The Promises and Challenges of Ethereum Off-Chain Scaling

Xiaozhou Li
Celer Network

Internet (web 2.0) vs. Ethereum (web3.0)

global **information** transfer

One second:
70K Google searches,
millions of emails and
messages,
60TB data transfer
……

≫

global **value** transfer

One second:
10 transactions

Can Ethereum match the scale of Internet?

global **information** transfer

One second:
70K Google searches,
millions of emails and
messages,
60TB data transfer
……

≈

global **value** transfer

One second (goal):
??? transactions

Can Ethereum match the scale of Internet?

global **information** transfer

One second:
70K Google searches,
millions of emails and messages,
60TB data transfer
……

≈

global **value** transfer

One second (goal):
hundreds of millions or billions of transactions

Billions of TPS 😱

How could we ever get there?

dApps

blockchain

Scalability

Mass Adoption

On-chain Limit

Where we are

Efforts

dApps

off-chain scaling

blockchain

⚠ CAUTION
WORK IN PROGRESS

Scalability

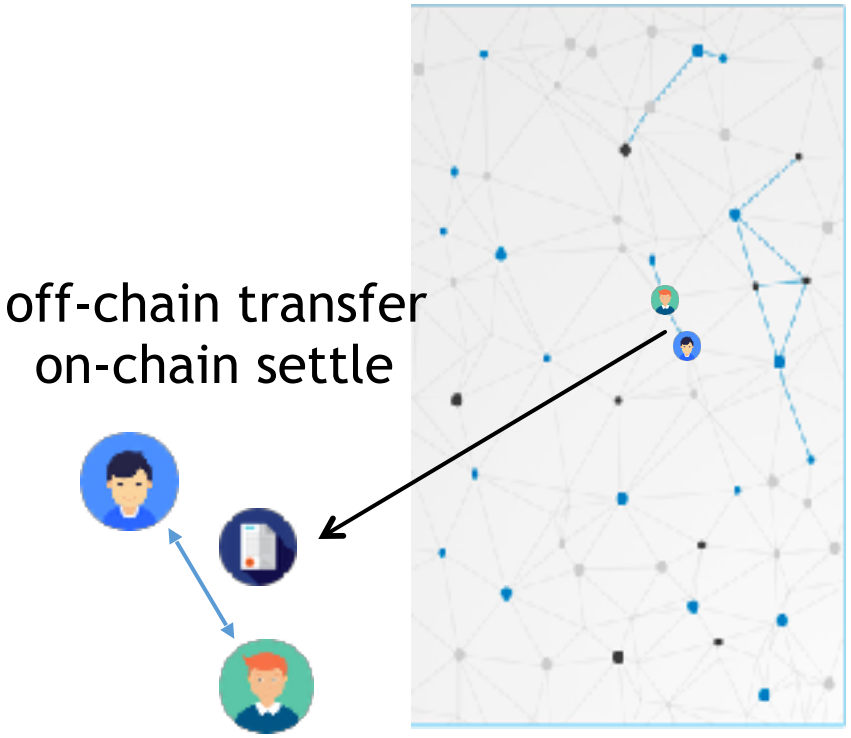
Mass Adoption

On-chain Limit

Where we are

Efforts

How does off-chain scaling work?



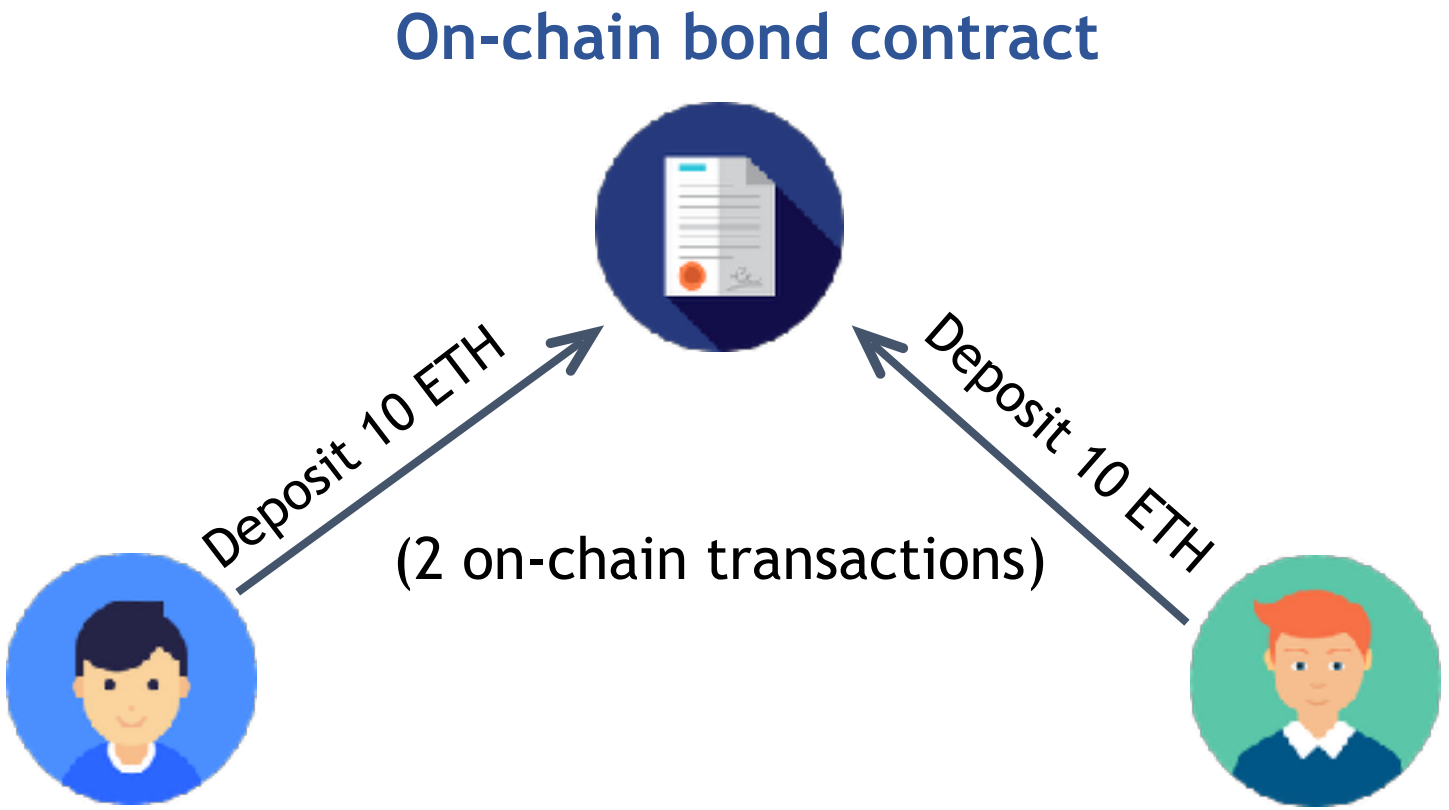
Concepts: state channel, sidechain
Only resort to on-chain consensus when absolutely necessary

Secure: blockchain acts as the “supreme court”

Fast: independent operations across nodes, fully scale-out

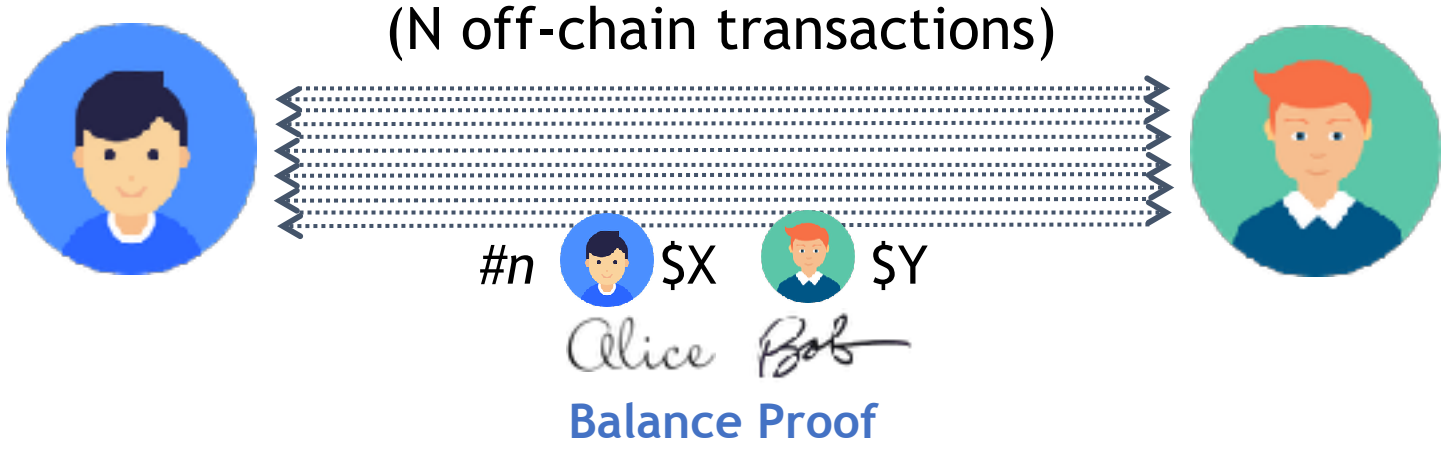
Private: most activities only seen by participants

Example: off-chain payment channel

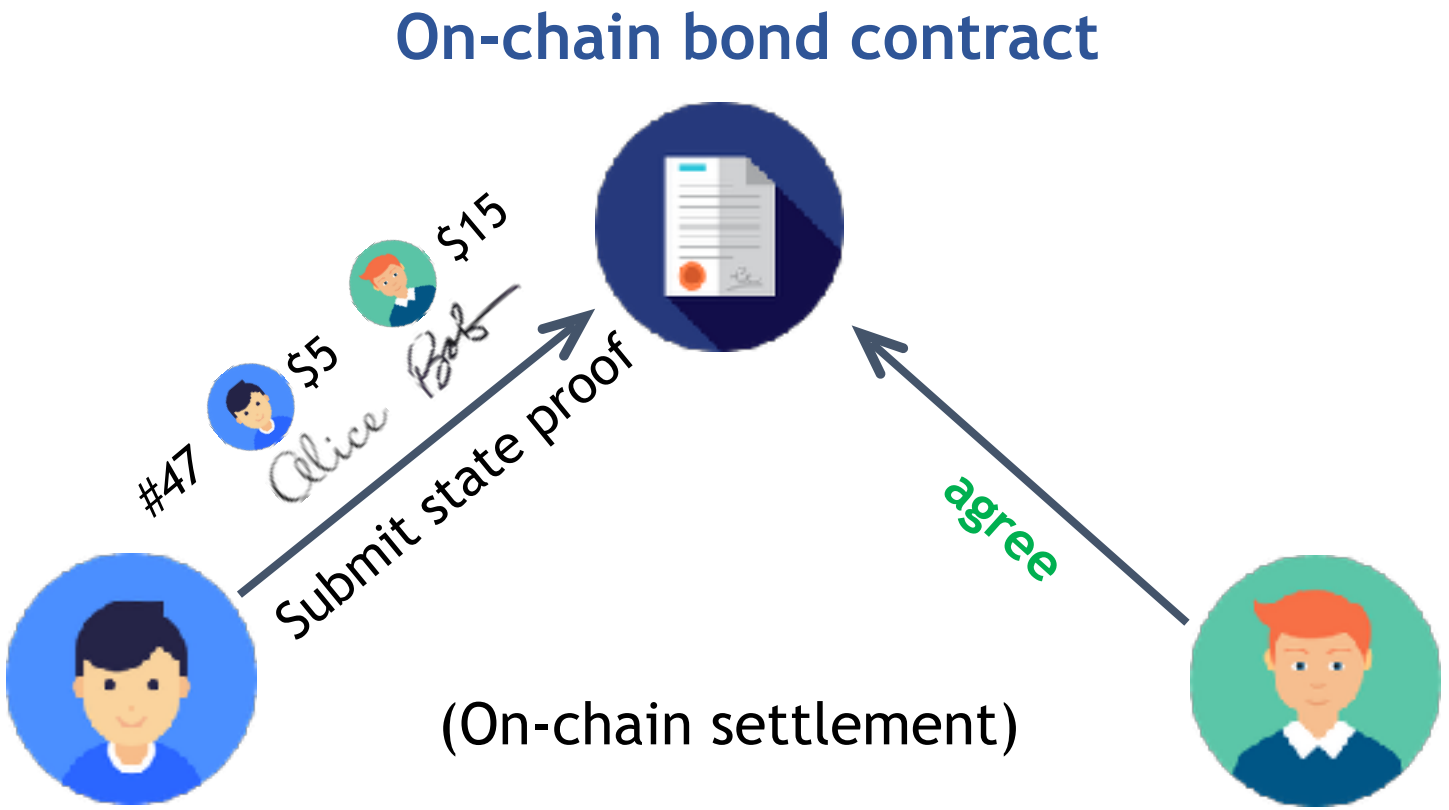


Example: off-chain payment channel

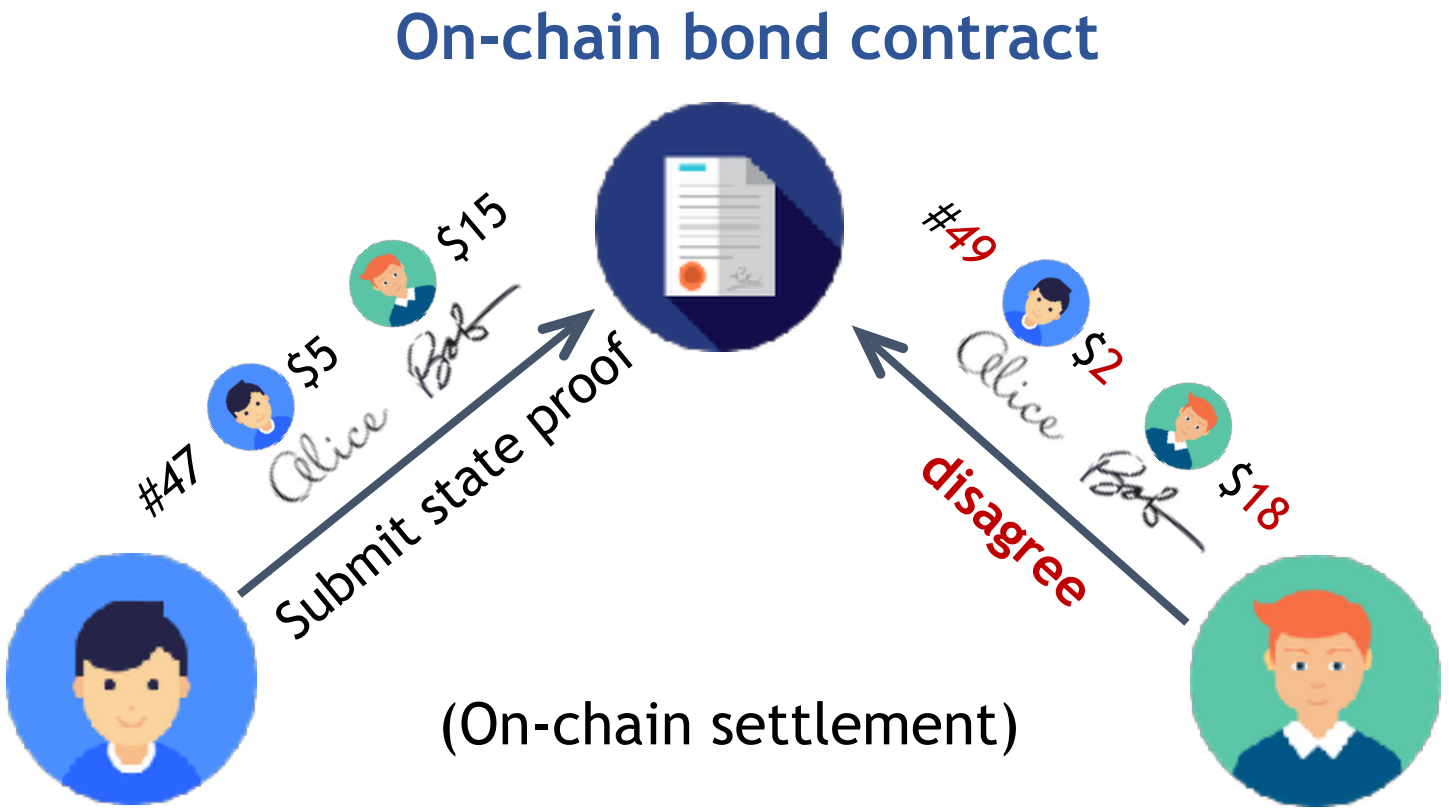
On-chain bond contract



Example: off-chain payment channel

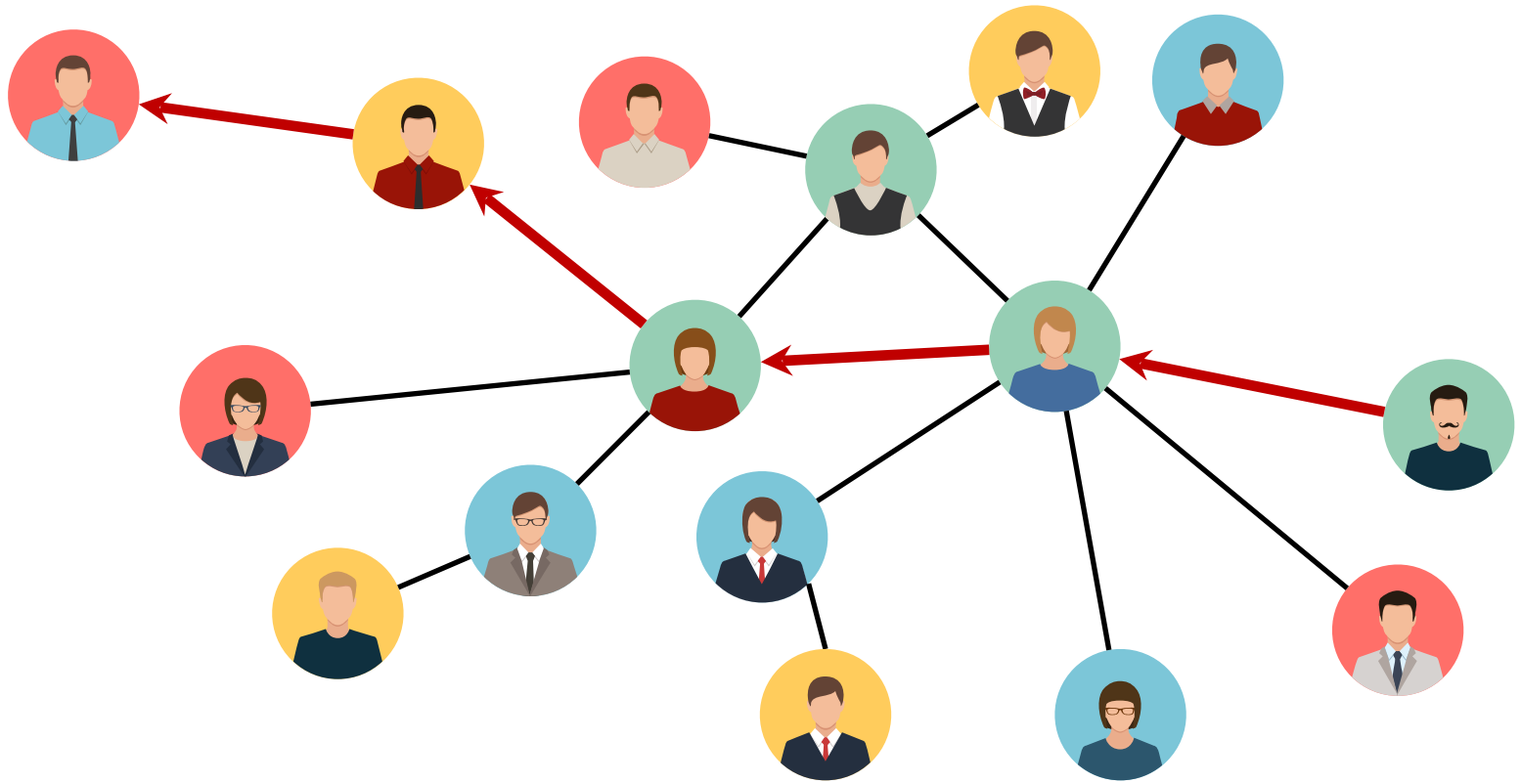


Example: off-chain payment channel

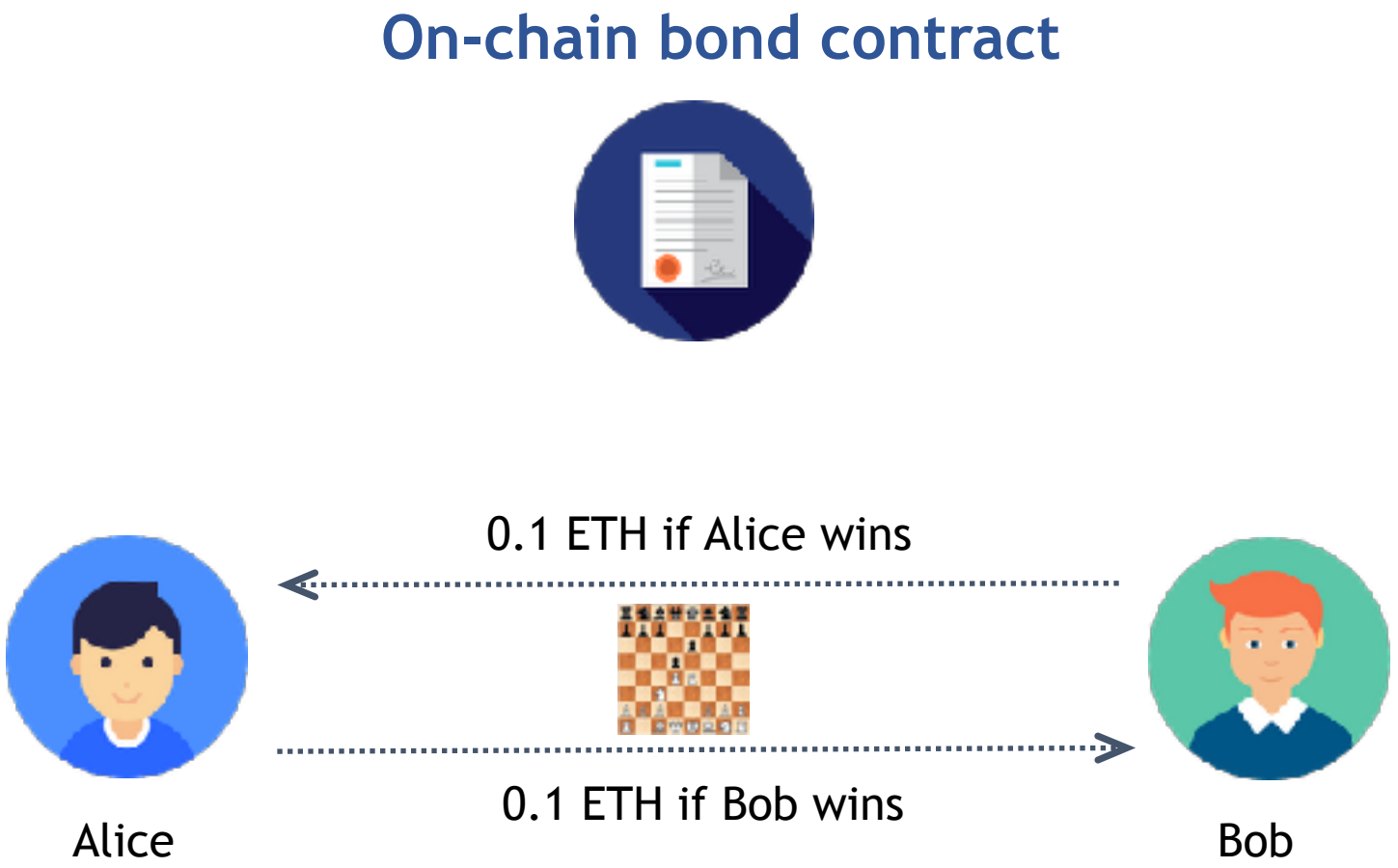


Example: off-chain payment network

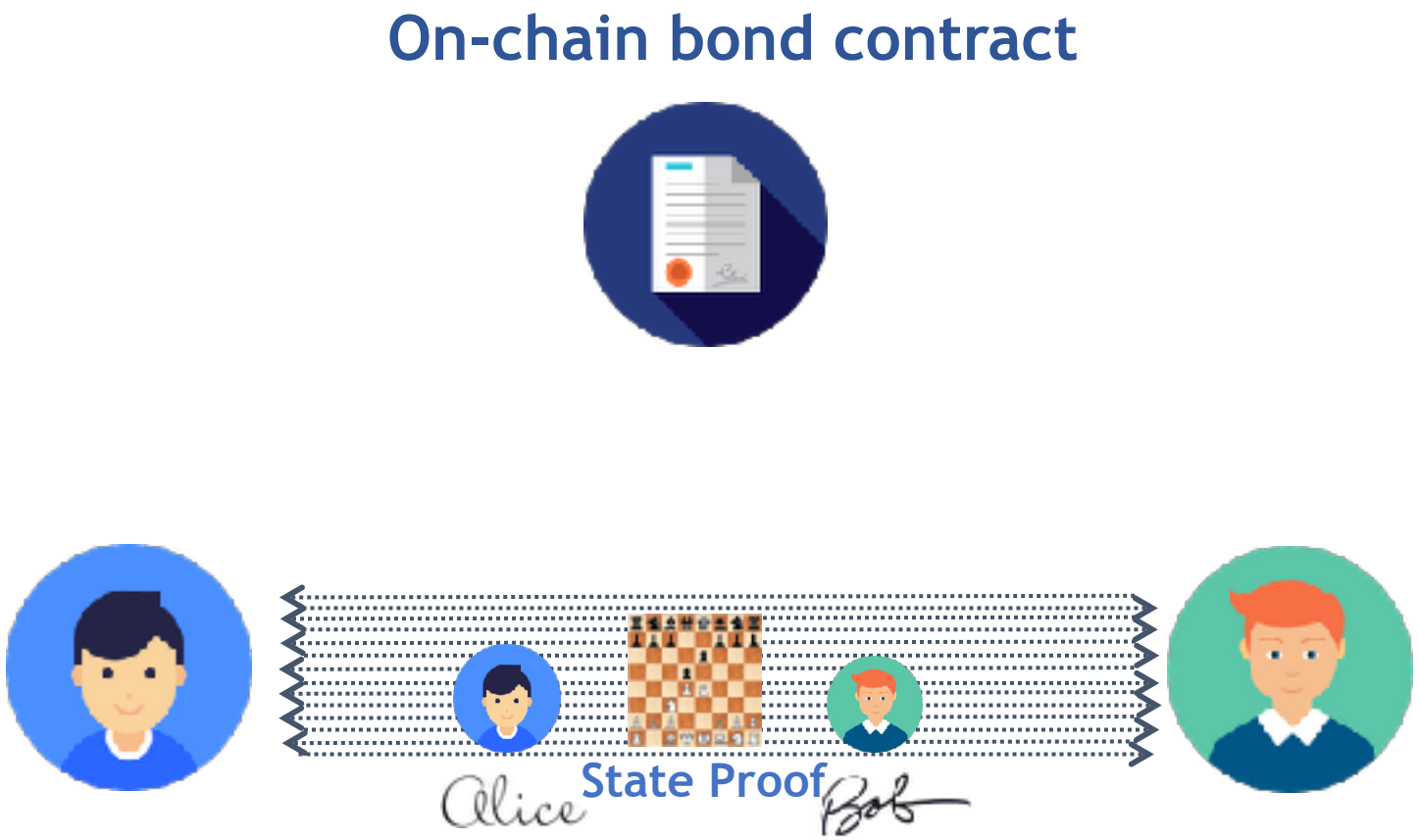
Open up great opportunities for ultra high throughput p2p micro-payments



Example: off-chain chess duel



Example: off-chain chess duel



Example: off-chain chess duel

On-chain bond contract

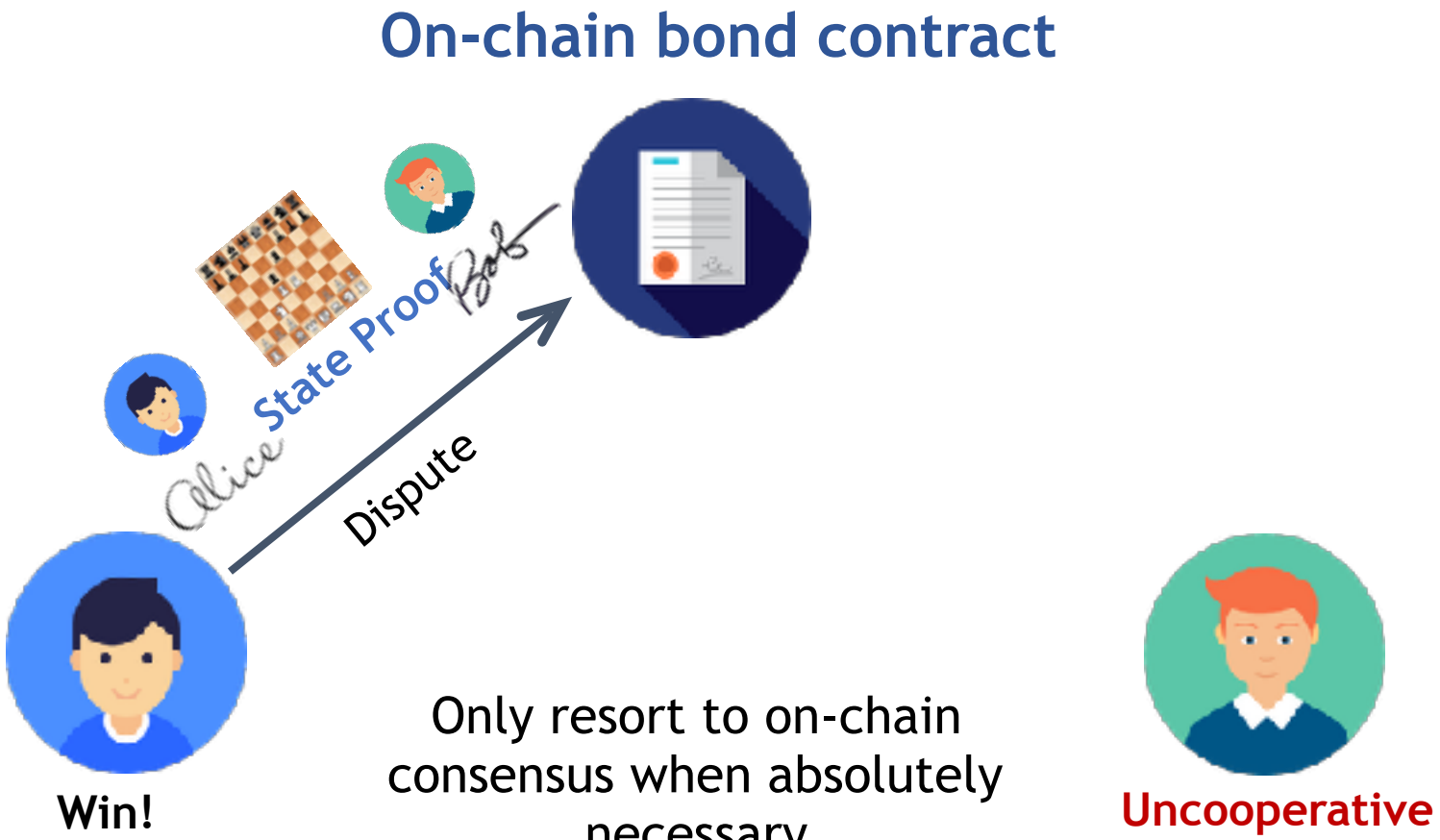
Win!

0.1 ETH

Zero on-chain operations when everyone is cooperative

Cooperative

Example: off-chain chess duel

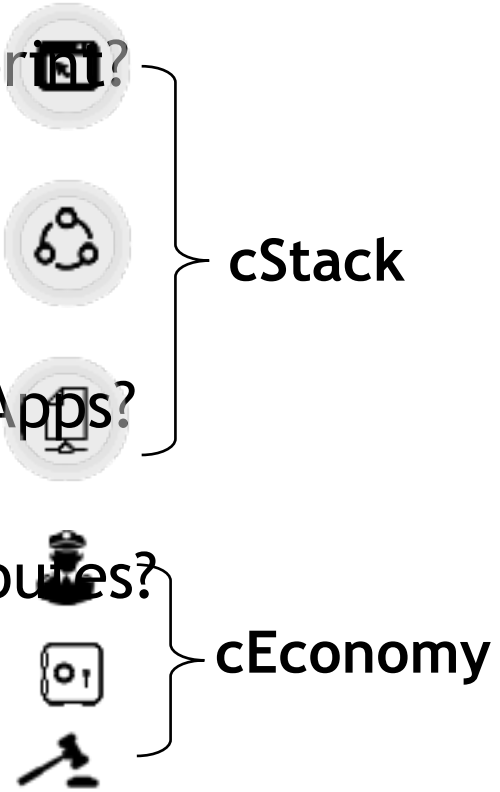


Only resort to on-chain consensus when absolutely necessary

Uncooperative

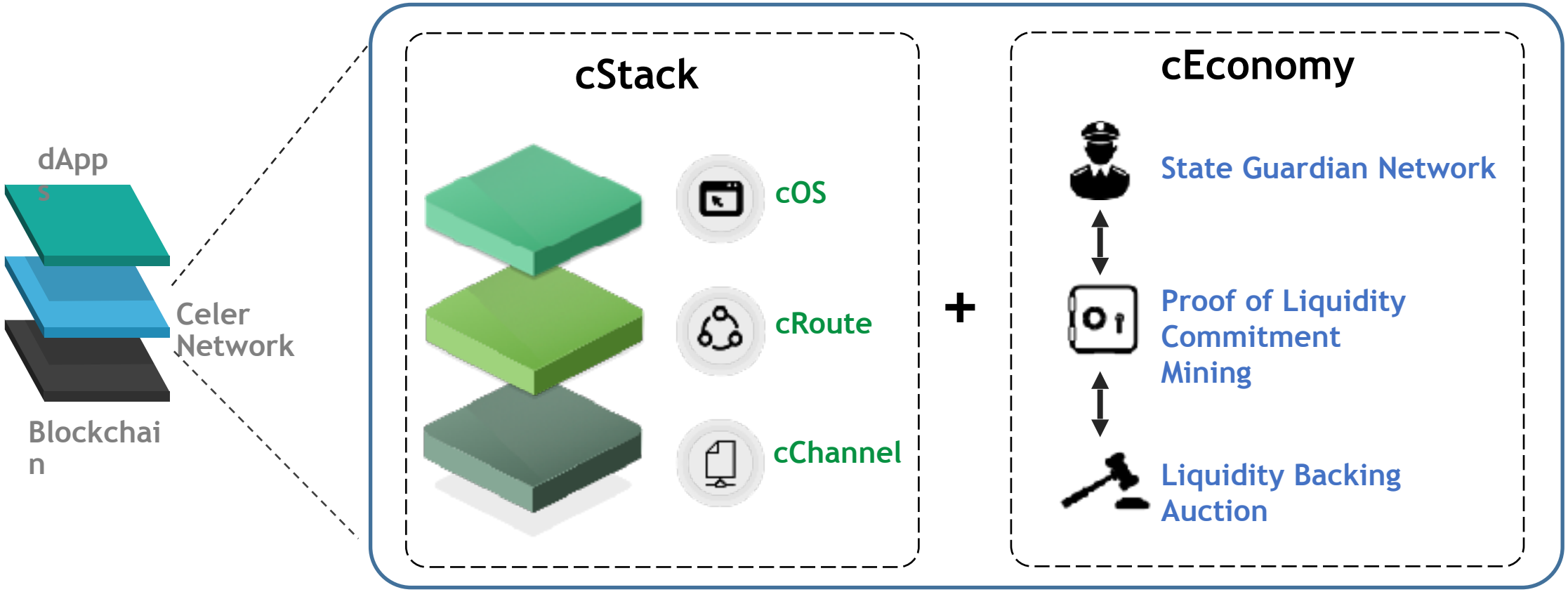
Bright future, challenges remain

- How to support generic operations with minimal on-chain footprint?
- How to route value transfers efficiently in off-chain networks?
- How to help developers to easily build and operate off-chain dApps?
- How to make off-chain states always available for on-chain disputes?
- How to obtain enough liquidity to run an off-chain service?

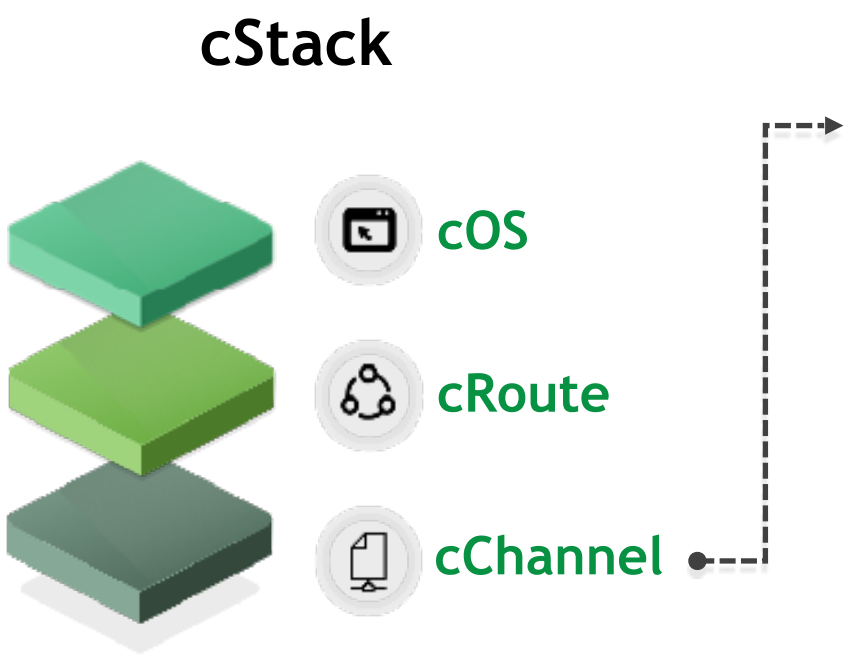


Celer Architecture

Celer Network is an Internet-scale, trust-free, and privacy-preserving platform where everyone can quickly build, operate, and use massively scalable decentralized applications.

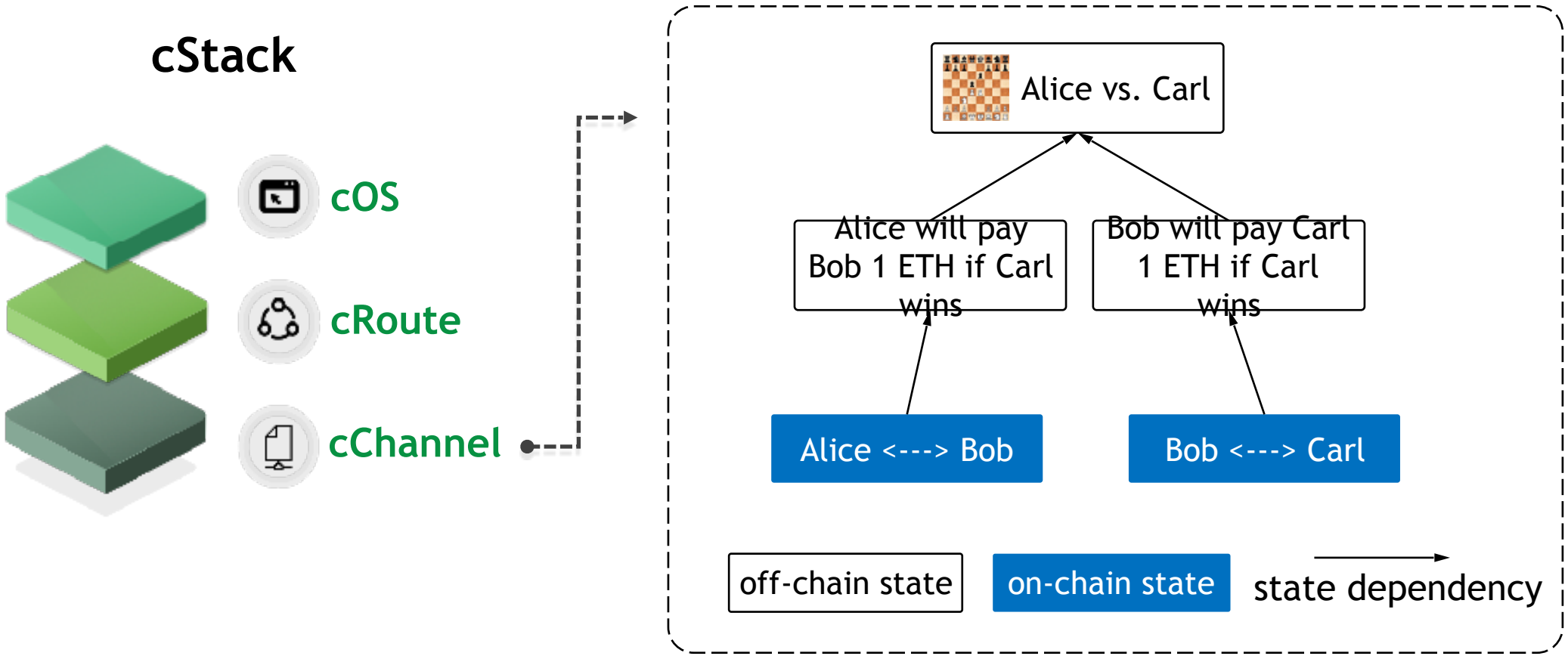


cChannel: Generalized dApp Support

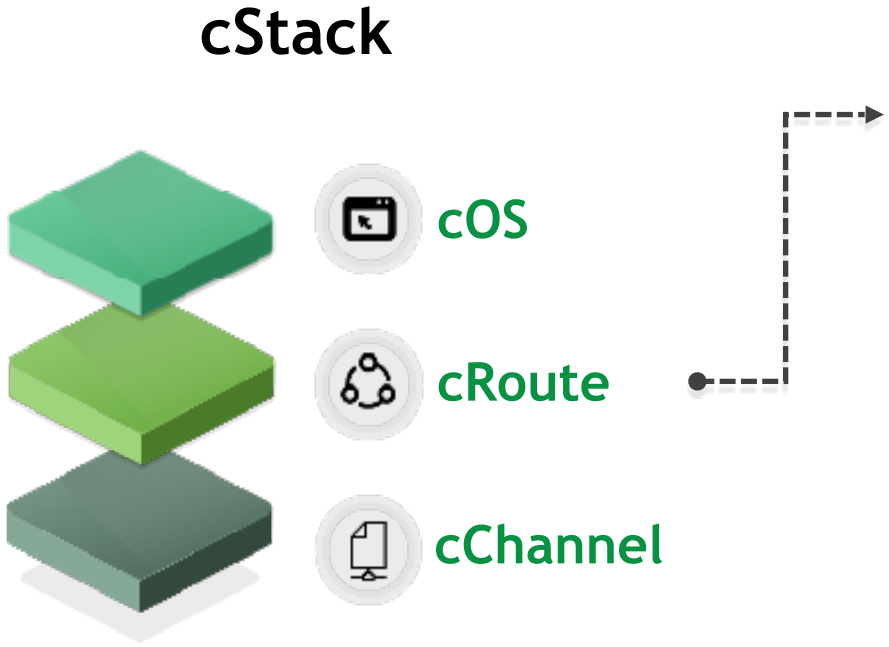


- Generic off-chain state transitions
  - conditional payment
  - multiparty gaming
  - second price auction
  - high frequency exchange
  - ...
- Pure off-chain contract
  - no on-chain deployment when everyone cooperates
- Multi-hop state relay
- Formal specification and verification

cChannel: Generalized dApp Support



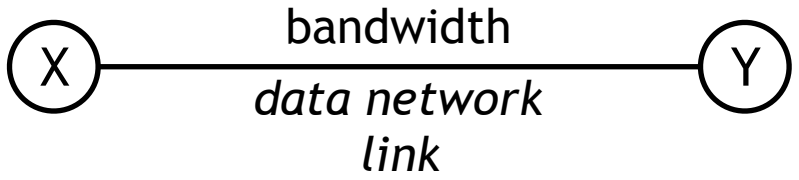
cRoute: Highly Efficient Payment Routing



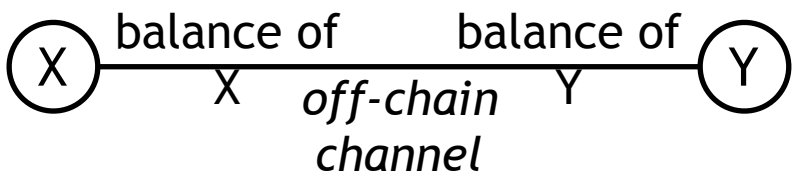
- Provably optimal throughput
- Transparent channel balancing
- Fully decentralized
- Failure resilience
- Privacy preserving

Why off-chain payment routing is challenging?

Off-chain payment networks are fundamentally different from data networks



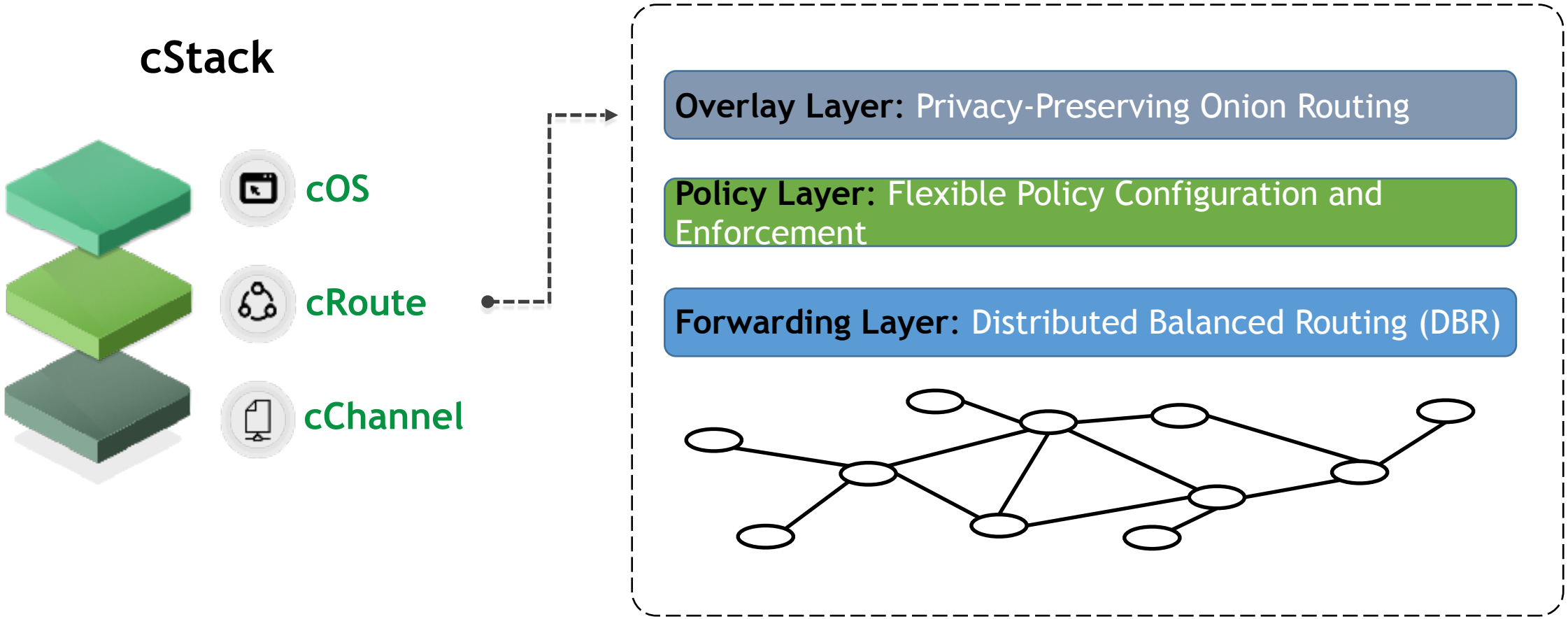
- Link state not affected by past transmissions
- Max rate is fixed (e.g., always 1Gpbs)
- **Network is relatively stable**



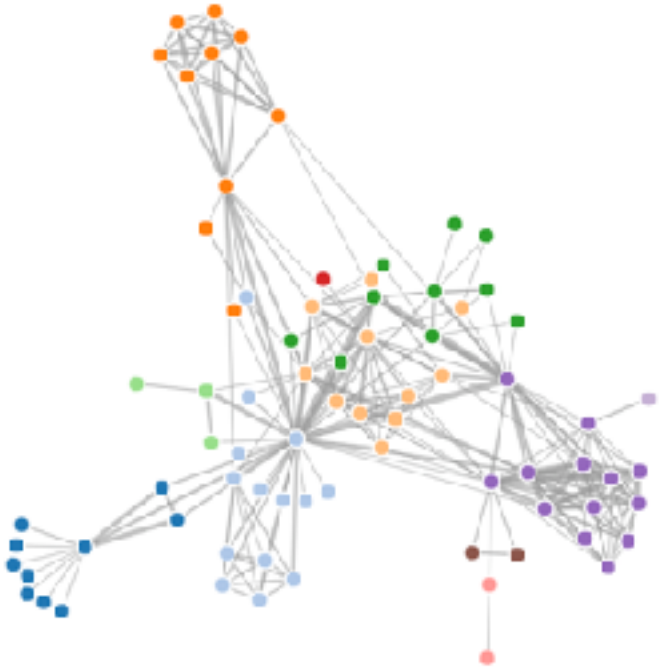
- Link state changed by *every* payment
- Max rate ranges from *zero* to *infinity*
- **Network is constantly changing**

Most distributed routing algorithms for data networks are hard to ***converge*** in off-chain networks

cRoute: Highly Efficient Payment Routing



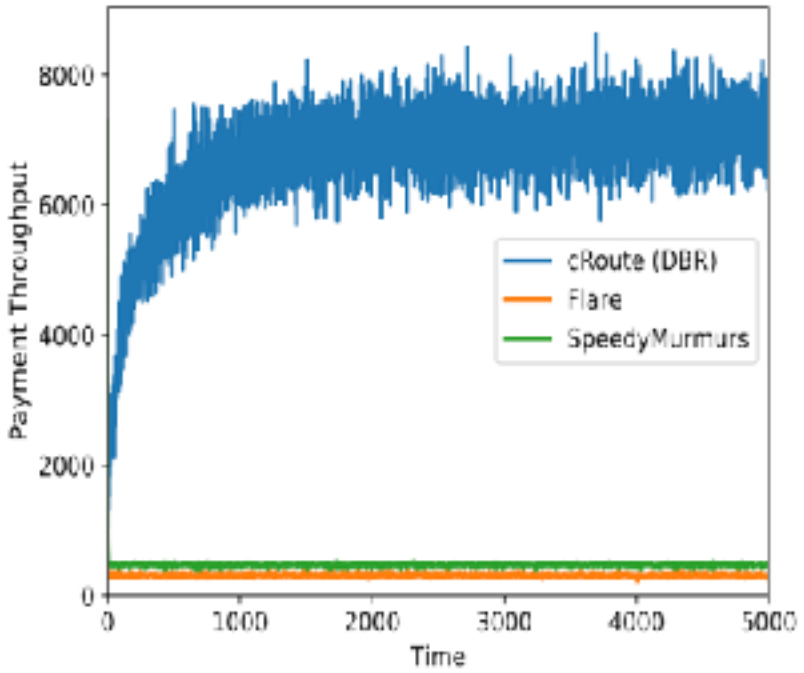
cRoute Simulation Results



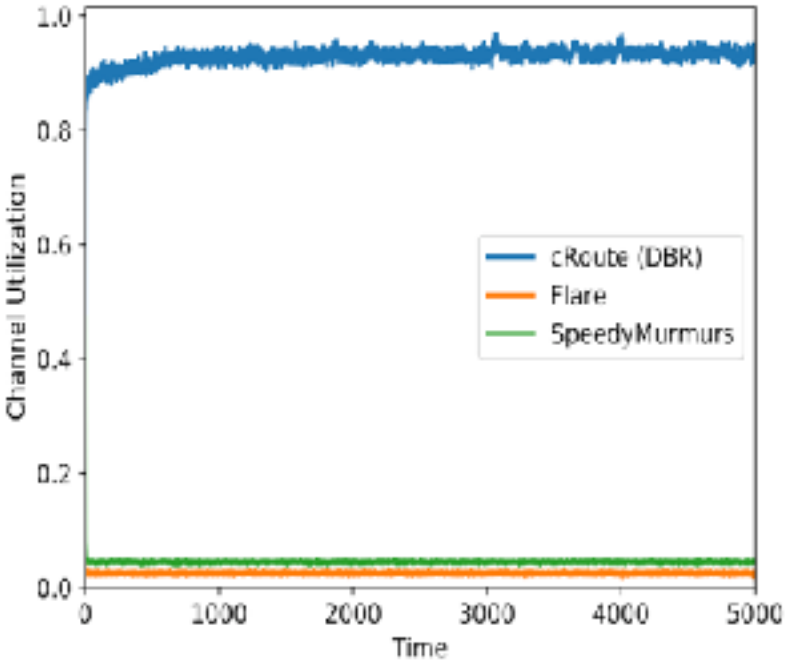
77 nodes

254 bi-directional payment channel

Poisson arrival with random src-dst

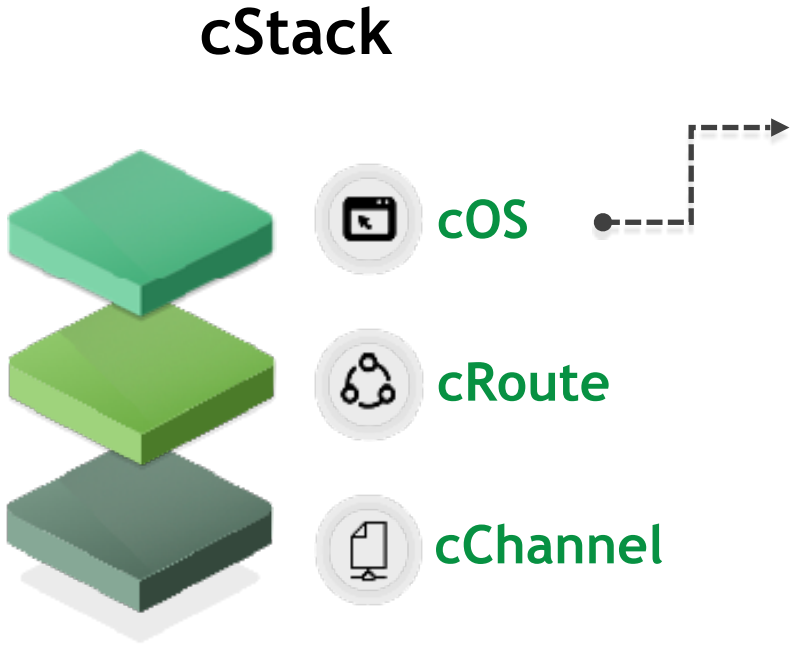


Payment throughput



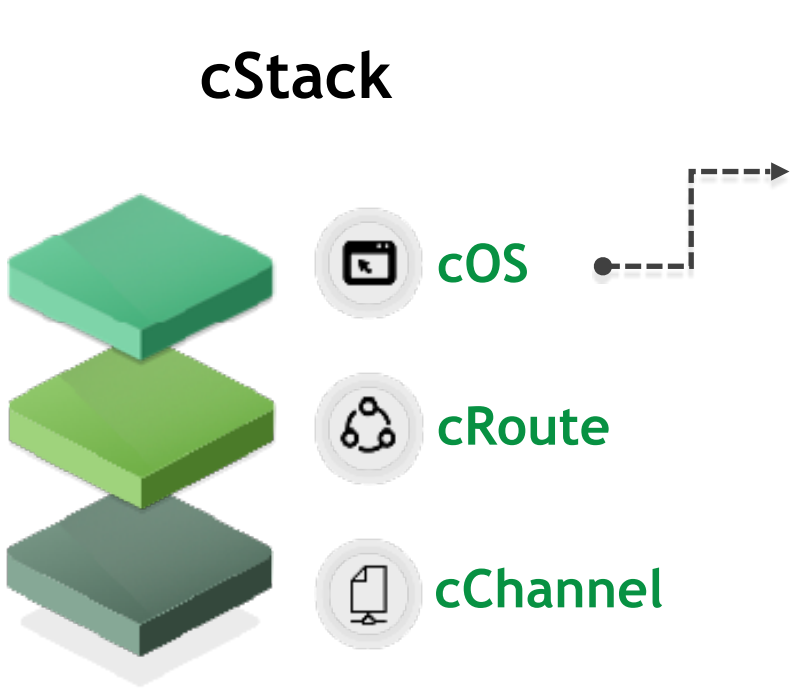
Channel utilization

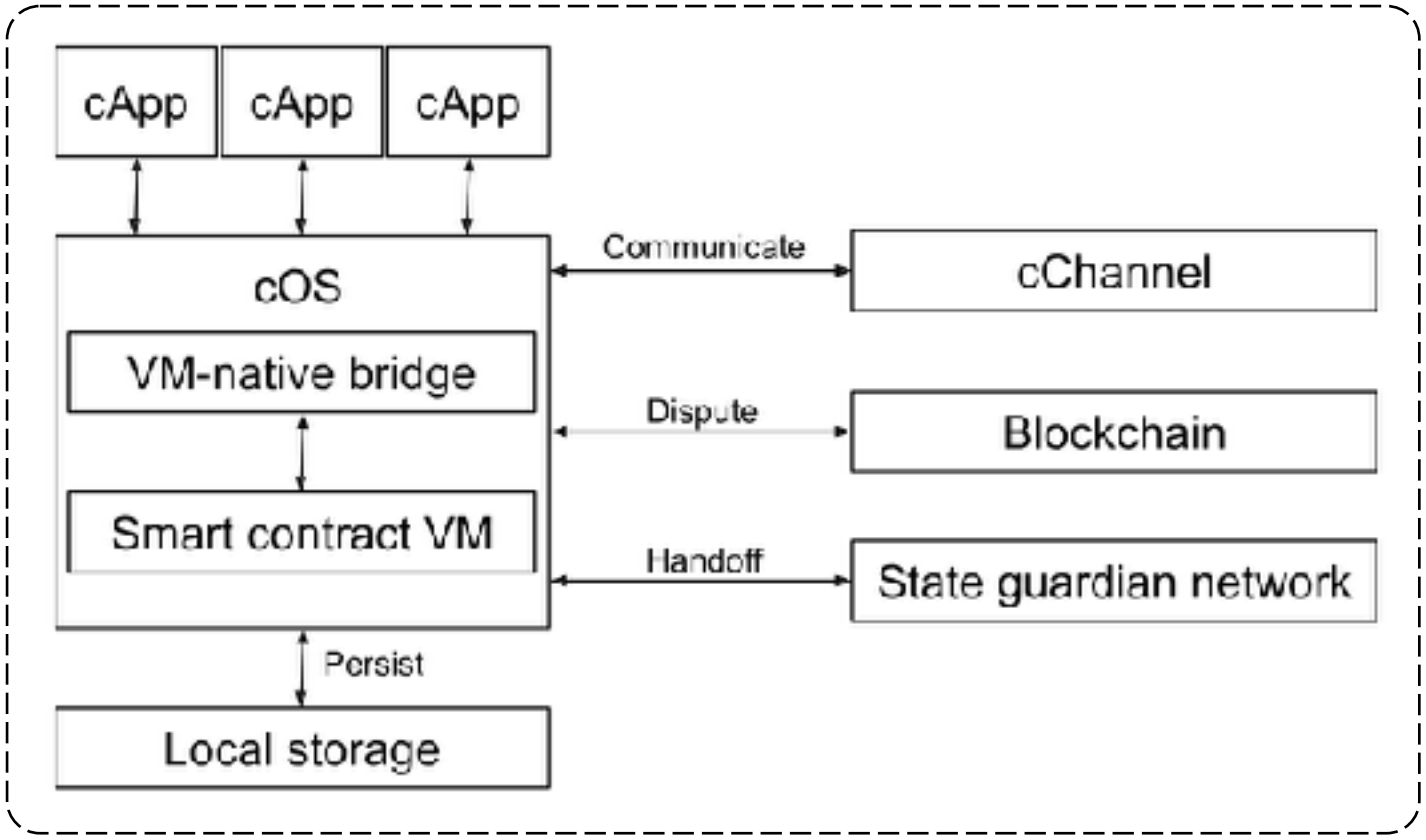
cOS: Development Framework and Runtime



- Provide common design patterns
- Enable “write once, run anywhere”
- Bridge on-chain and off-chain byte code
- Figure out state dependencies
- Track and dispute off-chain states
- Support concurrent off-chain dApps

cOS: Development Framework and Runtime





Scalability does not come for free!

Off-chain scaling is introducing new tradeoffs

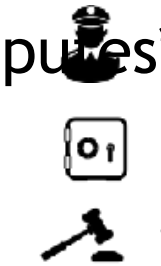
Bright future, challenges remain

Celer Network

- How to support generic operations with minimal on-chain footprint?
- How to route value transfers efficiently in off-chain networks?
- How to help developers to easily build and operate off-chain dApps?

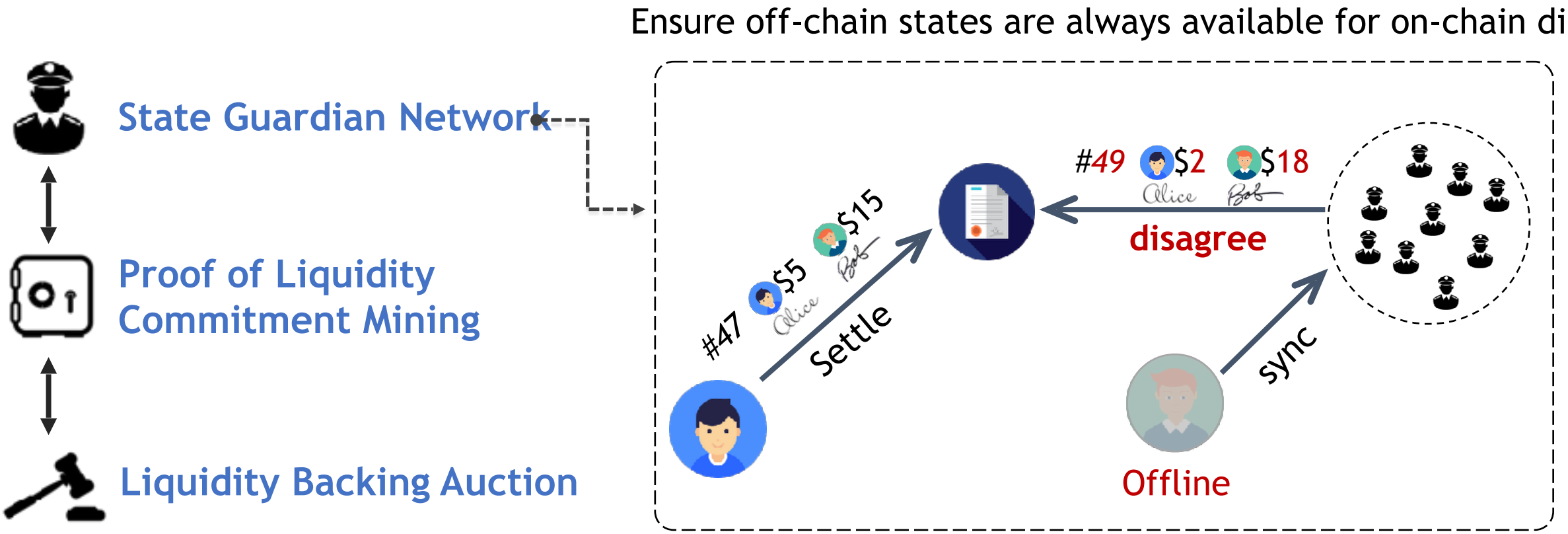
cStack

- How to make off-chain states always available for on-chain disputes?
- How to obtain enough liquidity to run an off-chain service?



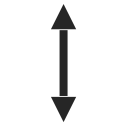
cEconomy

cEconomy completes the off-chain ecosystem

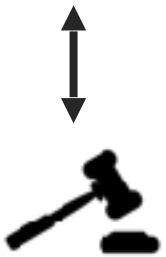


cEconomy completes the off-chain ecosystem

State Guardian Network

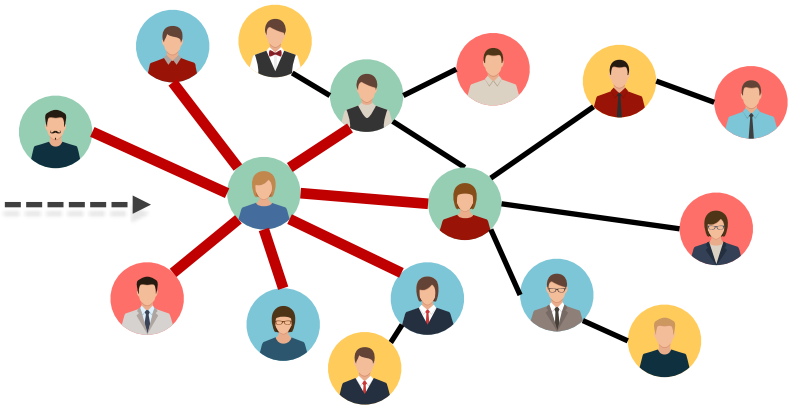


Proof of Liquidity
Commitment Mining



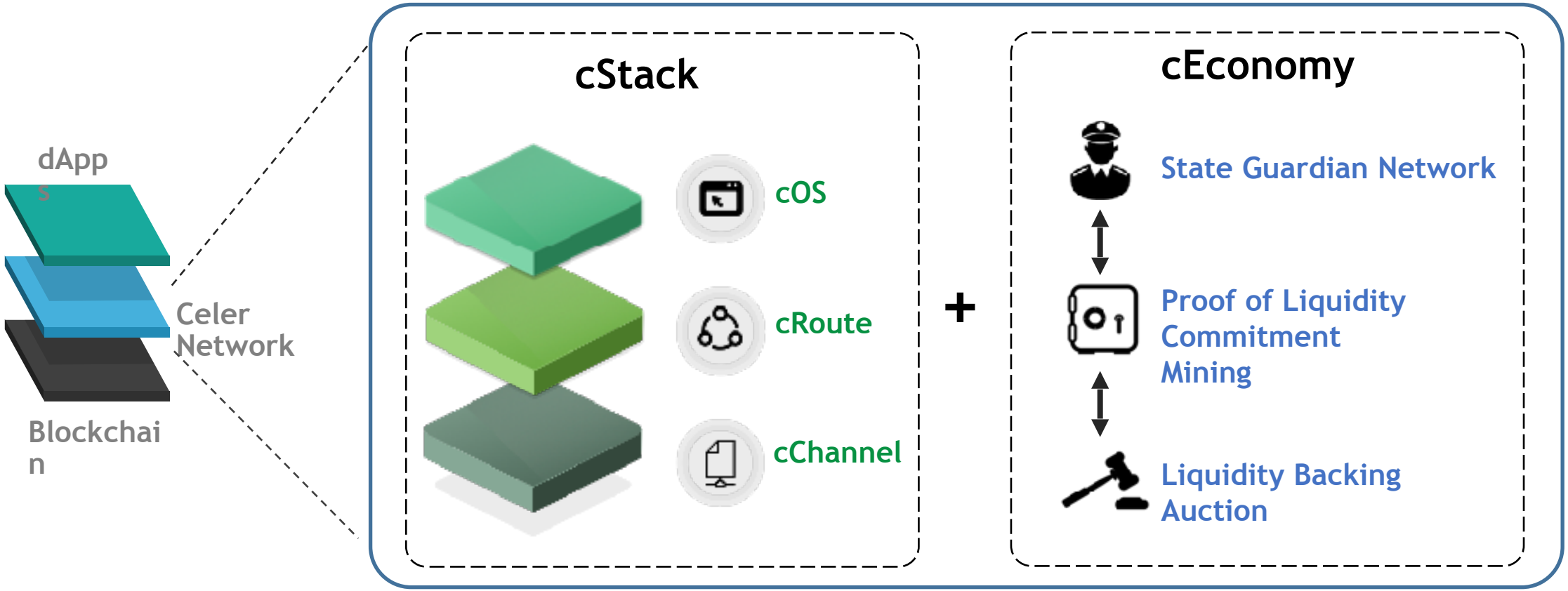
Liquidity Backing Auction

Incentivize an abundant and stable liquidity pool



- Off-chain service providers crowdsource liquidity
- Reward committed liquidity backers
- Ensure 100% safety for end users

Build and operate Internet-scale dApps on Celer Network



Thanks!

www.celer.network