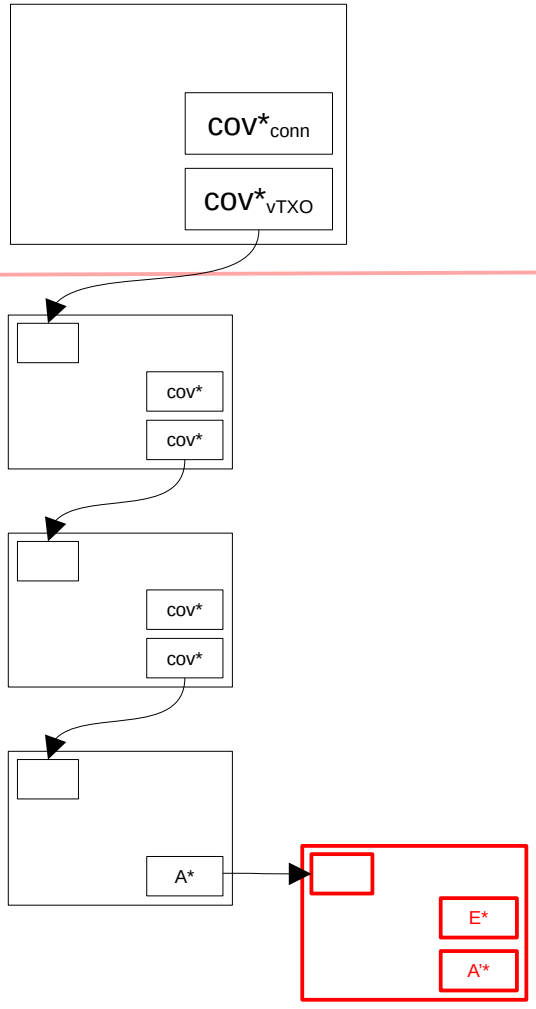
Out-of-Round Payments

- liquidity needed to spend in round
- we can send a VTXO directly
  → without participating in round
- “Somsen Shortcut”


on-chain
off-chain
COV*conn
COV*vTXO
connector output
vTXOs output
connector tree
vTXO tree
cov*
C2
C1
C4
C3
E*
A'*
A*
S
forfeit tx
S = ASP pubkey
A* = A+S OR (A after 24h)
cov* = cov OR (S after 14d)

on-chain

off-chain

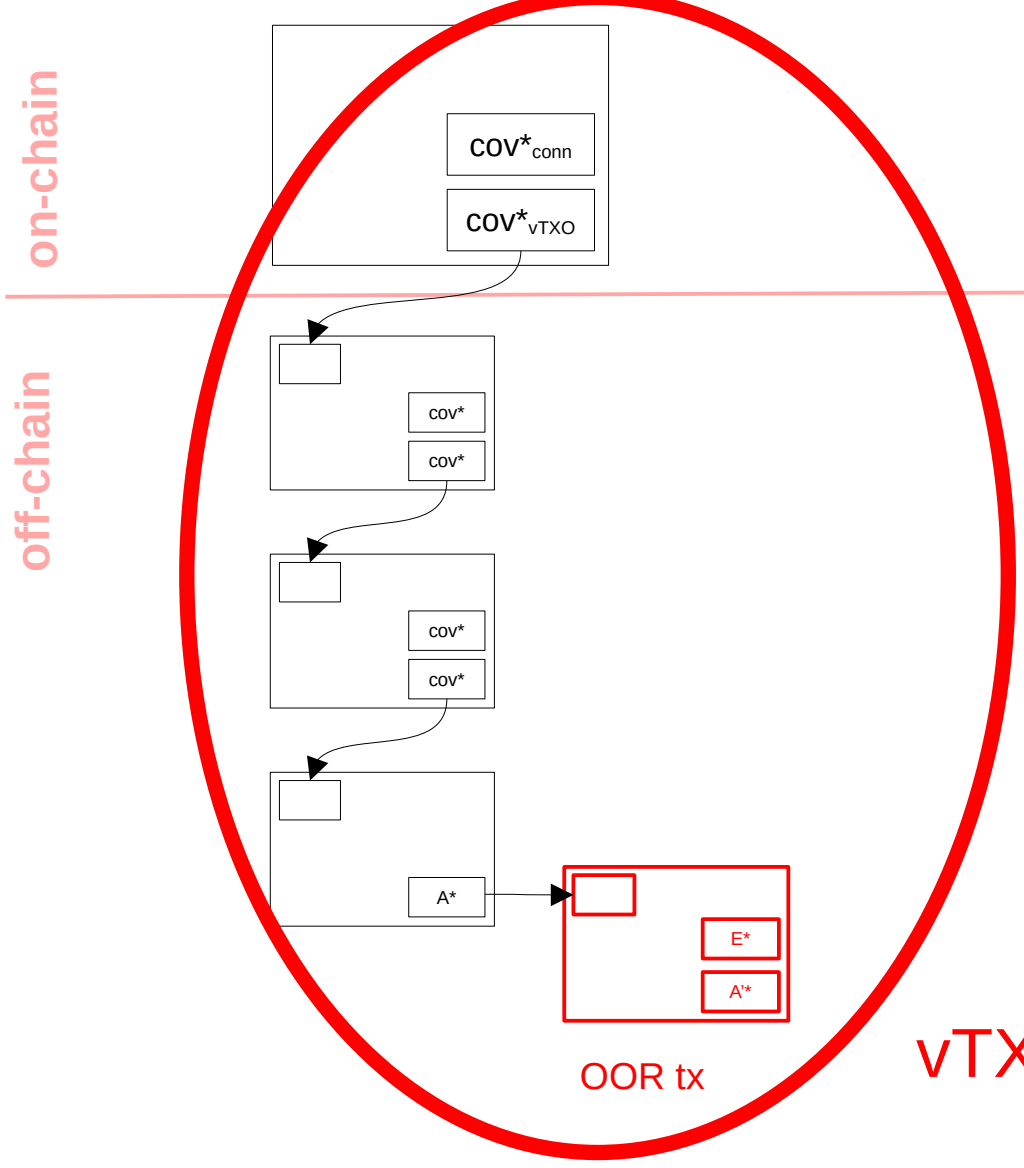


OOR tx

S = ASP pubkey

A* = A+S OR (A after 24h)

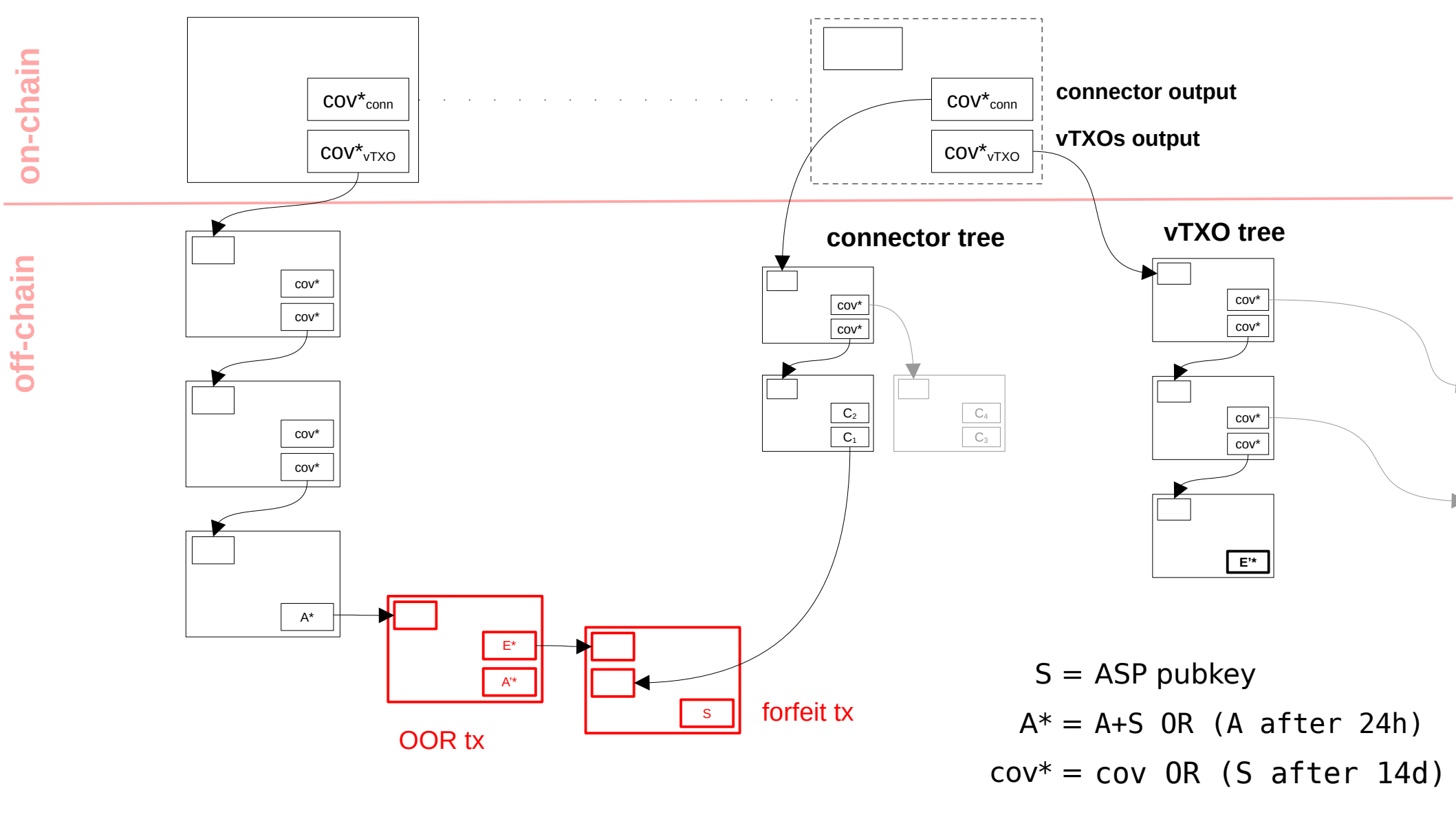
cov* = cov OR (S after 14d)

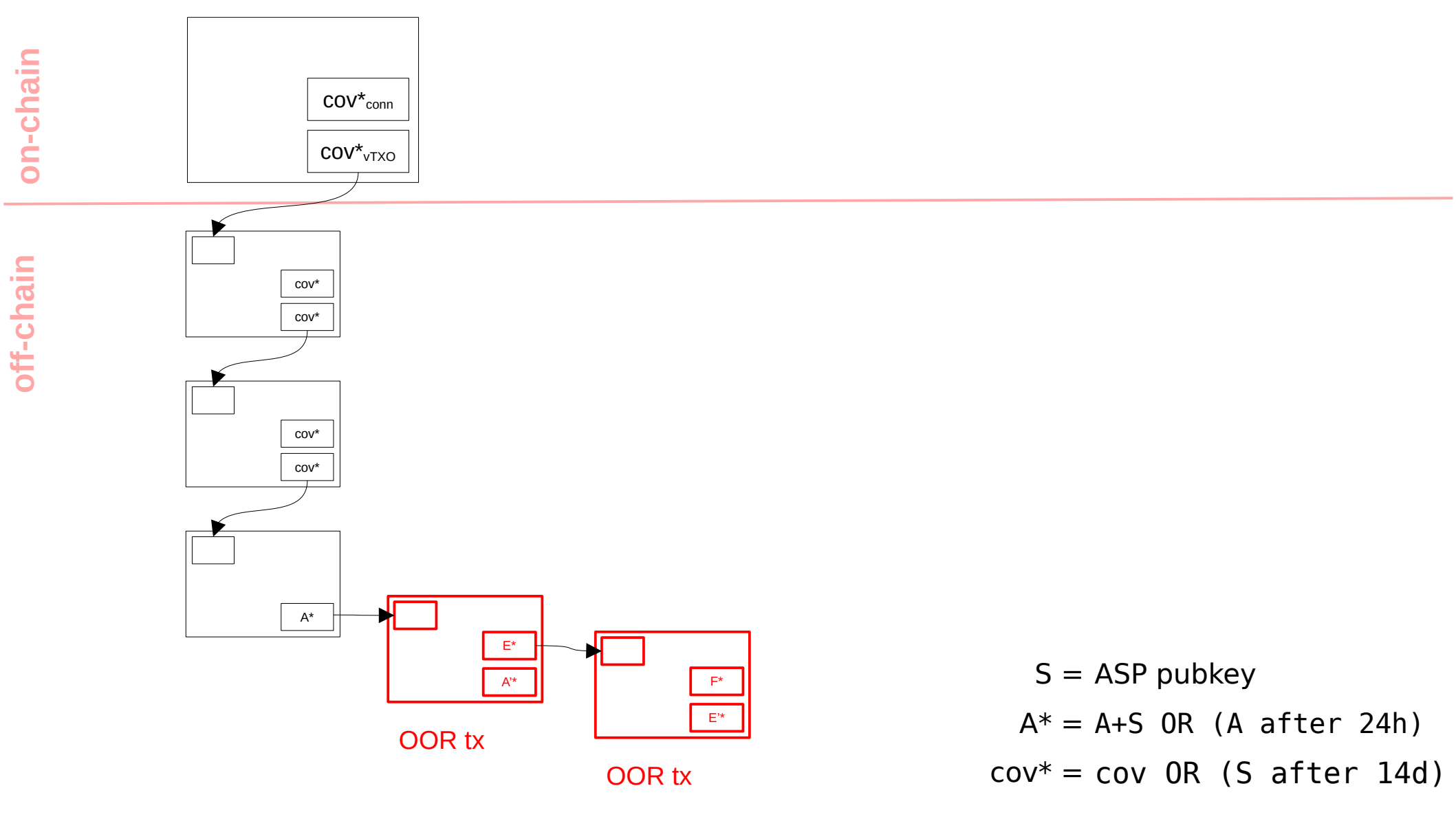


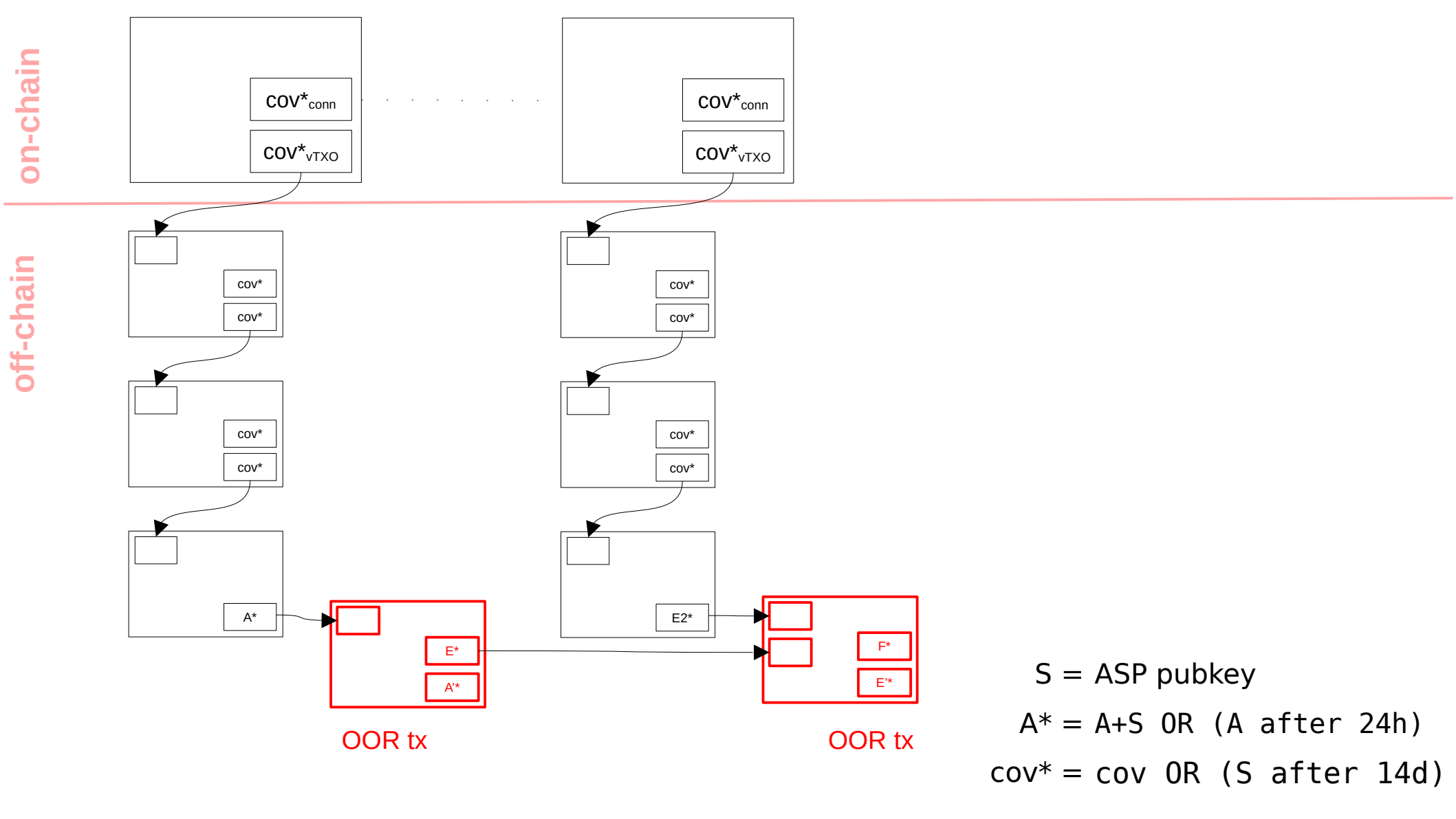
S = ASP pubkey

A* = A+S OR (A after 24h)

cov* = cov OR (S after 14d)


on-chain
off-chain
COV*conn
COV*vTXO
connector output
vTXOs output
connector tree
vTXO tree
cov*
C2
C1
C4
C3
E'*
A*
E*
A'*
S
OOR tx
forfeit tx
S = ASP pubkey
A* = A+S OR (A after 24h)
cov* = cov OR (S after 14d)


on-chain
off-chain
cov*conn
cov*vTXO
cov*
cov*
cov*
cov*
A*
E*
A'*
F*
E'*
OOR tx
OOR tx
S = ASP pubkey
A* = A+S OR (A after 24h)
cov* = cov OR (S after 14d)


on-chain
off-chain
cov*conn
cov*vTXO
cov*conn
cov*vTXO
cov*
cov*
cov*
cov*
cov*
cov*
cov*
cov*
A*
E2*
E*
A'*
F*
E'*
OOR tx
OOR tx
S = ASP pubkey
A* = A+S OR (A after 24h)
cov* = cov OR (S after 14d)

Arkoor

- instant VTXO txs
- super cheap
  - → no liquidity fee
- opt-out of by cycling in next round
- save liquidity fee by holding OORs
- in cycles, senders == receivers → clArk++

Unilateral Exit

- ASP disappears → mass exit
  - → larger VTXO holders carry the cost for smaller
- ASP censors specific VTXOs
  - → unilateral exit
  - → cost can easily exceed value of VTXO
- watchtowers can oversee ASP behavior

State of the Ark

- impl of clArk on Bitcoin using MuSig2
- impl on Liquid using covenants

Thanks

- https://roose.io/presentations
- https://ark-protocol.org/
- Questions?