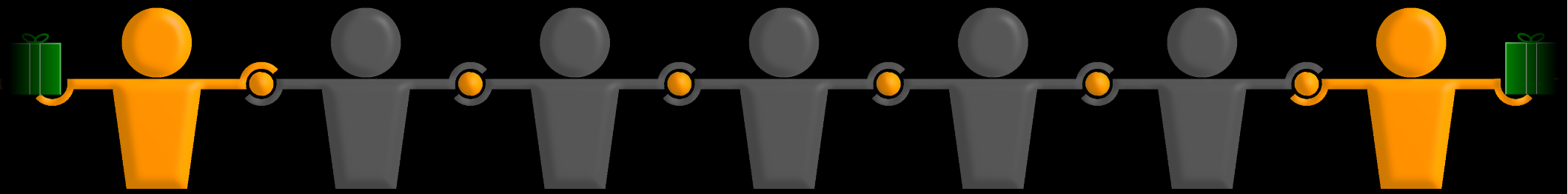
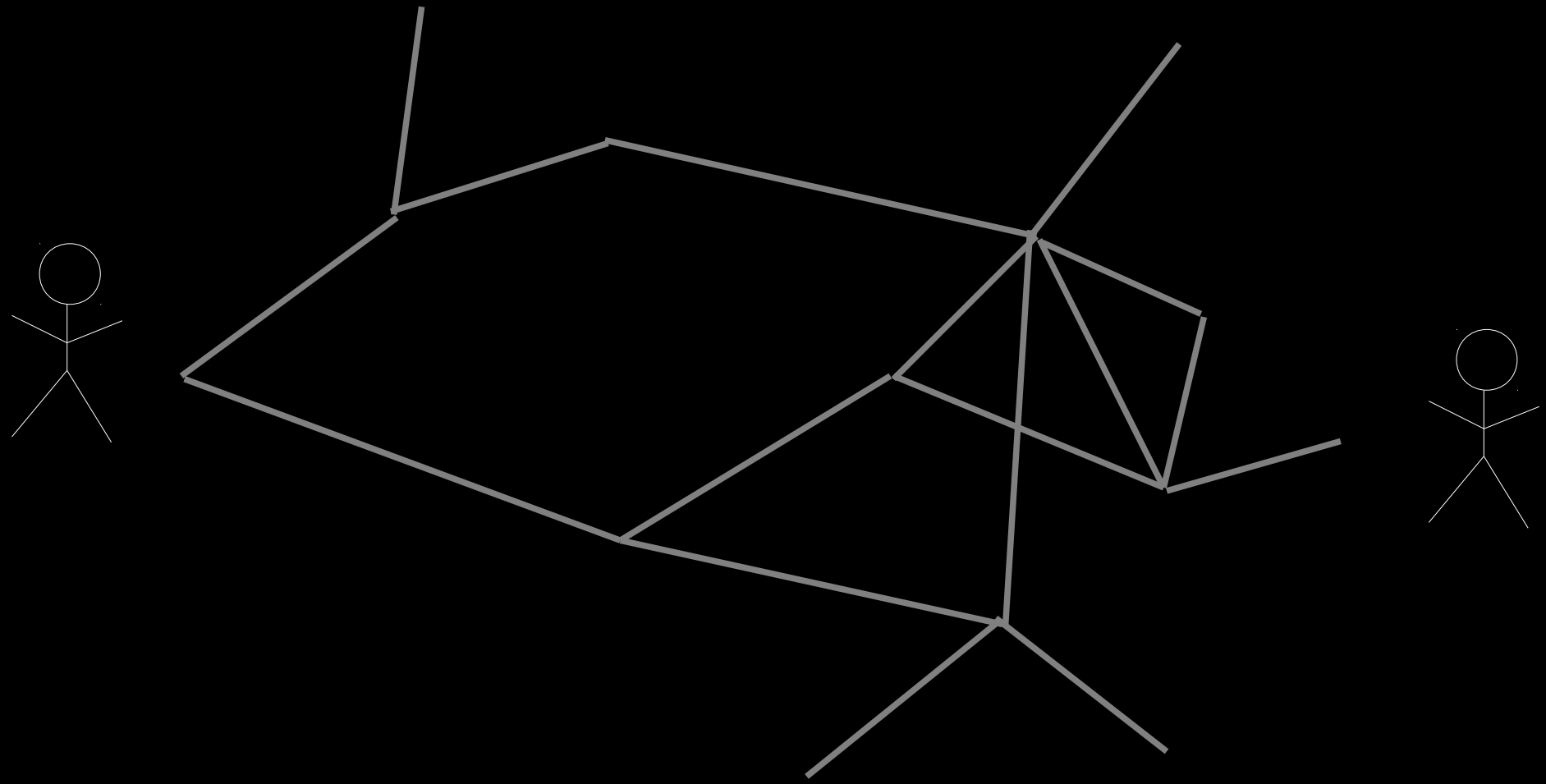
Amiko Pay

Experiences Working on Layer 2 Scalability Solutions

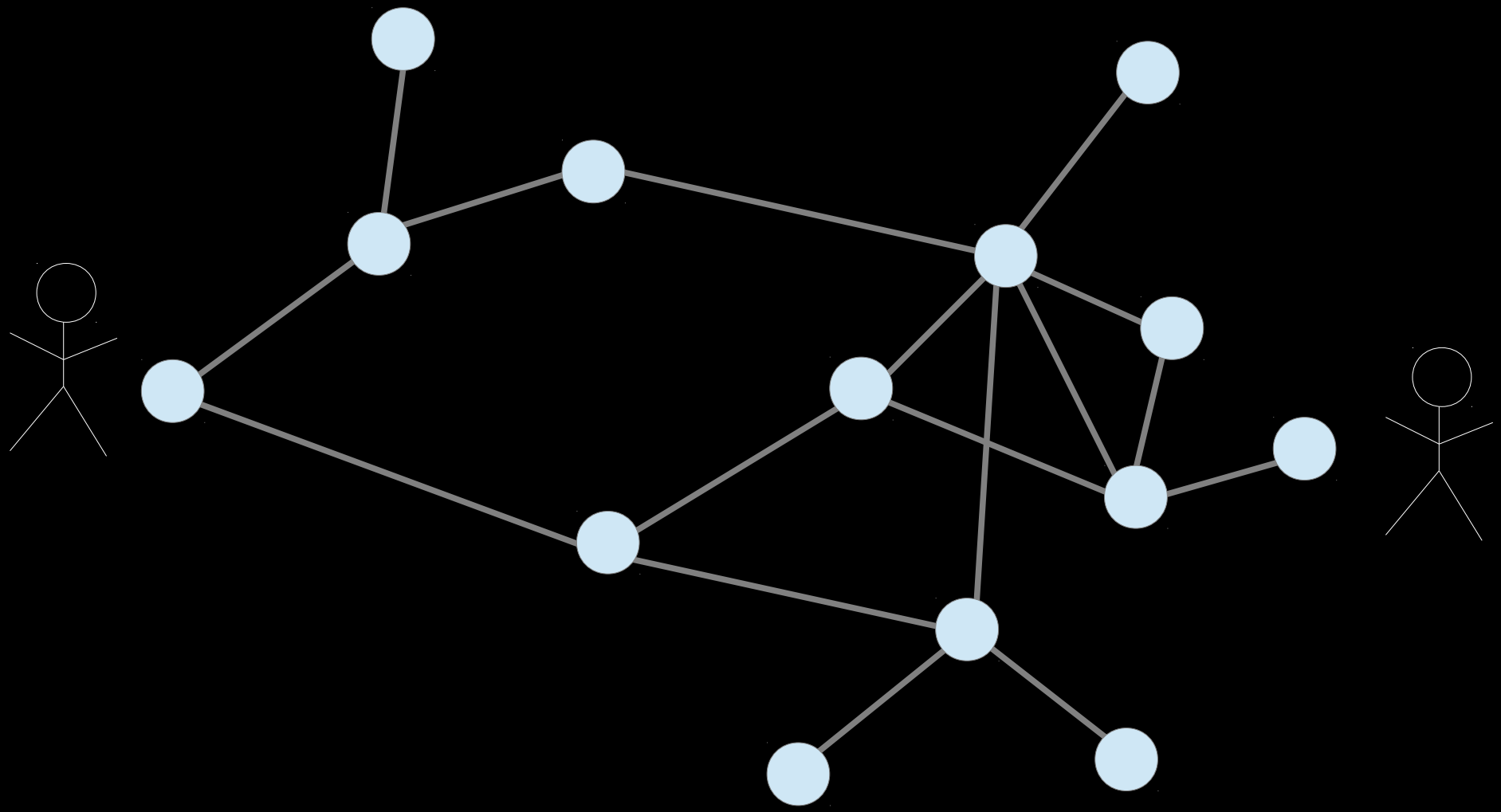
Corné Plooy



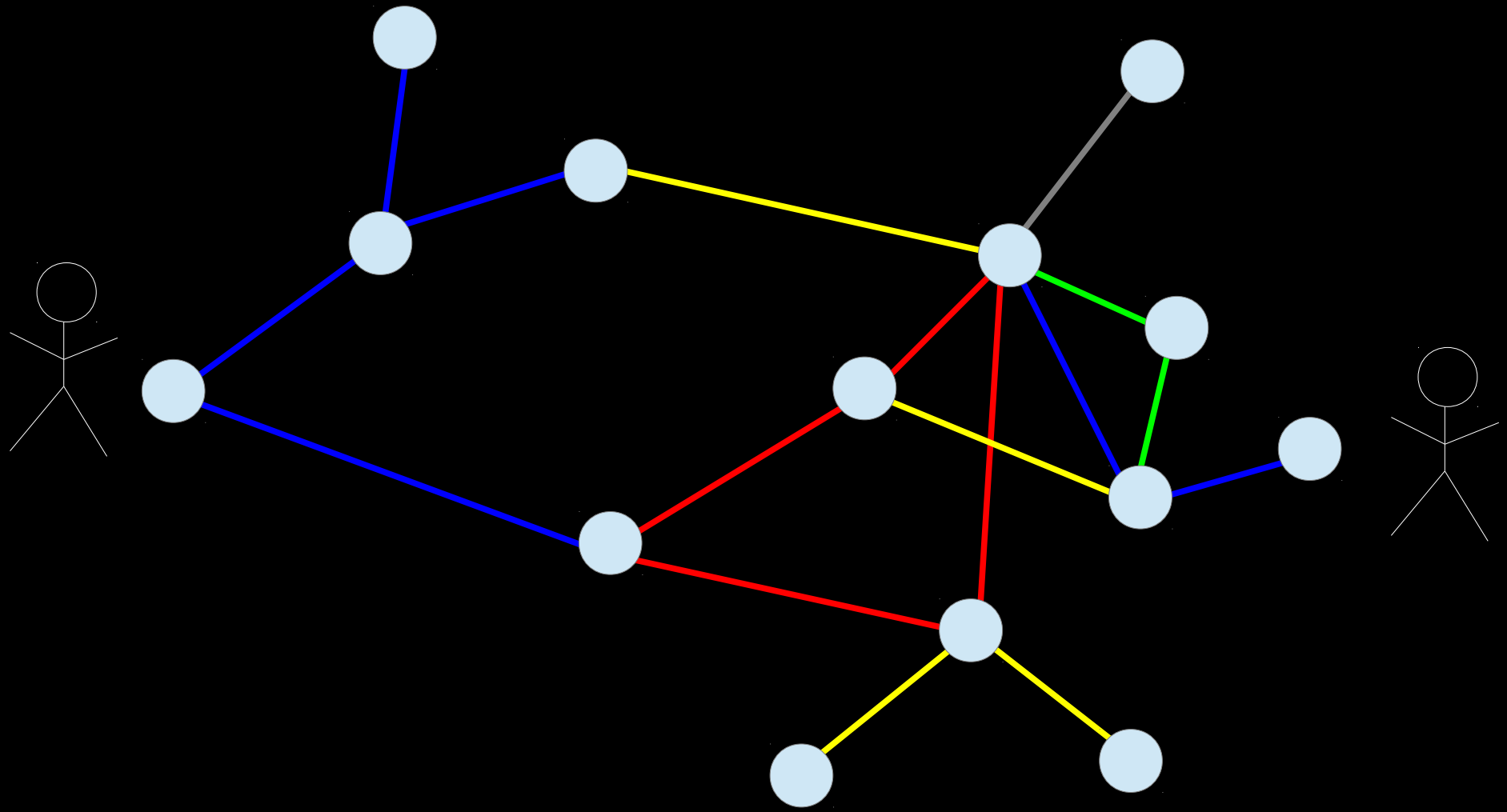
A network of channels



Nodes and Channels



Different channel types



Channel types

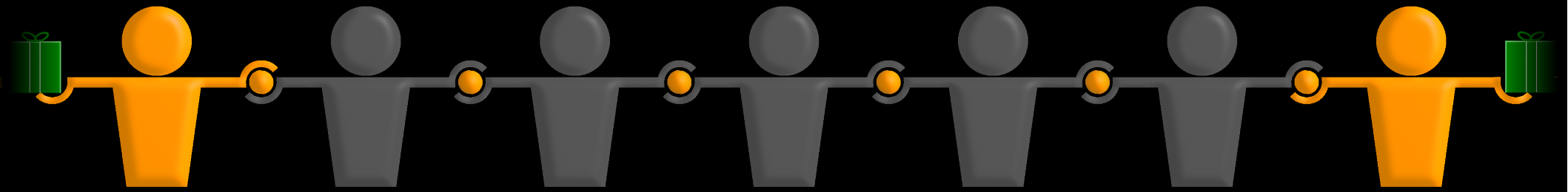
- Done: Ripple-style IOU “channel”
- Work In Progress: true Lightning channel
  - Lightning developers: side chain
  - Amiko Pay: emulation by escrow service

Issues / Thoughts

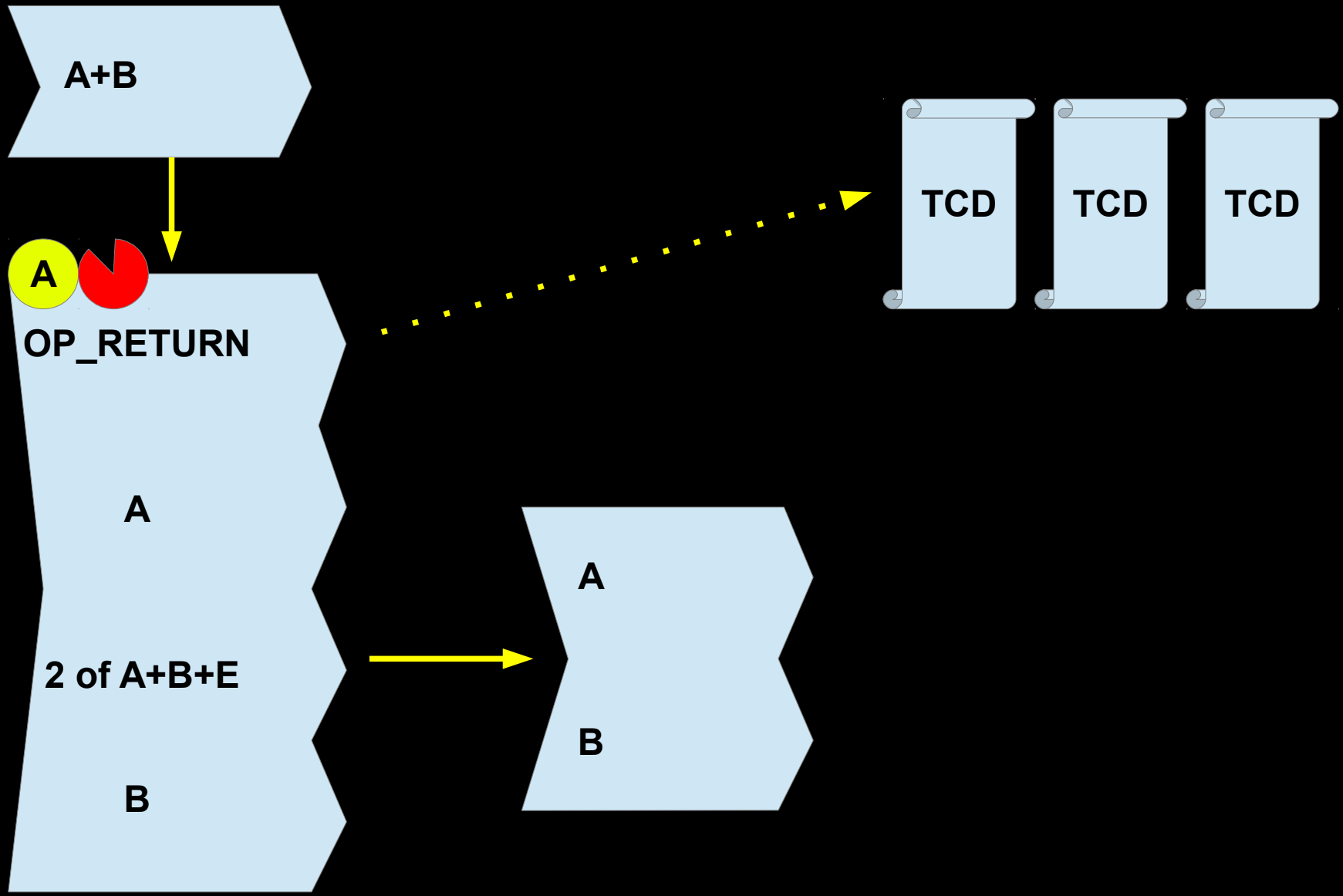
- Software complexity
- Transaction malleability
- The routing details really matter
- Economic modeling of the network

Thank you

https://github.com/cornwarecjp/amiko-pay



Emulation by escrow services



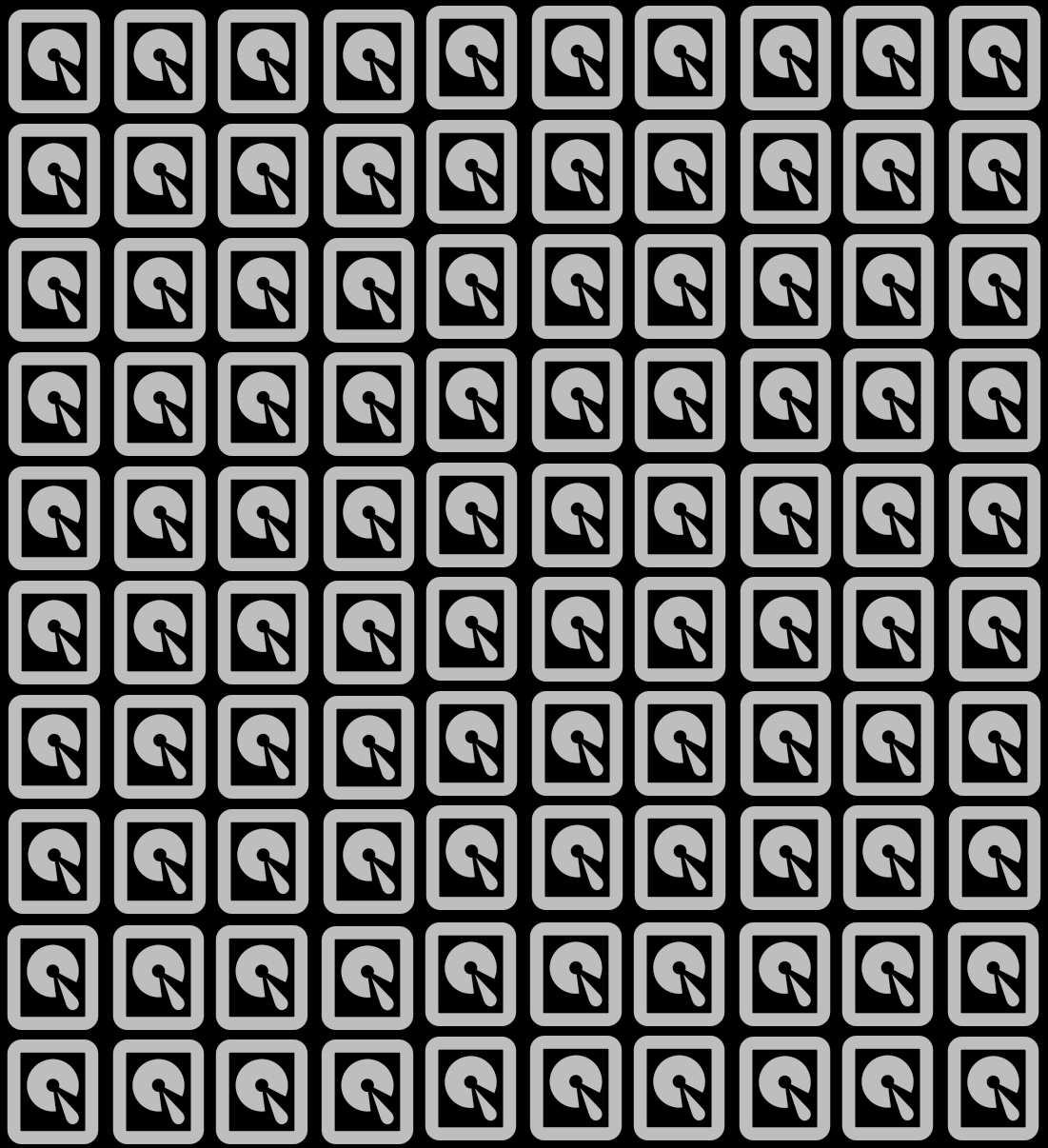
Required for Lightning Network

- Fixes for transaction malleability
- New SIGHASH types
- Lock time evaluation in script
  - "`OP_CHECKLOCKTIME`"
- Relative lock time
  - Optional: necessary for unlimited-lifetime micro-transaction channels

Very rough storage estimate

$10 000 savings per person
$1 per small transaction
____ /
10 000 unspent transactions per person
10 billion people
100 bytes per transaction
____ x
10^{16} bytes = 10 000 TByte = block chain size
(when using block chain pruning!)
Also: 10 GByte per block

100 x 100 TByte



Very rough storage estimate

1 channel per person
10 billion people
100 bytes per transaction
____ x
10^{12} bytes = 1 TByte = block chain size
→ **Factor 10 000 reduction!**

Also: 10 MByte per block
→ **Factor 1000 reduction!**