

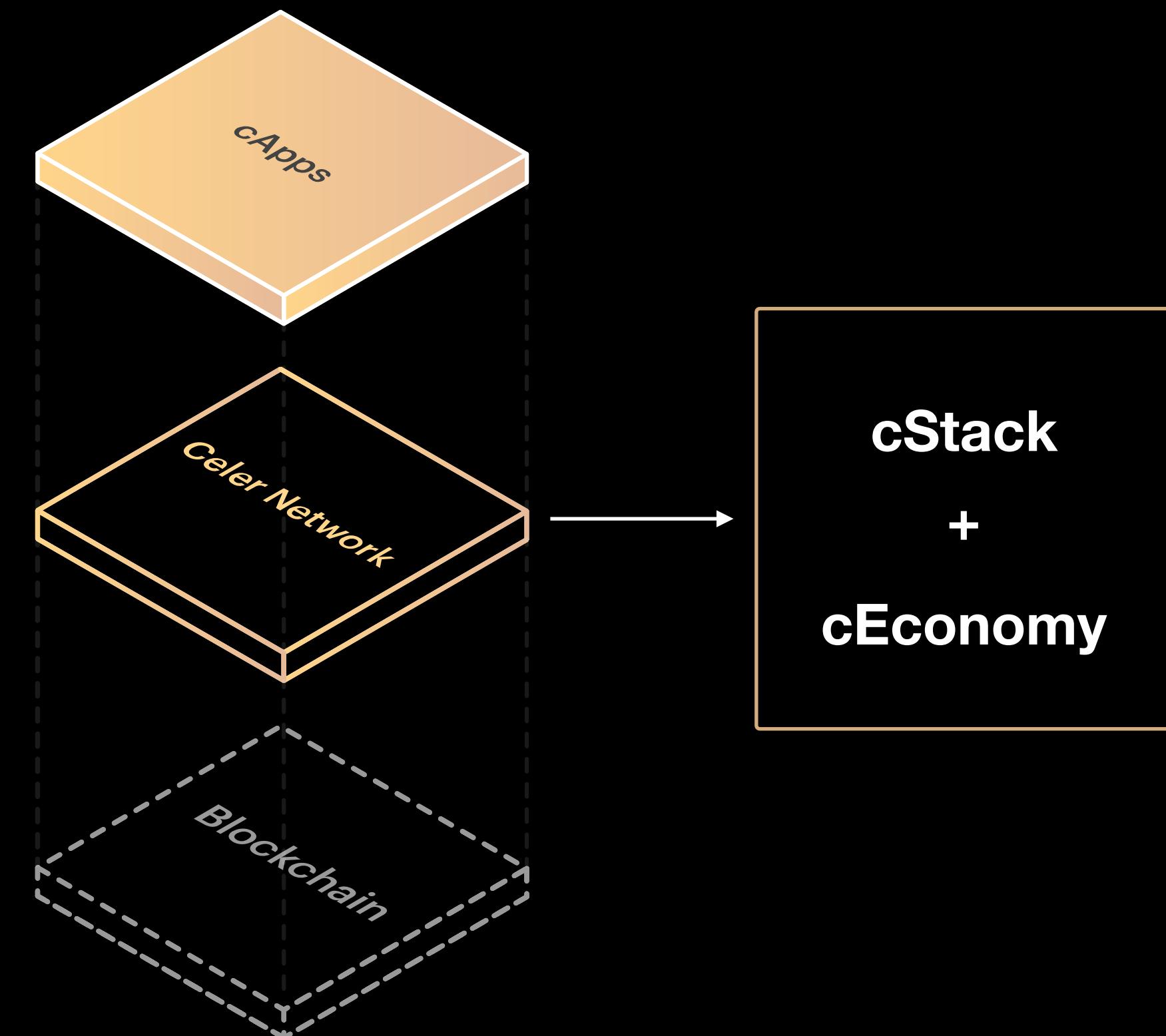


Scale dApps with Celer Network

Junda Liu

Celer Network

Coherent Off-chain Scaling Platform

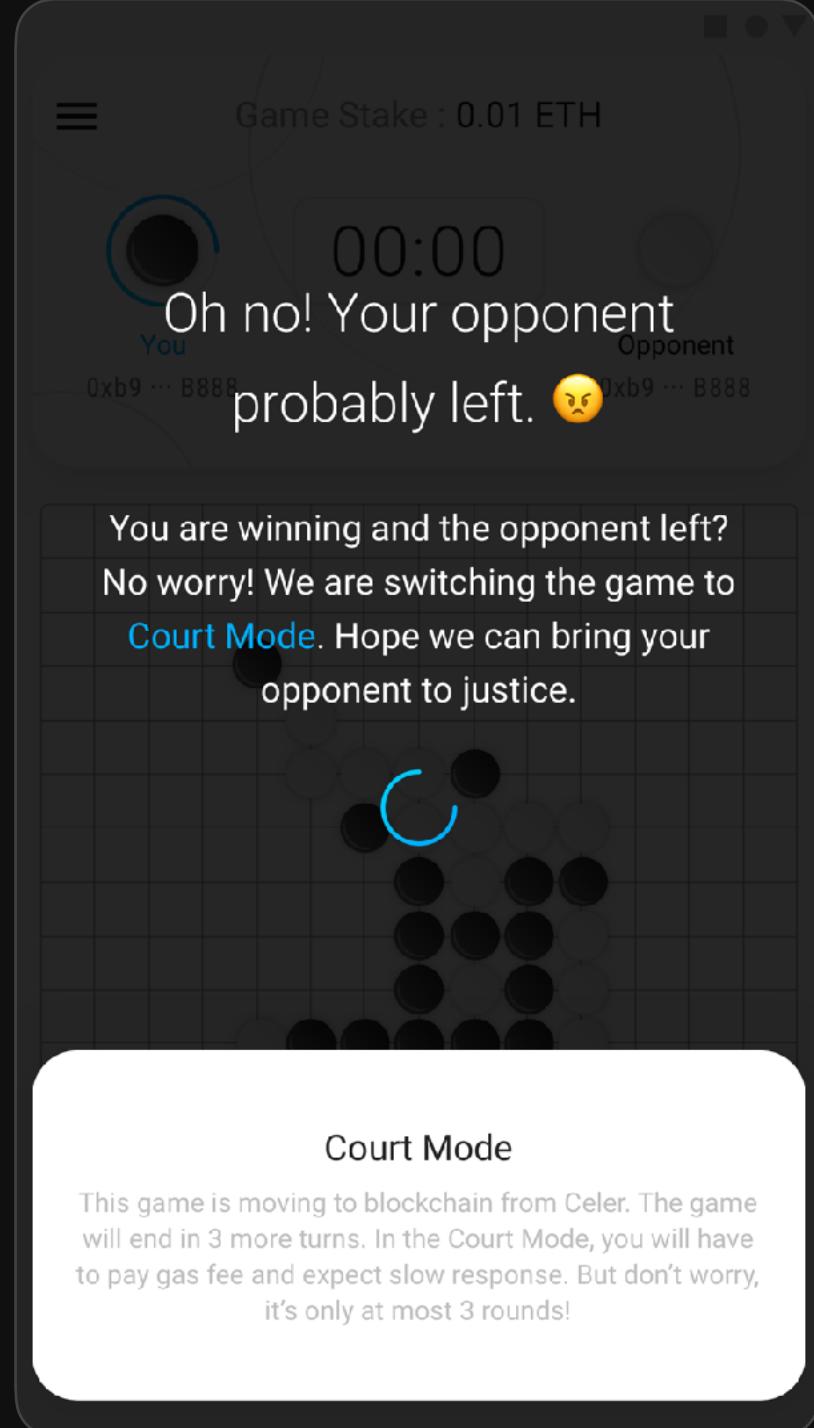
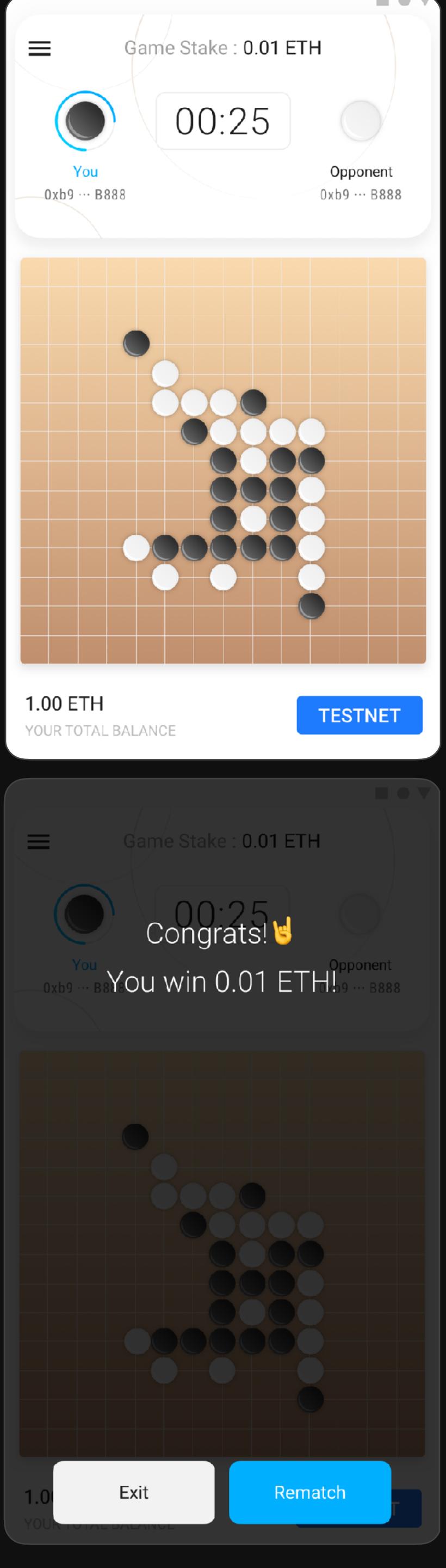
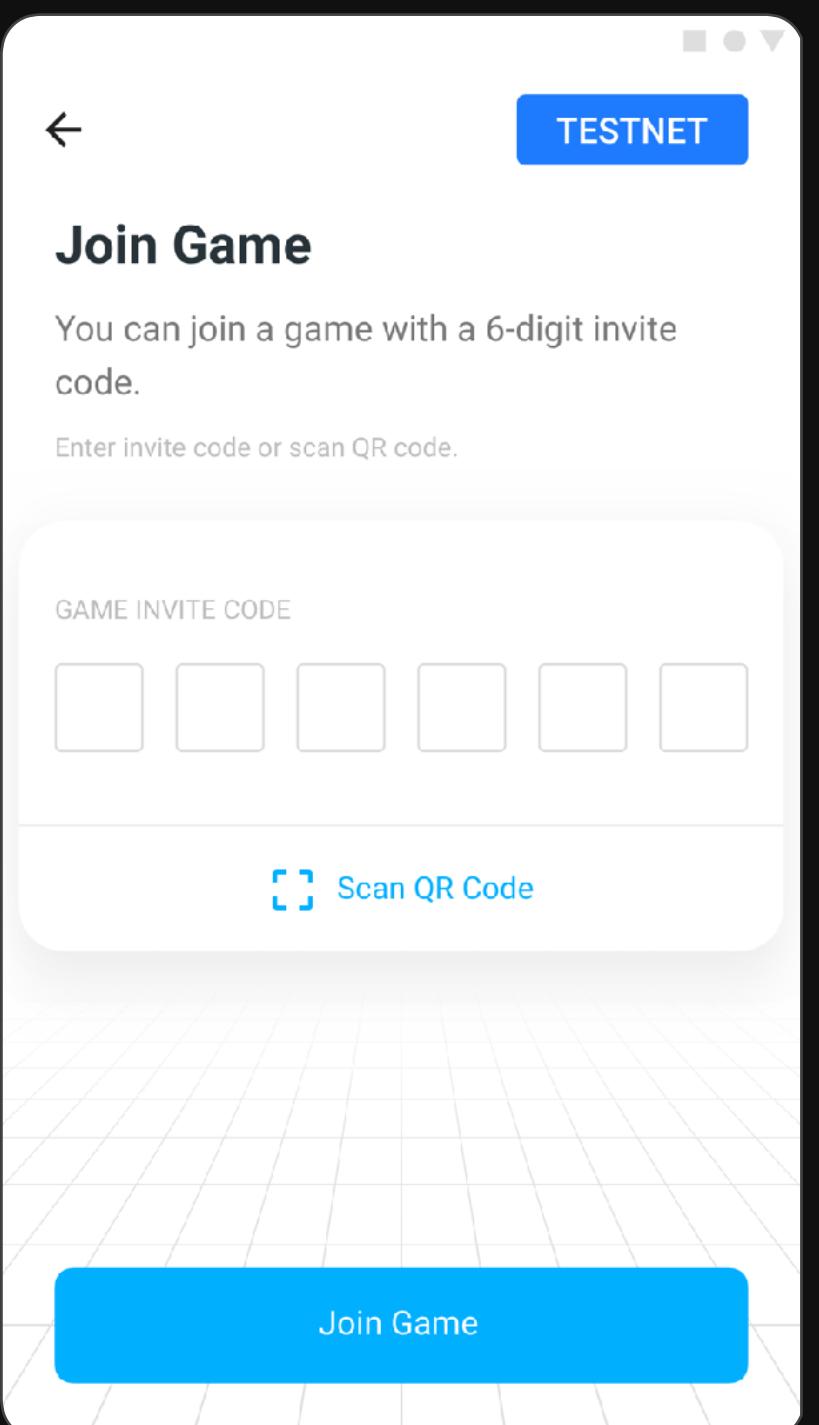




cWallet



get.celer.app





get.celer.app

Celer Network ▾ Go to docs Ⓜ Mo

Celer Network

Spark plan

3 apps | cPay celer.app.data celer.pay +

Users in last 30 minutes 12

Analytics

Daily active users 342 +90%

Day 1 retention 45.8% +175%

...

Kames Vessel for Serendipity @KamesCG · Oct 22 State channels are absolutely amazing!

#GameChanger

Oh no! Your opponent probably left. 😞

The game is switching to Court Mode. Hope we can bring your opponent to justice.

0x42...e619 0x80...8002

You will be matched with a random player.

GAME STAKE 0.05 cETH

You have 0.5 cETH available

0.01 0.05 0.1

Court Mode

1 2 9 1

Kames Vessel for Serendipity @KamesCG · Oct 22 Absolutely fantastic job @CelerNetwork team! I will be singing your praises and using this tech where ever possible.

cWallet also serves as a reference developers can easily transform th... checkout our SDK and join cPilot early developer program!

50 Comments Share Edit Post Save Hide ...

Evan Van Ness @evan_van_ness Following

Playing Gomoku on @CelerNetwork's app is lots of fun (running on state channels). If you haven't tried it, you should

Not-a-black-window Celer Testnet: 401 Users, 4 award-winning cAp... We are totally amazed by ETHSanFrancisco (hosted by ETHGlobal). If you missed this great event, we invite you to feel the vibrant Ethereum... medium.com

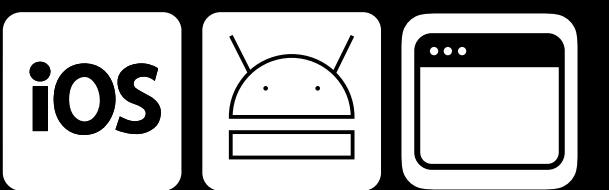
1:36 PM - 19 Oct 2018

5 Retweets 28 Likes

3 5 28

BUIDL Today

Easy-to-use SDK, IOS & Android
Golang also available



Slick (Gas Option)
All-in-one service smoothes ETH gas price volatility via transaction queuing, and creates Ethereum's first gas option

STORY **UPDATES**

Slick **ETH San Francisco**

CREATED BY

Concept development, research into options & derivative marketplaces.
Service design, design of user journeys/flows, final UI mock-up.
Managed the team's development schedule.

Joe Mallonee Hands-on, versatile Product Designer.

Tina Zhen Serial tech entrepreneur; blockchain researcher, Princeton Economics

AirPrediction
An off-chain lightning fast prediction market

PREDICTION MARKET

Inspiration
Prediction market is a marketplace for people to bet on the outcomes of future events. In an active market, many people give opinions about specific events and generate an overall prediction based on the knowledge of all of them. However, traditionally, the process is like a black-box. You don't know if your dealer is trustworthy. The key factor to obtain precise predictions is to make the marketplace transparent and open to everyone. This is why prediction market is a very good use case for blockchain applications. However, blockchain-based prediction market can lead to very slow order matching, payment and also has a problem to get the result information. Thanks to some awesome projects, we can solve these problems and build a fast and secure off-chain prediction market.

What it does
We built a off-chain prediction market with the help of Celer Network and Chainlink. In our off-chain prediction market, users can submit an event for prediction by providing an oracle contract address. Then all users can place orders to bet on the result with some ethers. These orders will form an online order book. Once YES and NO orders (think about bids and asks in a stock market) match, the corresponding issuers will get noticed and establish a bidirectional conditional payment through Celer Network. Based on the result given by the oracle contract, Celer Network will resolve the

All Classes

Balance
BalanceCondition
CApp
CAppCallback
Chain
Group
GroupContract
GroupItem
GroupRepo
Mobile
RoundRobin

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY NESTED FIELD CONSTR METHOD DETAIL FIELD CONSTR METHOD

newwork.callable
Class CApp
java.lang.Object
newwork.callable.CApp
All implemented interfaces
go.ReqObjIntf, go.SreqProxy

public class CApp
extends java.lang.Object
implements go.SreqProxy

Constructor Summary

Constructors

Constructor and Description

CApp()

Method Summary

All Methods **Instance Methods** **Concrete Methods**

Modifier and Type Method and Description

boolean equals(java.lang.Object o)
CAppCallback getCallback()
java.lang.String getContractAddress()
java.lang.String getContractName()

Micro Subscription
Micro subscription solution for cloud service based on conditional micro payment and onchain oracle

STORY **UPDATES**

ETH San Francisco

CREATED BY

Xuefeng Zhu

Inspiration
I was inspired by SLA example in chainlink repo. I realized that currently, it is not very convenient and efficient to resolve SLA dispute. With blockchain, I can programmably resolve the dispute. Another issue I identified in current cloud service is that most of cloud service is subscribed by month, but it is not fair price model for user with low utilization. I think it will be very helpful to enable micro payment for services they actually use. There are multiple micro payment solutions out there, but I found out only Celer is able to support conditional payment by querying on-chain information. With conditional payment, I can use service condition to trigger payment.

What it does
I built a new payment and dispute solution for cloud service. Users are able to make micro payment for the service they actually use by minutes. It also uses on-chain oracle to monitor the service status. If the service is always on within a minute, user will pay the service provider. If the service goes down, user does not need to pay the service provider, and the service provider will pay the user instead for its lost based on SLA.

Building Community



Online Course



Cultivating dev community

A screenshot of the Celer Network blog feed on Medium. The feed shows several posts: "Testnet and SDK Launch", "Lighting up the Centaur! Celer Network SDK and Testnet Launch at SFBW", "Celer Network MVP: The Most Advanced State Channel Full-Stack Solution", "Celer Network 10th Weekly Project Progress Report (9/17/18 - 9/21/18)", and "Celer Network 11th Weekly Project Progress Report (9/24/18 - 9/28/18)". Each post includes a small thumbnail image and some descriptive text.

Educating market

State Channel Researchers Call #1

Participants:

- CELER
- L4
- UCL
- PERUN
- SparkChain
- Magmo
- PARAGATE
- TECHNISCHE UNIVERSITÄT DARMSTADT

Agenda:

- Perun 10-min lighting intro
- Magmo 10-min lightning intro
- Celer 10-min lightning intro
- Discussion of interested topics

Note:

- We forgot to record call #0...
- The first few calls will be about intro to every participating project and free-style Q&A

Defining standard

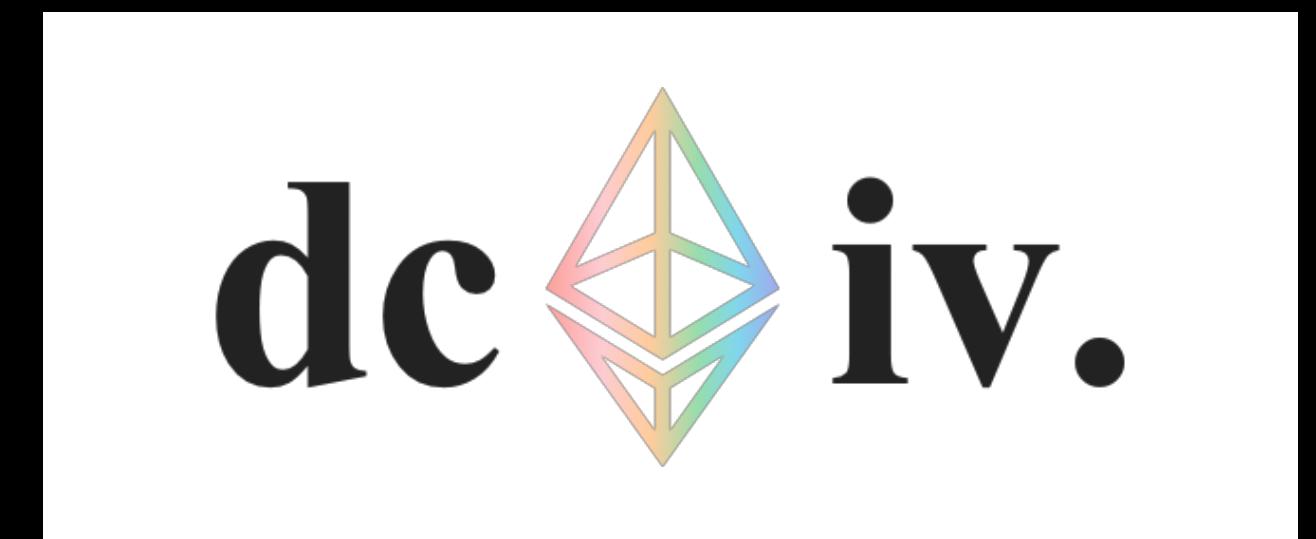
Building Community



ETH San Francisco



CESC 2018

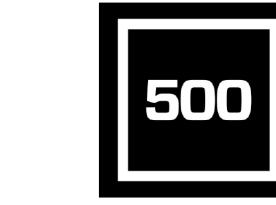


Devcon 4

Creating Ecosystem

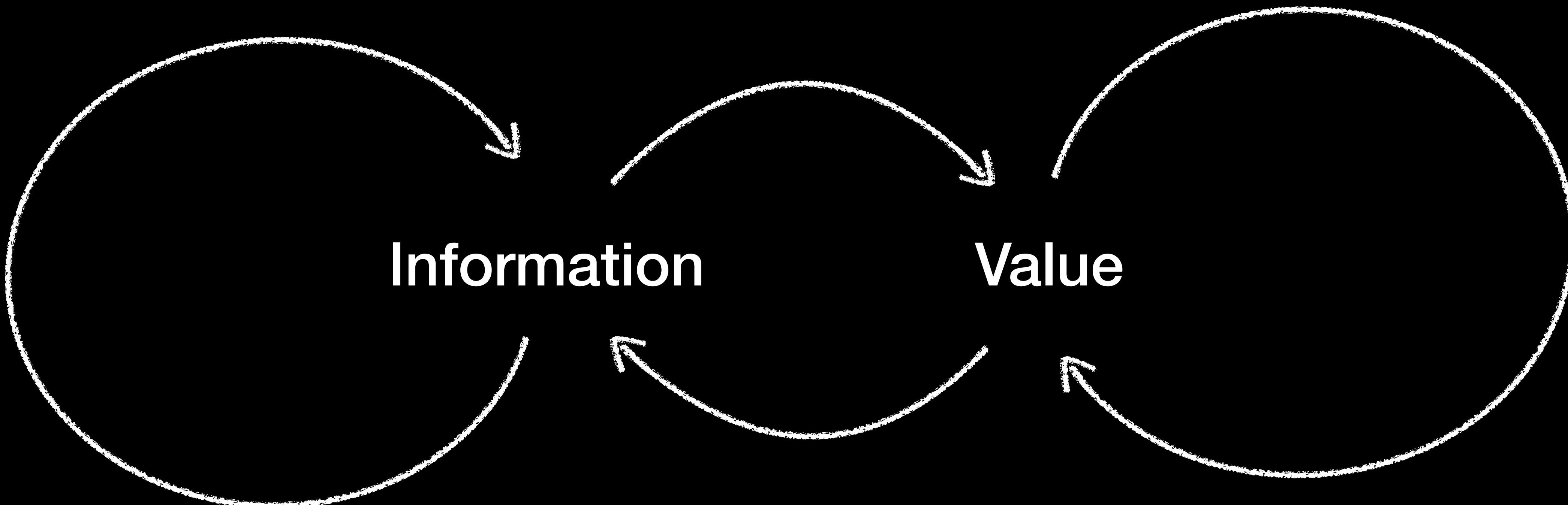


Investors

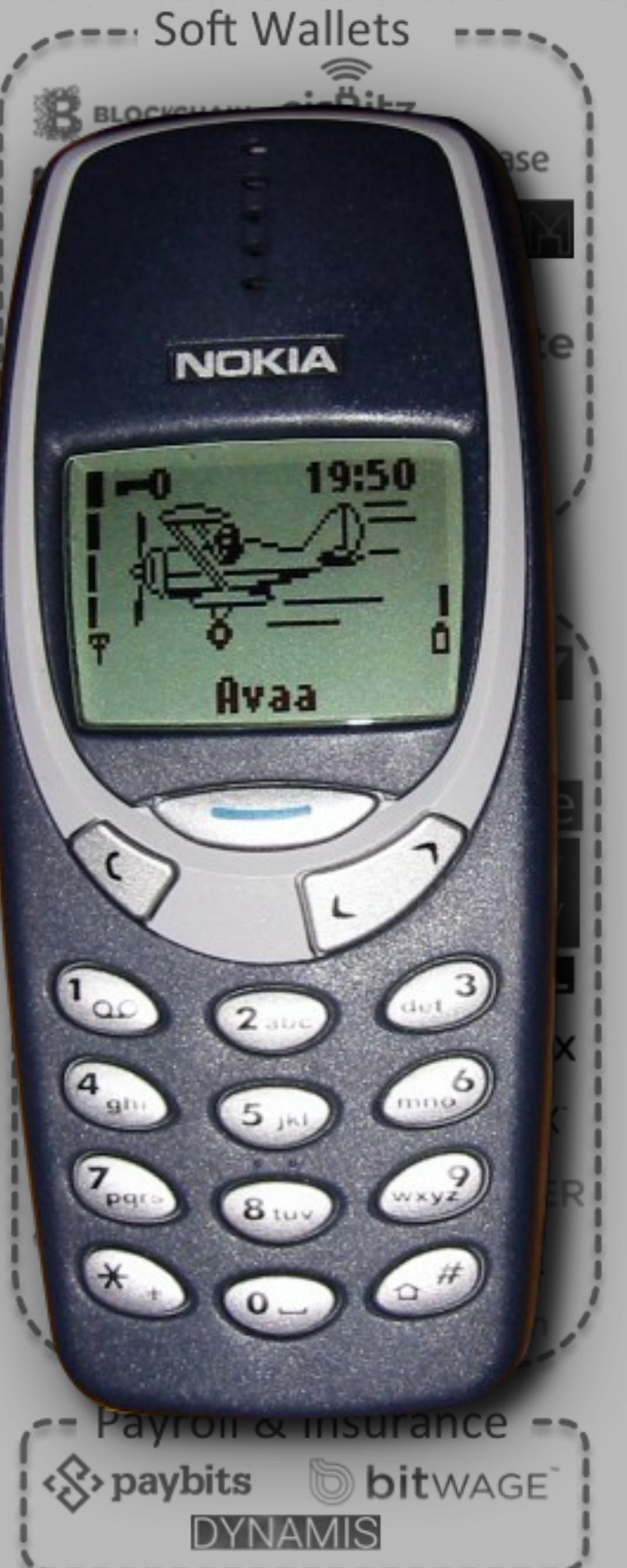


Economic Activity

Lagging behind due to trust barrier



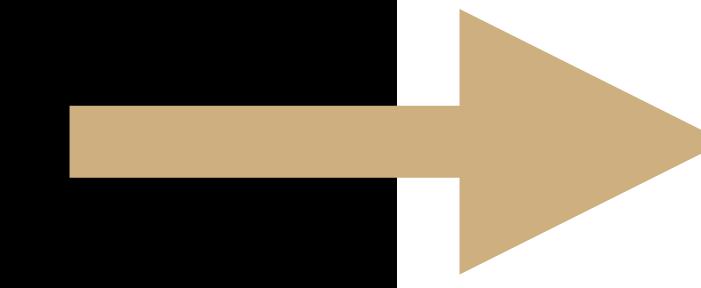
APPLICATIONS & SOLUTIONS



What is the end goal for
blockchain scalability?



**2M Emails,
65K Google Searches,
72K Youtube Video,
53TB Data
per second**



1000 KB/tx

**Goal: Match the Scalability of
Information**



**Mass Adoption:
530 Million TPS**

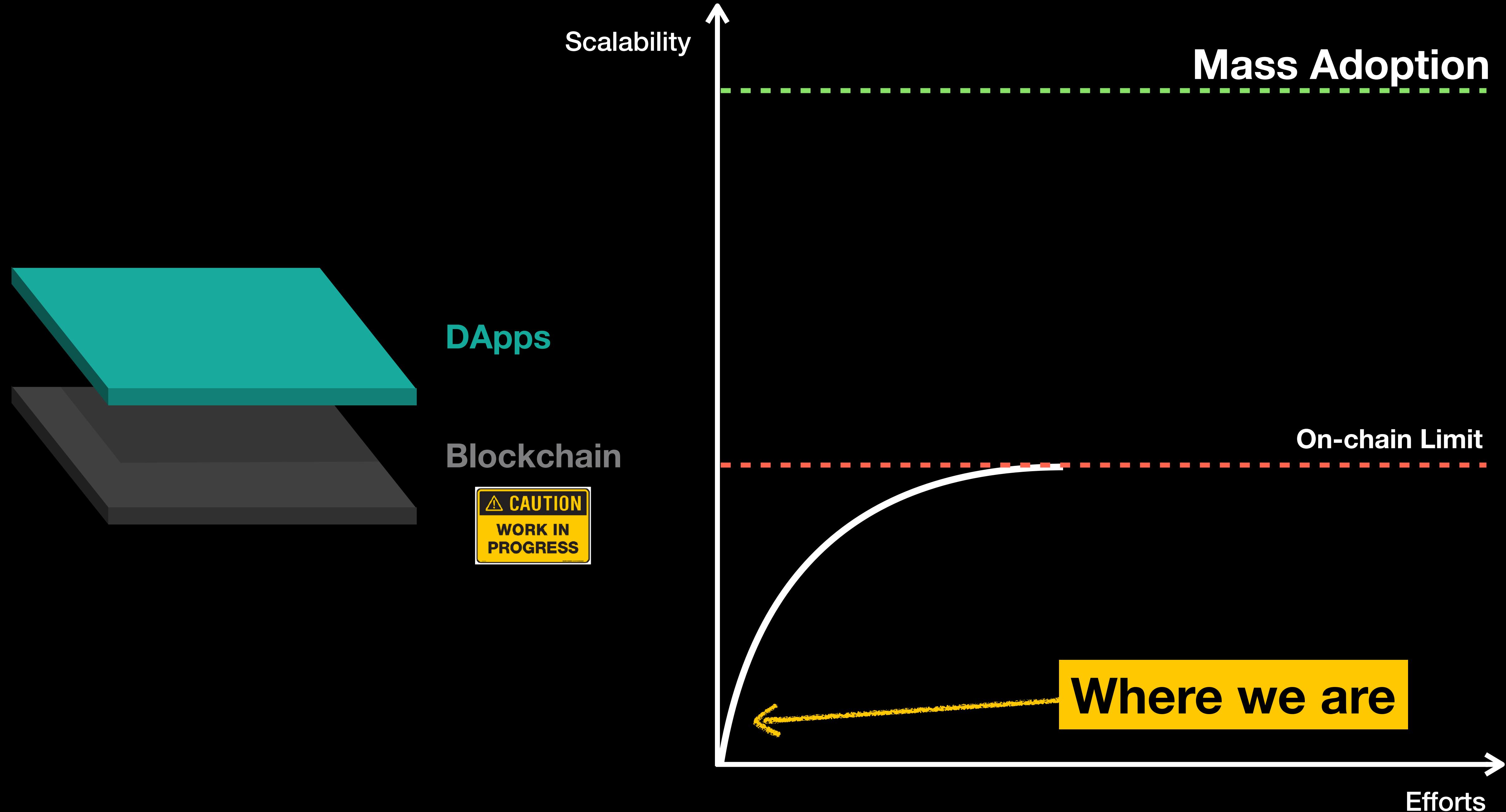
500M TPS ...

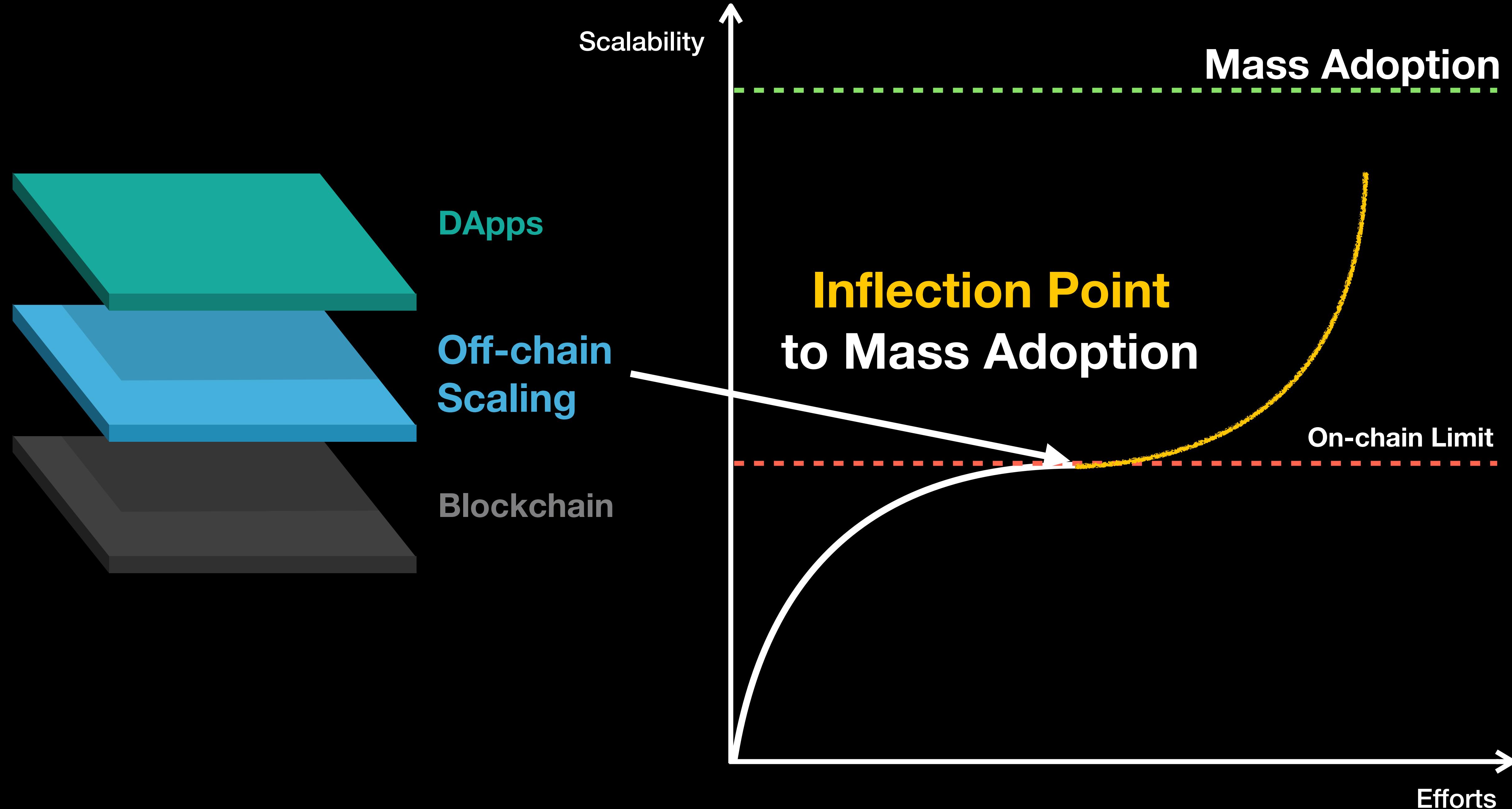


How could we ever get there?

TPS is not everything!

Latency is key for user adoption



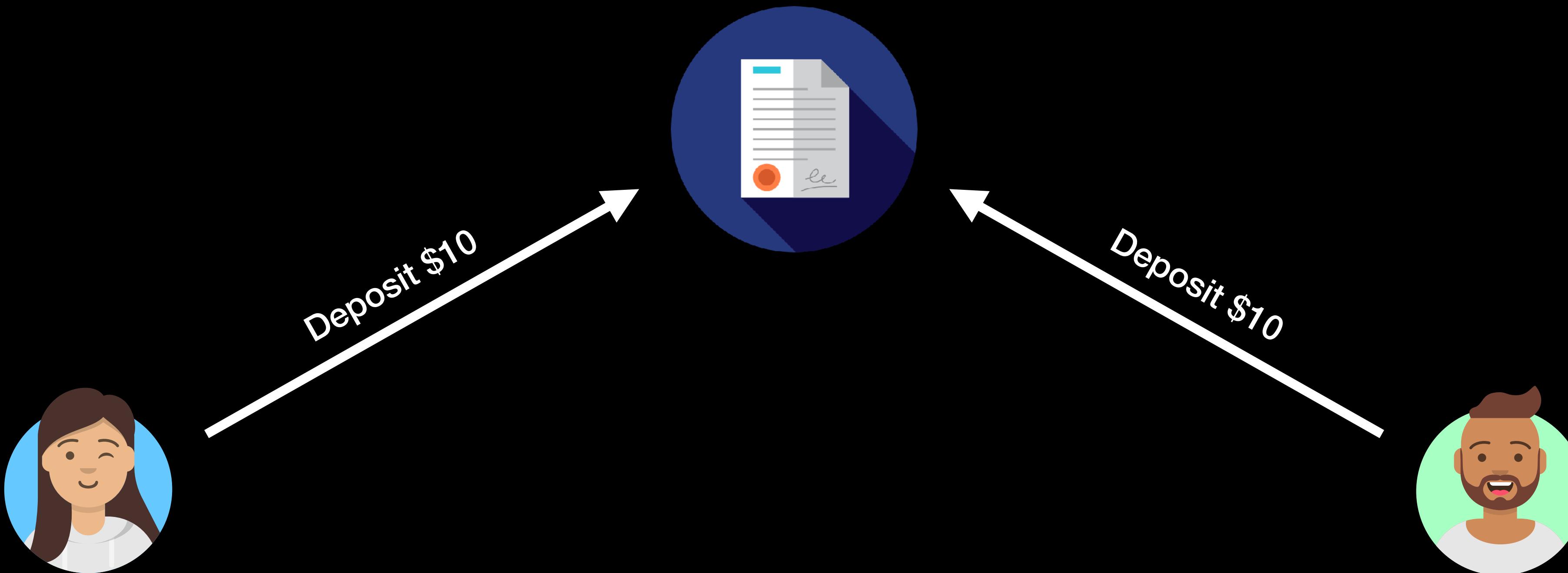


Off-chain Scaling Refresh

Off-chain Payment Channel

On-chain deposits

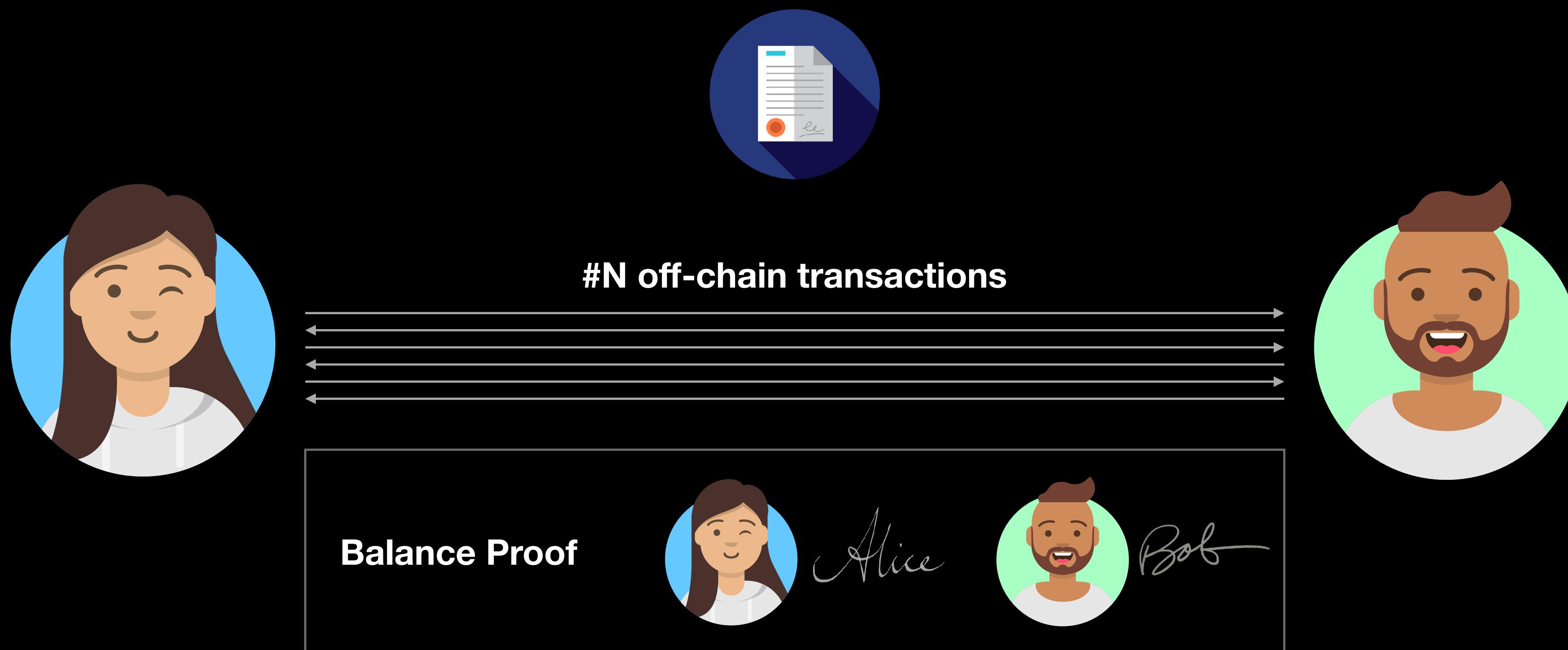
On-chain bond contract



Off-chain Payment Channel

Off-chain transactions

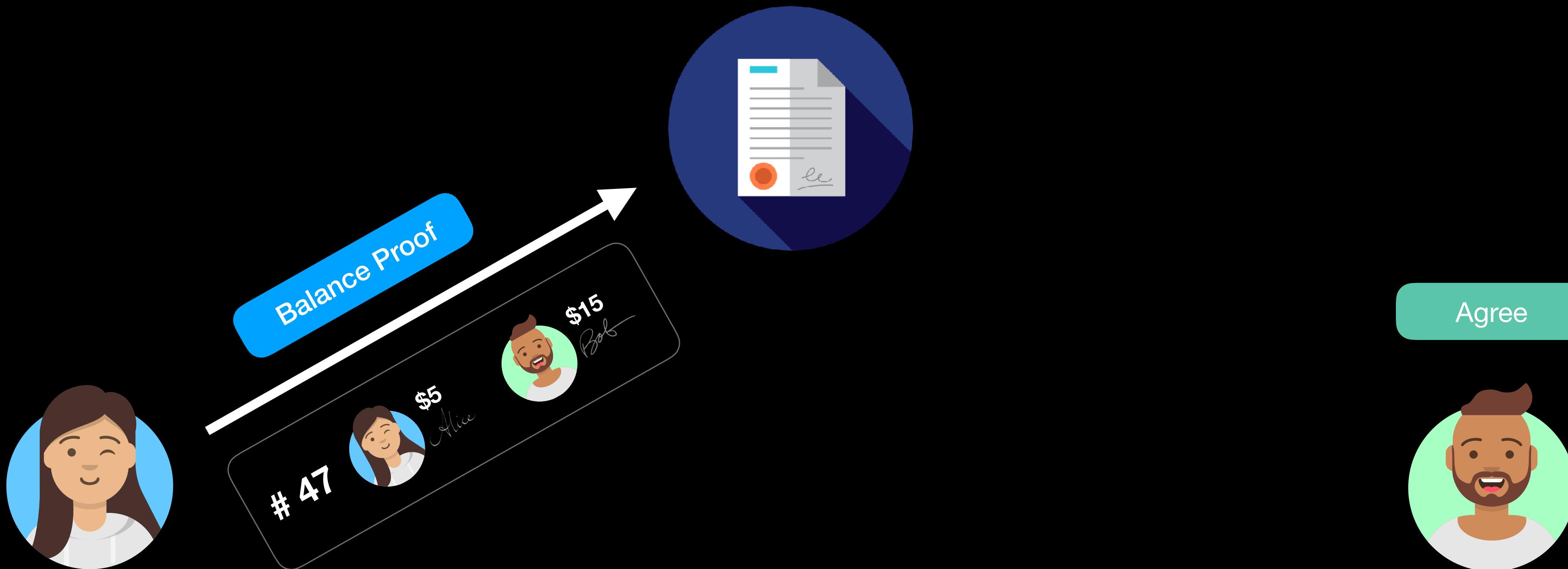
On-chain bond contract



Off-chain Payment Channel

On-chain settlement

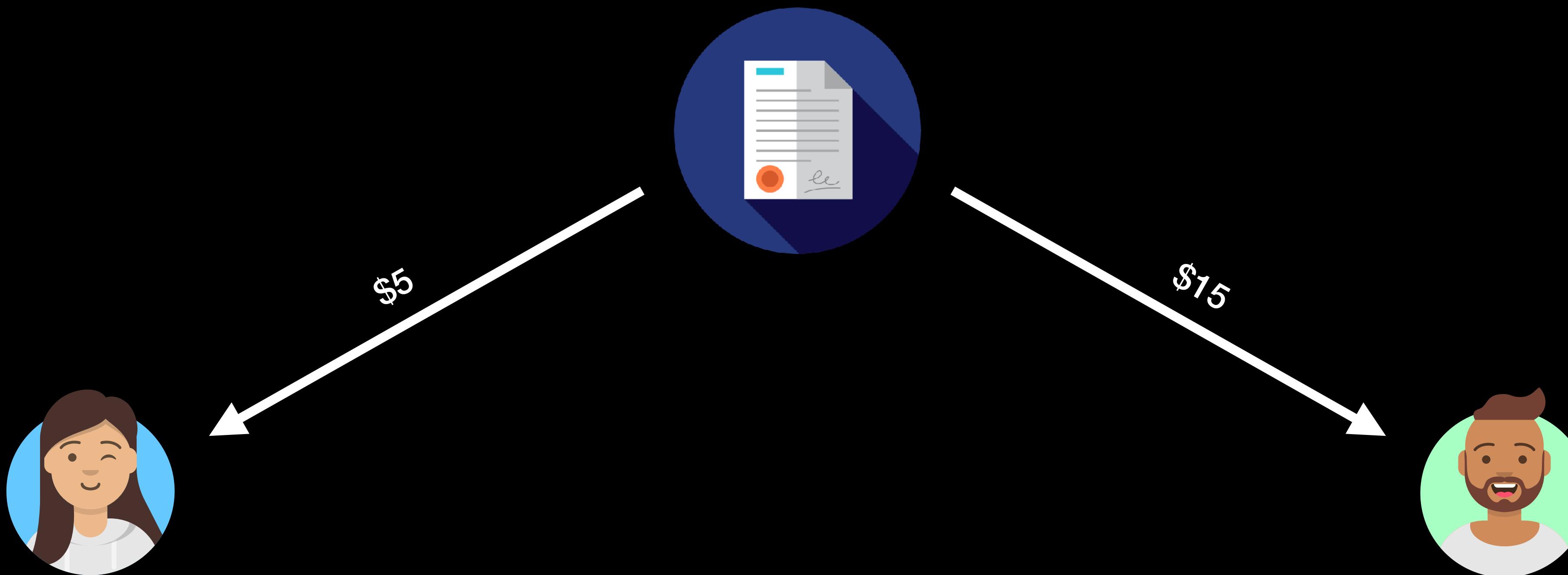
On-chain bond contract



Off-chain Payment Channel

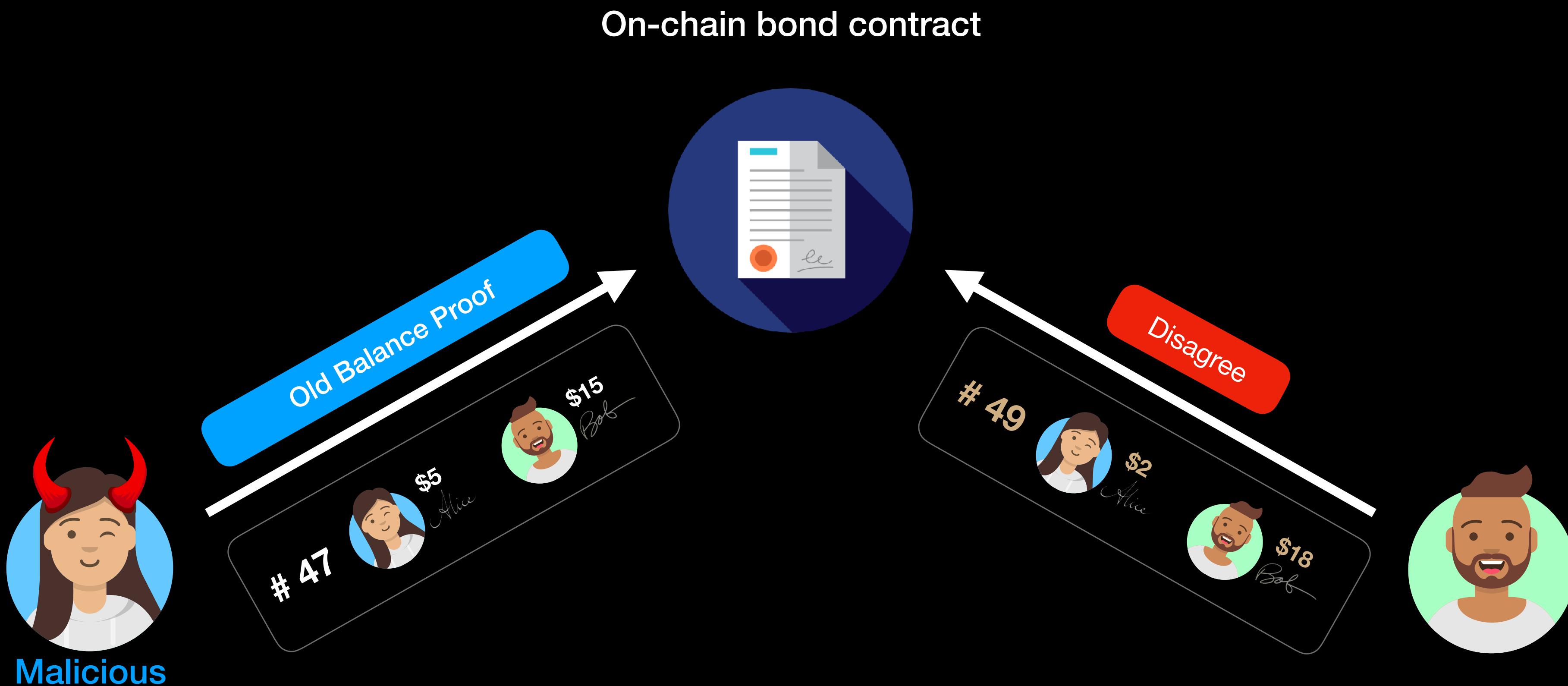
Channel Settled

On-chain bond contract



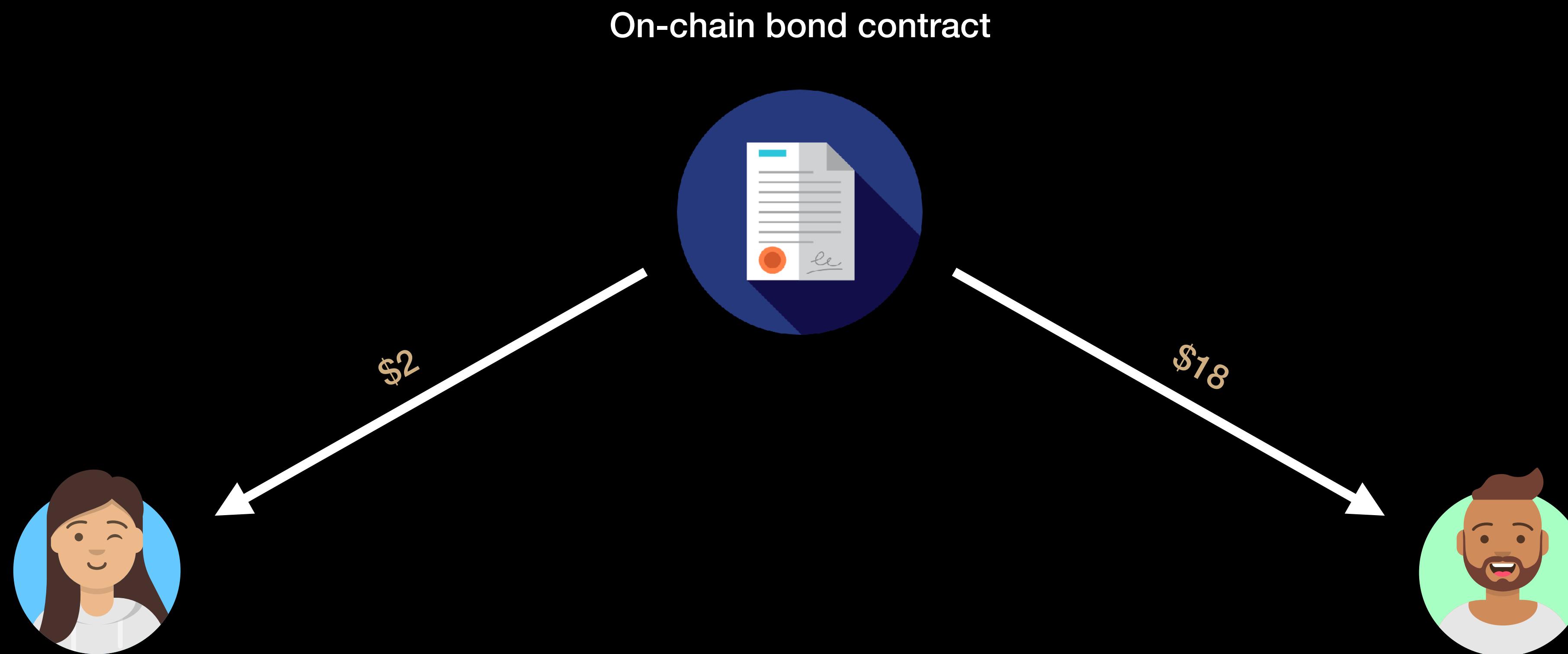
Off-chain Payment Channel

On-chain settlement - Malicious Case

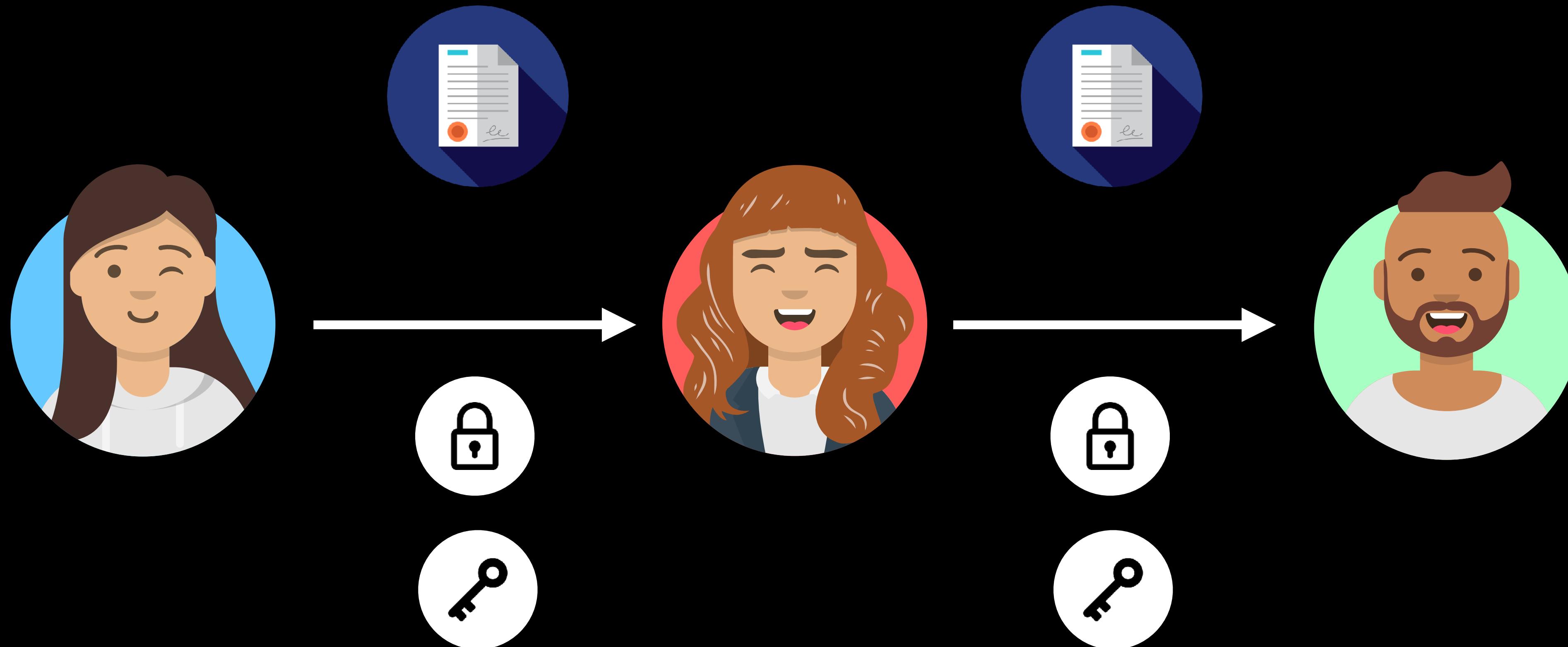


Off-chain Payment Channel

Channel Settled - Malicious Case

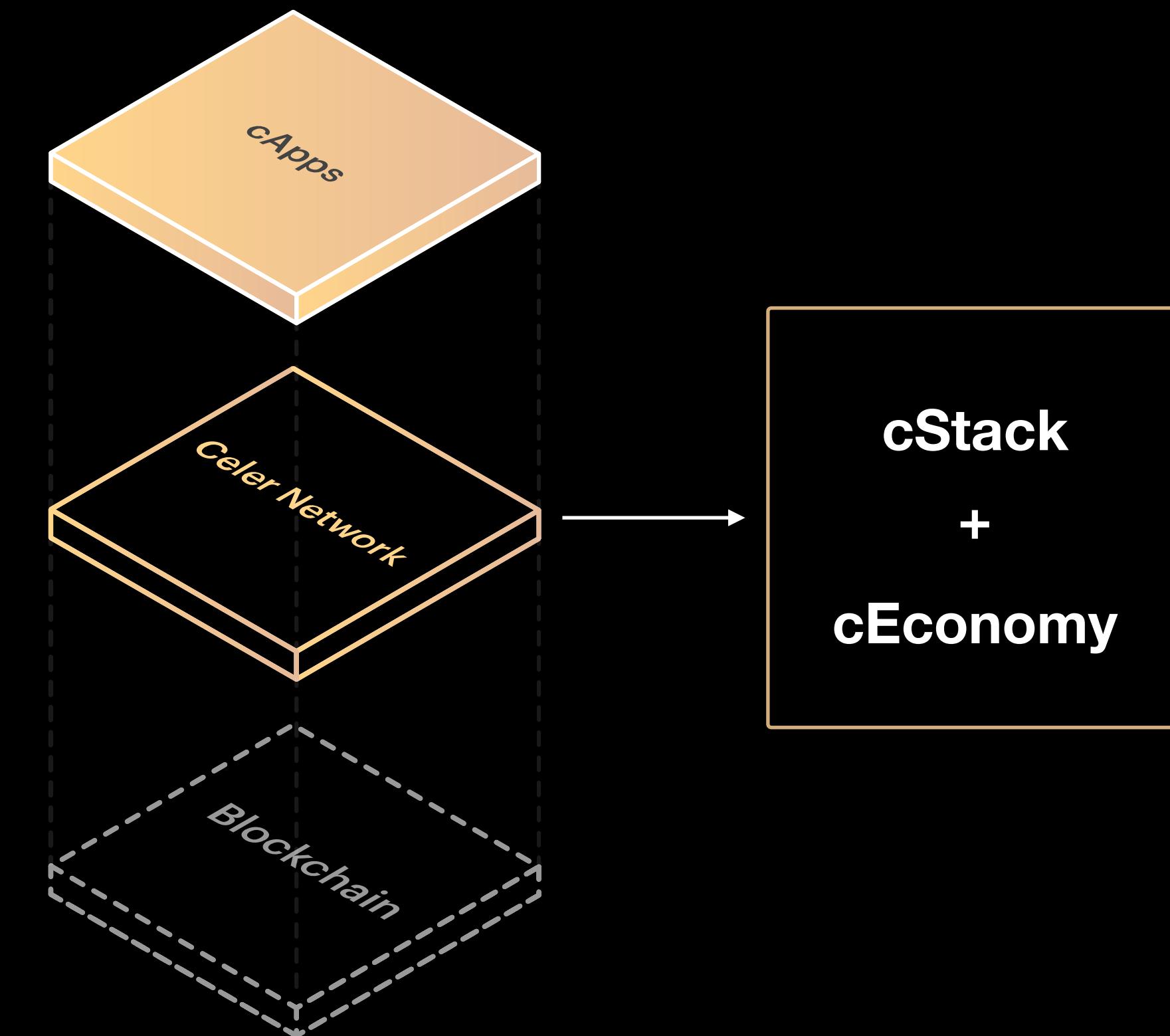


Off-chain Payment Channel



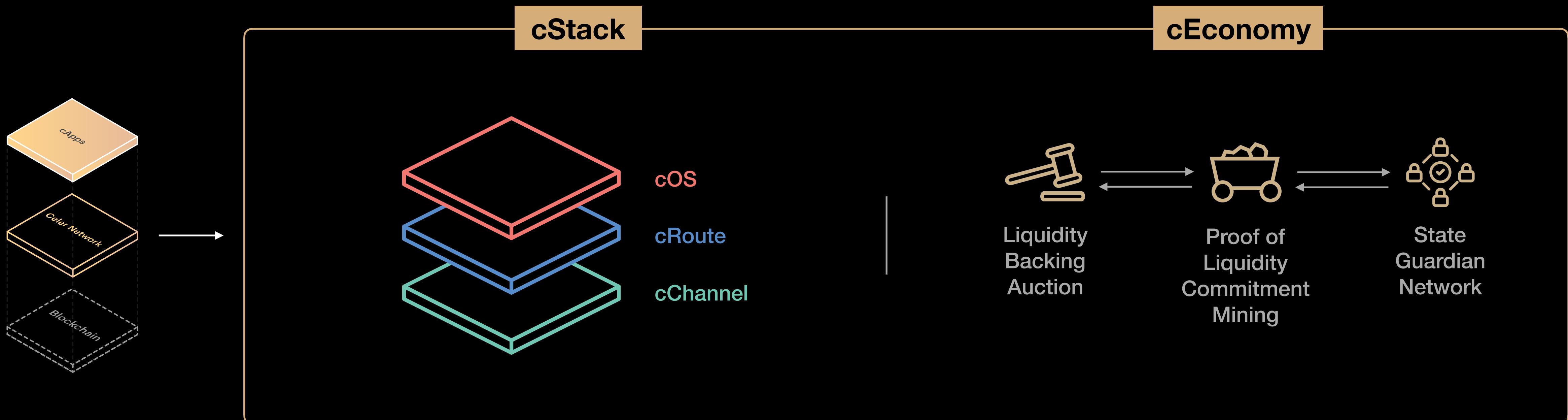
Celer Network

Coherent Off-chain Scaling Architecture



Celer Network

Coherent Off-chain Scaling Architecture



cStack



cOS

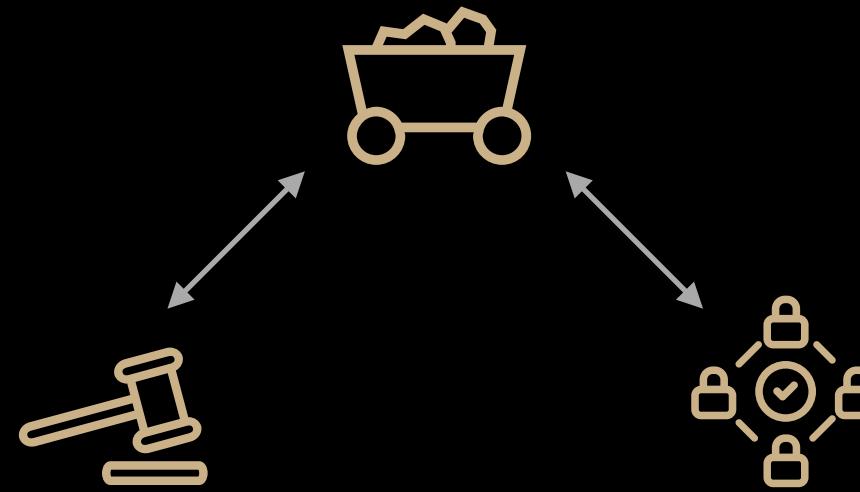
cRoute

cChannel

- **How to support generalized off-chain scaling?**
- **How to route value transfers efficiently in off-chain networks?**
- **How to bring mass adoption to off-chain dApps?**

cEconomy

Proof of Liquidity
Commitment Mining

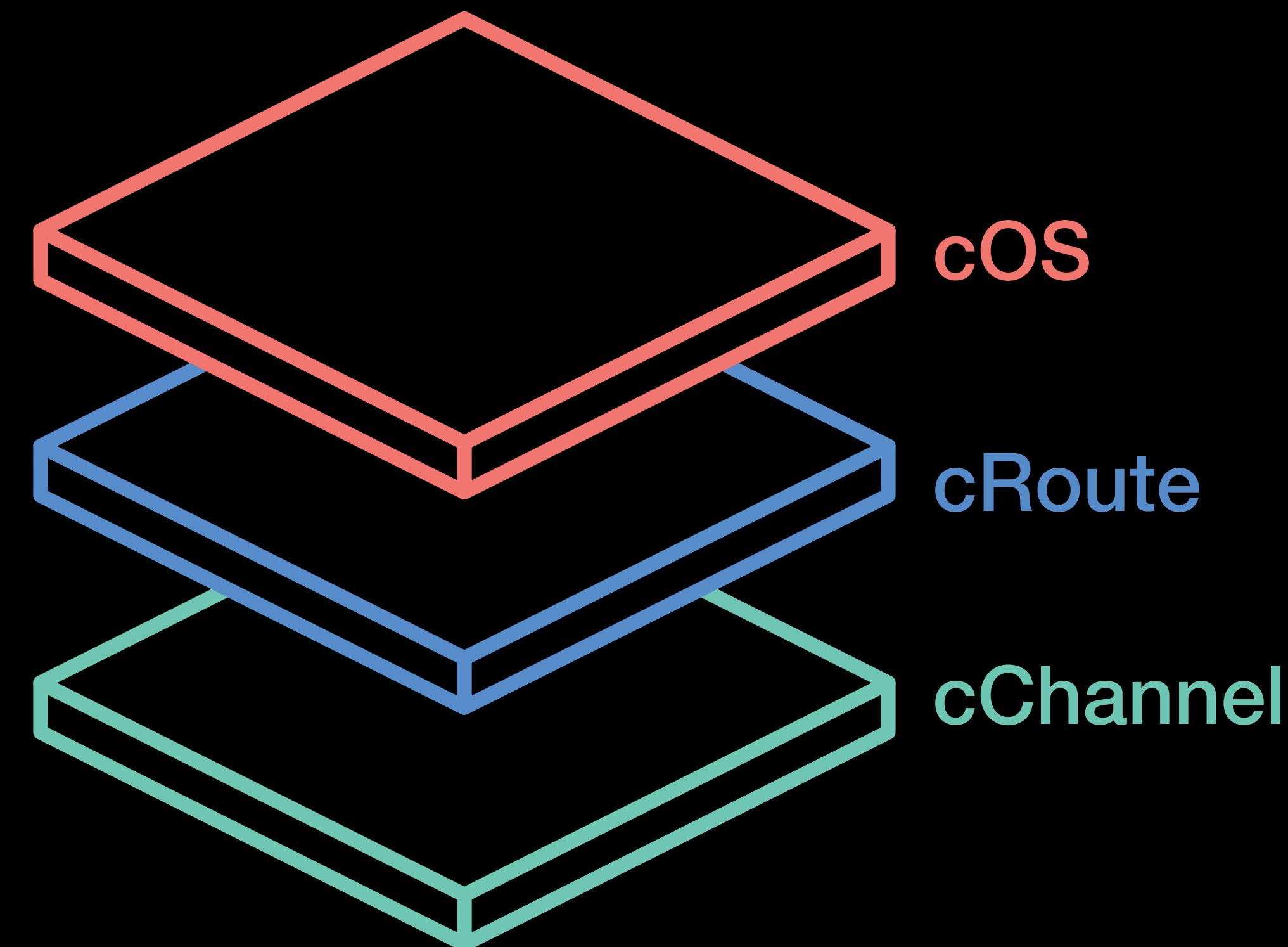


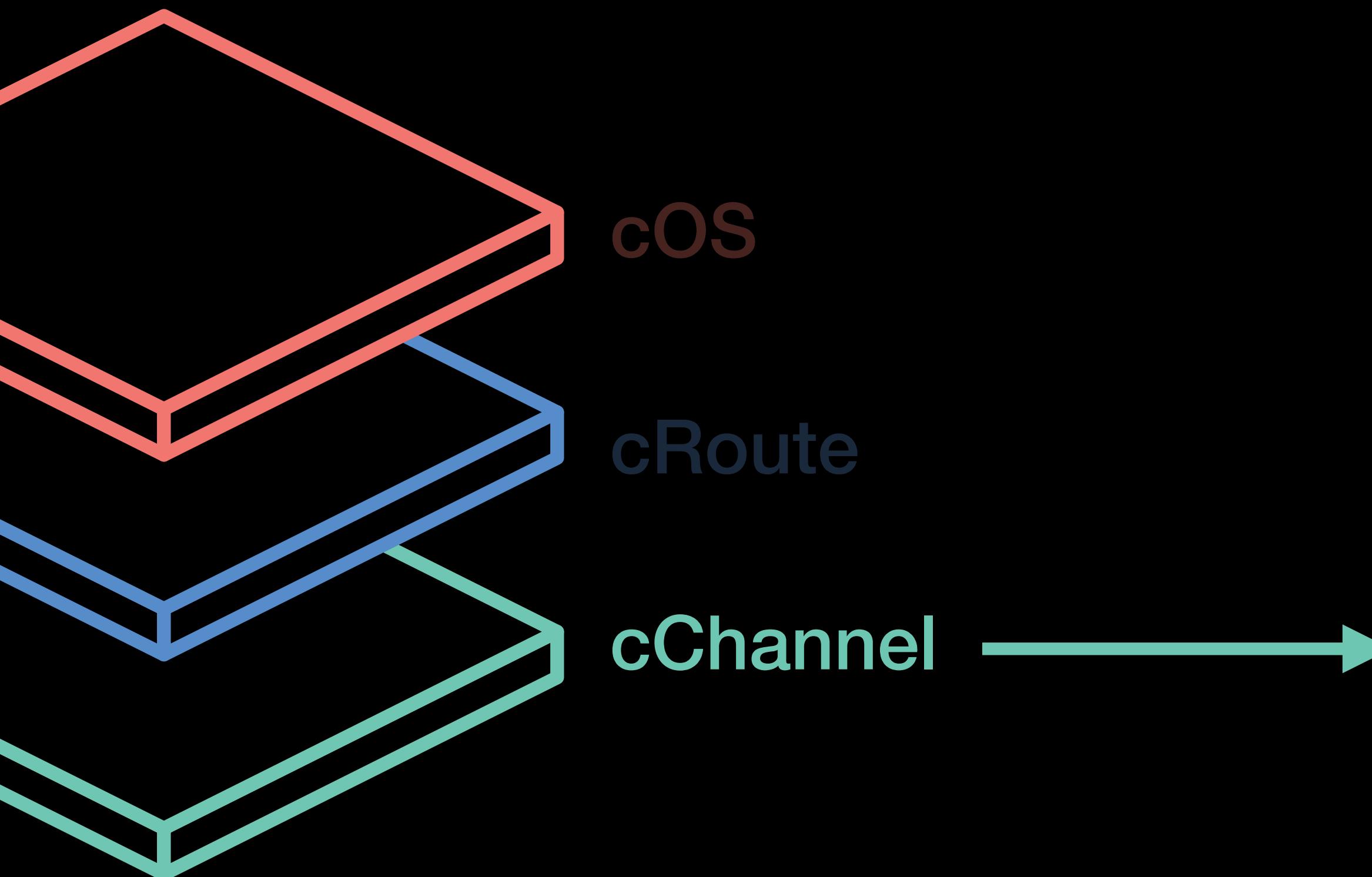
Liquidity Backing
Auction

State Guardian
Network

- **How to make off-chain states always available for on-chain disputes?**
- **How to solve the data connectivity problem?**
- **How to obtain enough liquidity to run an off-chain service?**

cStack



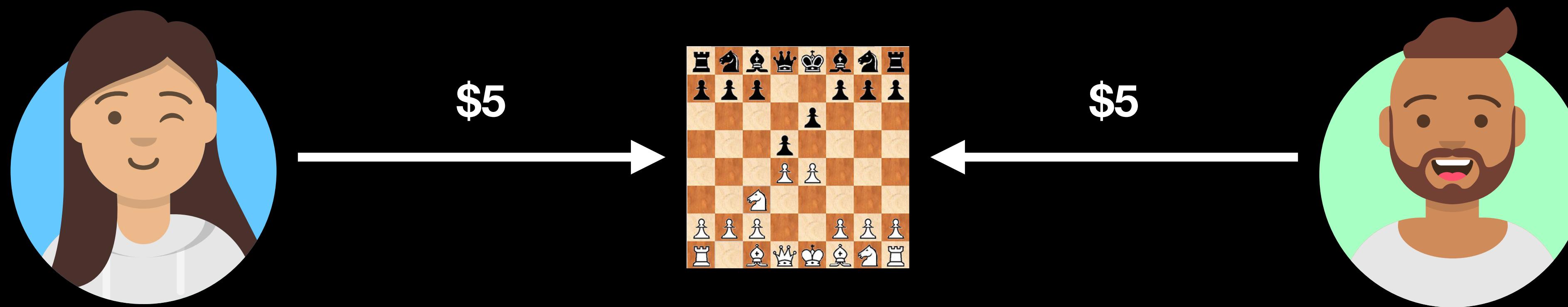


cChannel

- Conditional dependency
- Generalized off-chain dApp
- Low messaging overhead
- State resolution protocol
- Networked channels

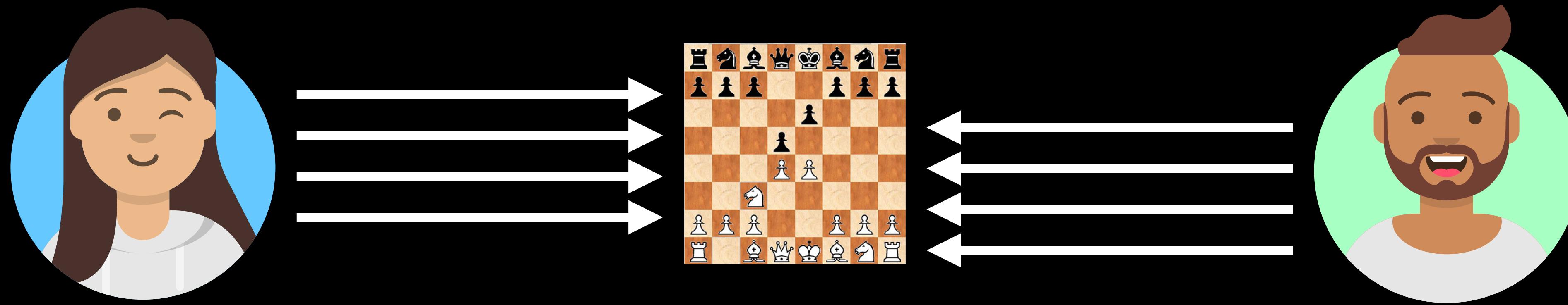
Alice & Bob \$5 Chess Duel

On-chain



Alice & Bob \$5 Chess Duel

🐌 On-chain



Alice & Bob \$5 Chess Duel

🐌 On-chain



\$10

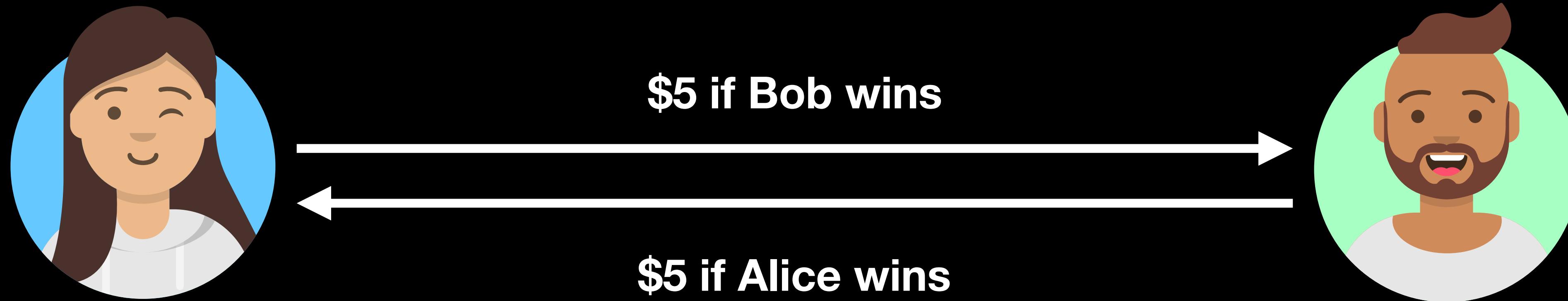
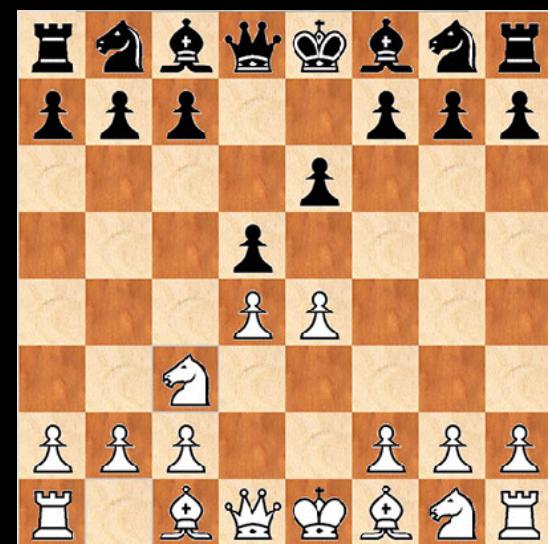


I finally win



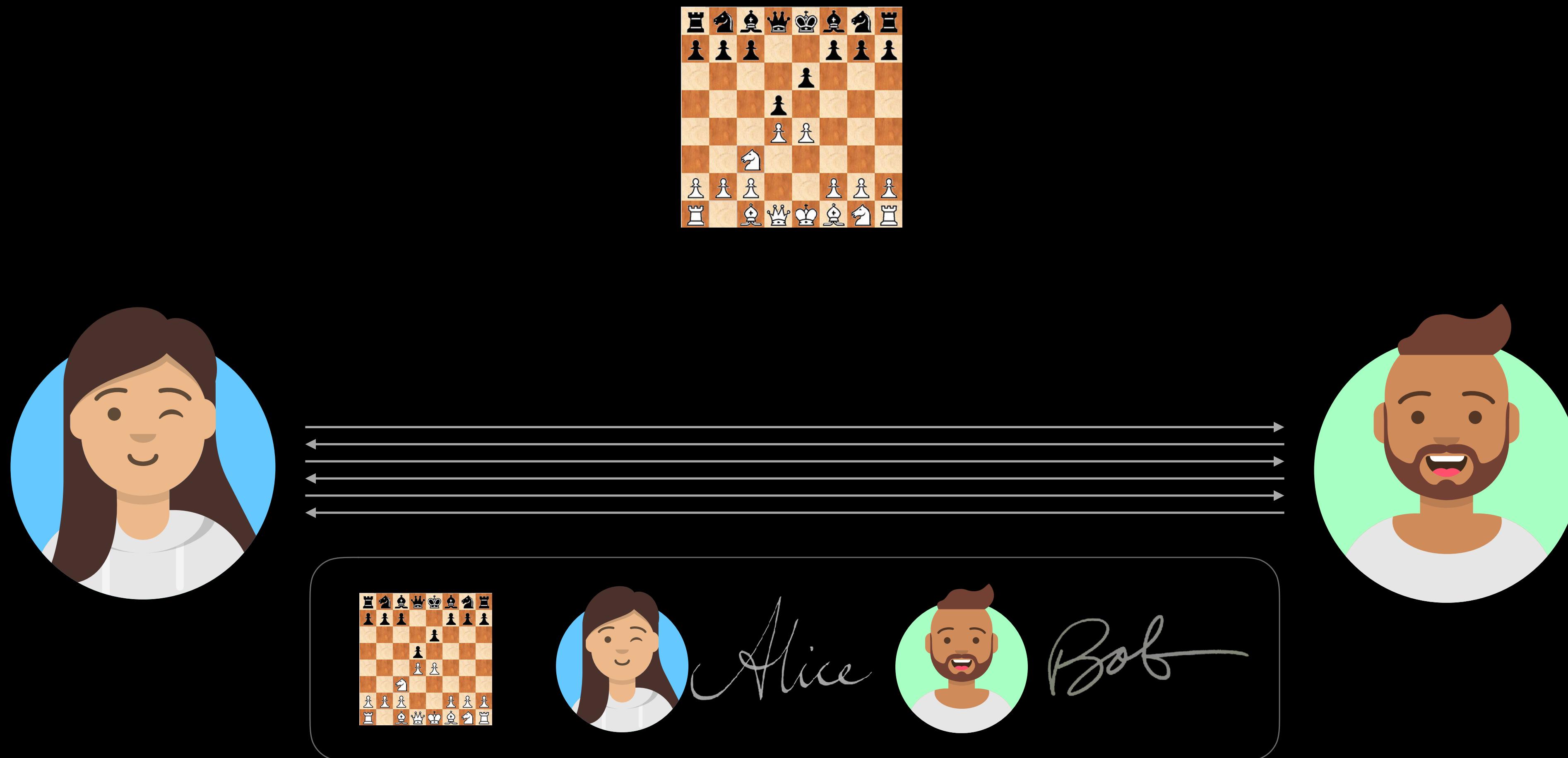
Alice & Bob \$5 Chess Duel

With cChannel



Alice & Bob \$5 Chess Duel

With cChannel



Alice & Bob \$5 Chess Duel

With cChannel



Pure virtual contract



Alice & Bob \$5 Chess Duel

With cChannel



I win !



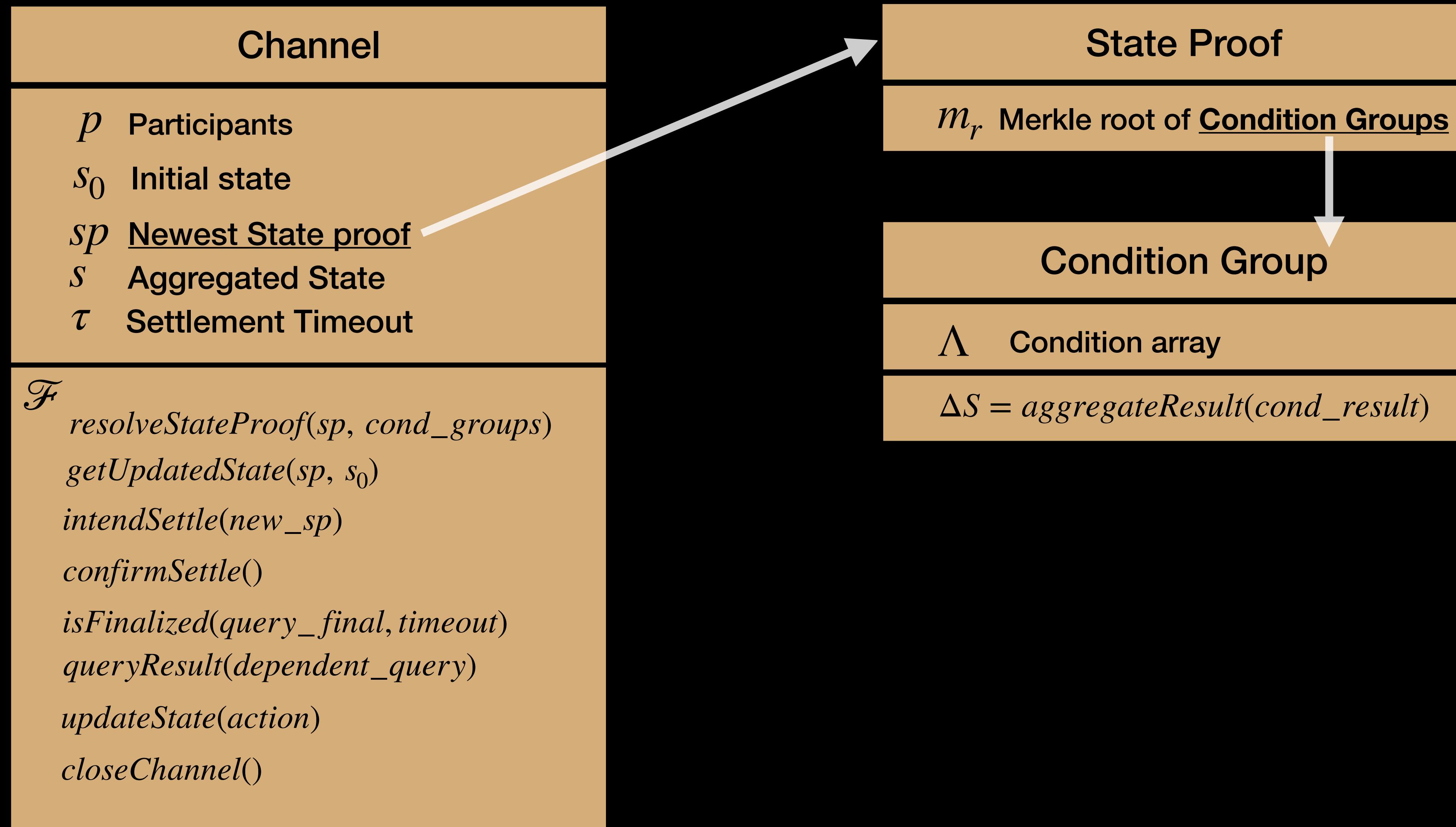
\$5



cChannel: Interface Spec

Channel
p Participants
s_0 Initial state
sp Newest State proof
S Aggregated State
τ Settlement Timeout
\mathcal{F}
$resolveStateProof(sp, cond_groups)$
$getUpdatedState(sp, s_0)$
$intendSettle(new_sp)$
$confirmSettle()$
$isFinalized(query_final, timeout)$
$queryResult(dependent_query)$
$updateState(action)$
$closeChannel()$

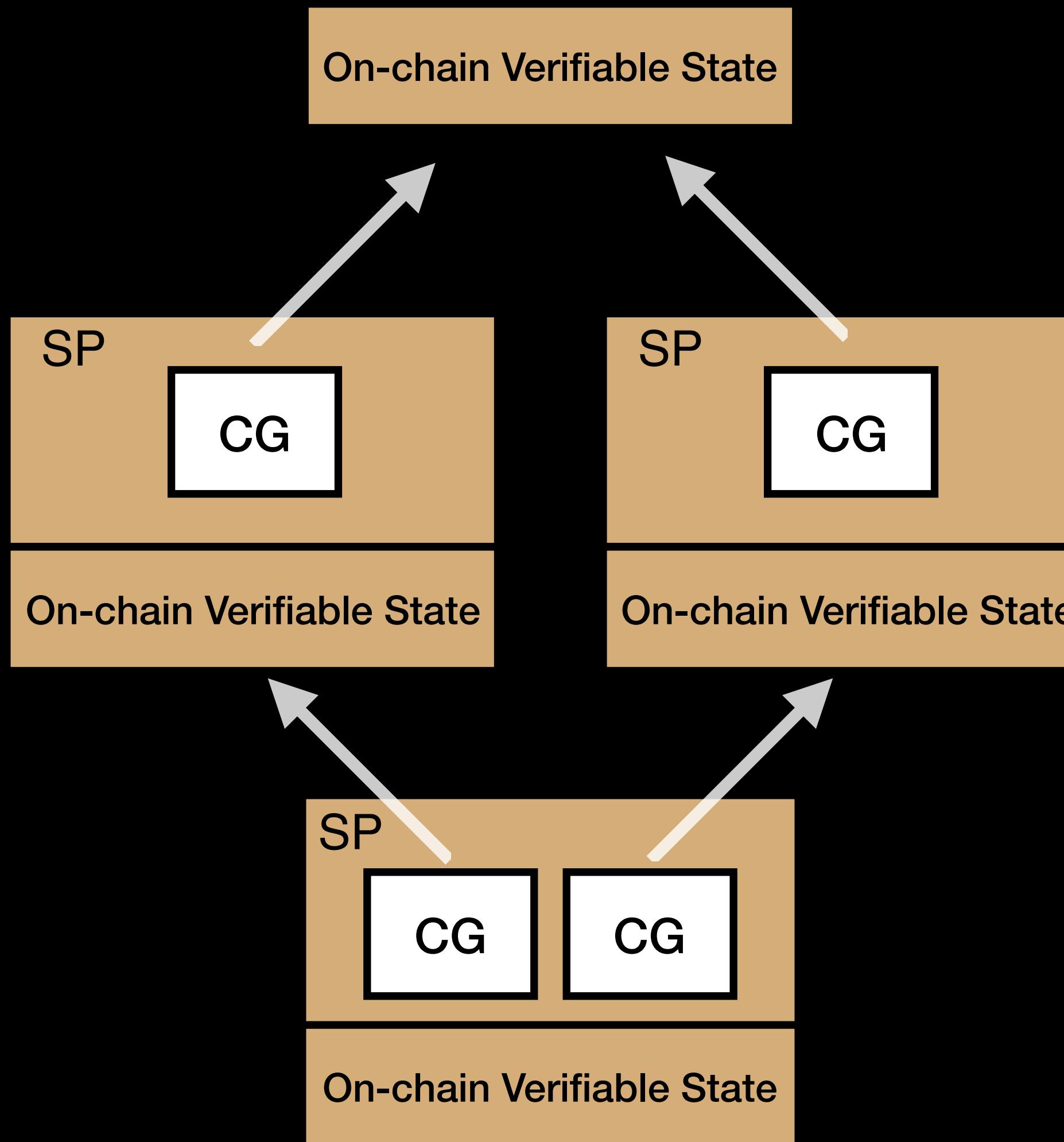
cChannel: Interface Spec



cChannel: Interface Spec

Channel	Condition Group
p Participants	Λ Condition Array
s_0 Initial state	$aggregateResult(cond_result)$
sp <u>Newest State proof</u>	
S Aggregated State	
τ Settlement Timeout	
\mathcal{F}	Condition
$resolveStateProof(sp, cond_groups)$	$timeout$ Condition timeout
$getUpdatedState(sp, s_0)$	$finalize_p(query_final, timeout)$
$intendSettle(new_sp)$	$queryResult_p(dependent_query)$
$confirmSettle()$	
$isFinalized(query_final, timeout)$	
$queryResult(dependent_query)$	
$updateState(action)$	
$closeChannel()$	

cChannel: Conditional Dependency DAG

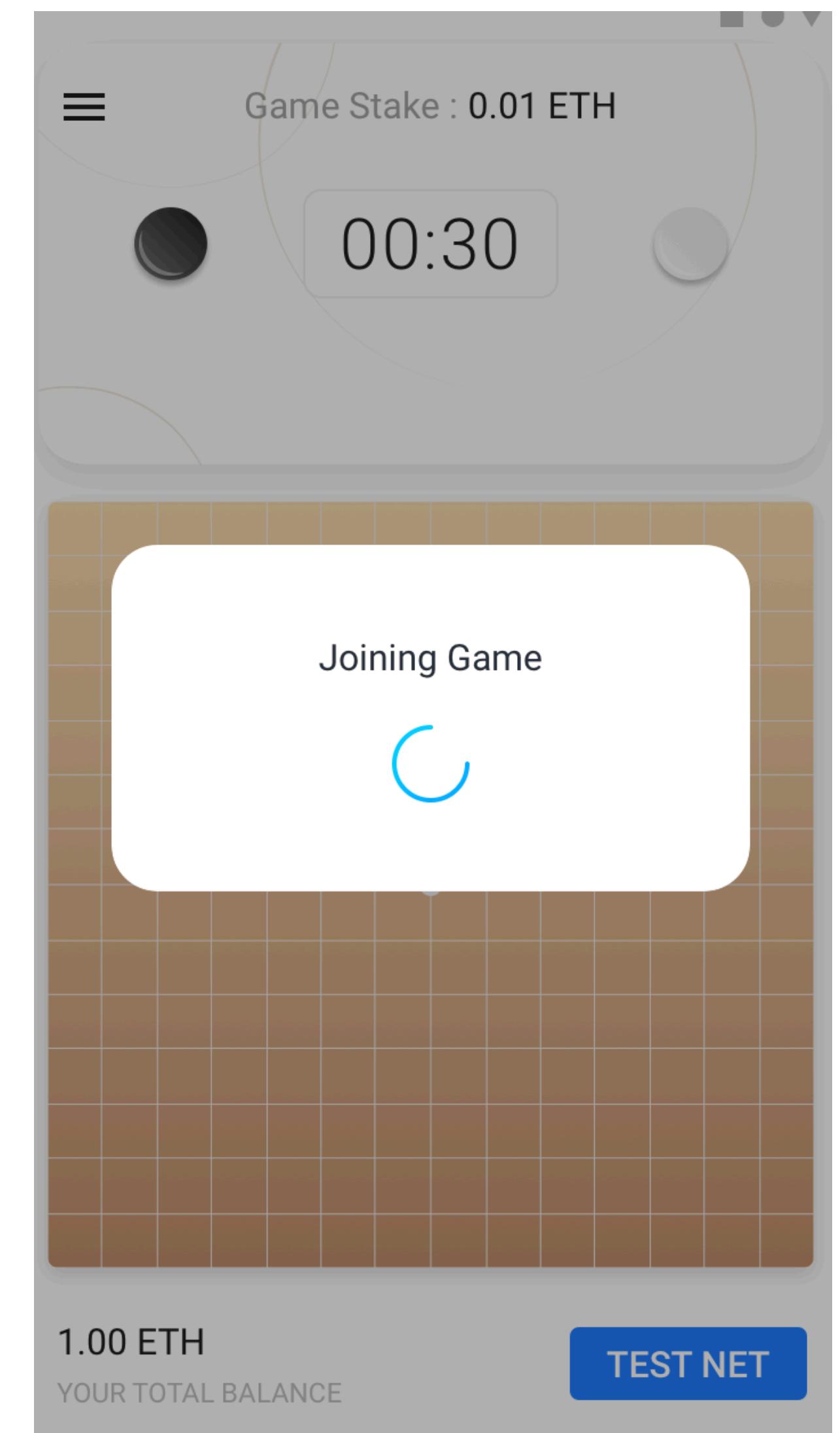
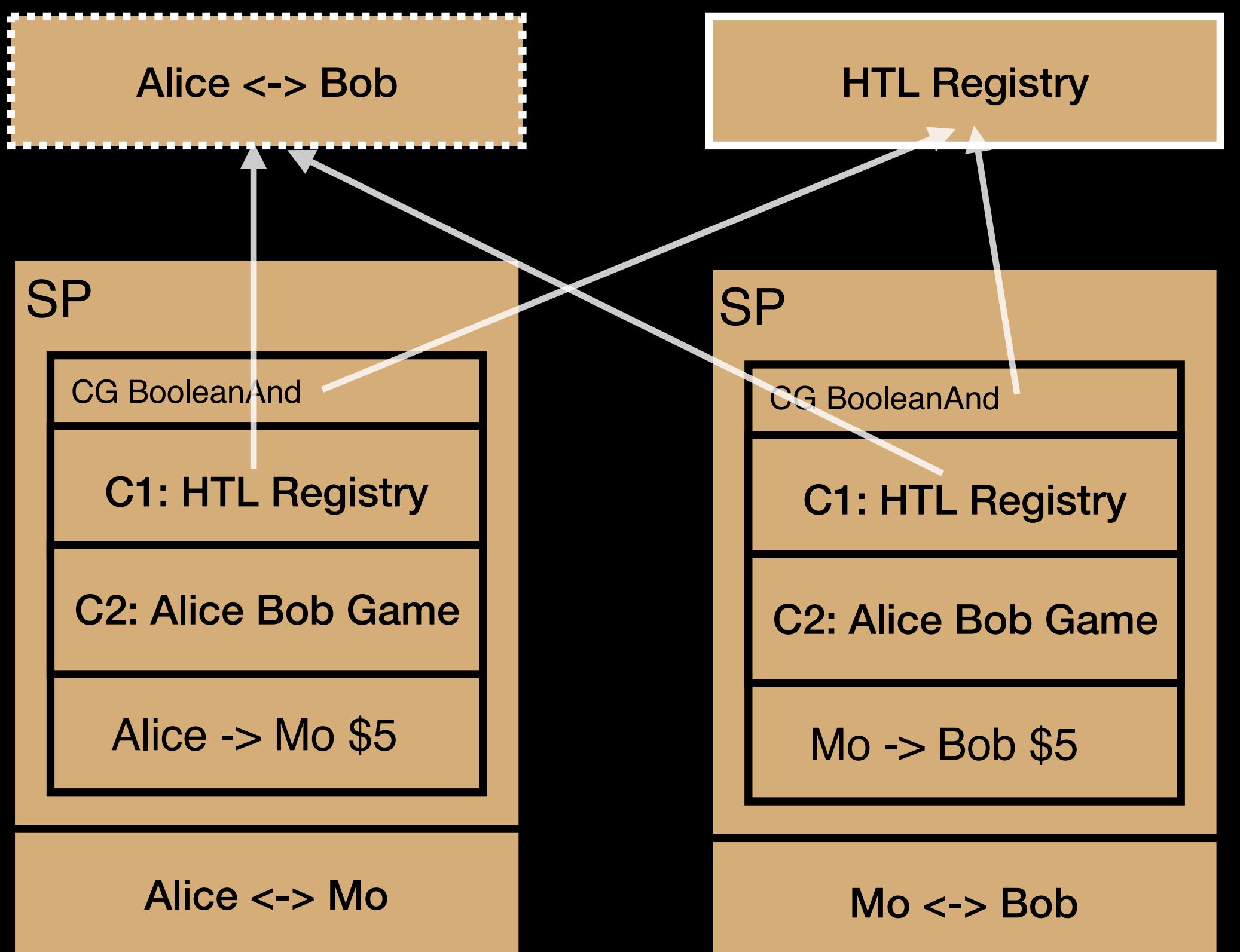


On-chain Verifiable

1. Actually On-chain if there is any state race
2. Can be “Virtual”, but can be deterministically referred and has no possible state race

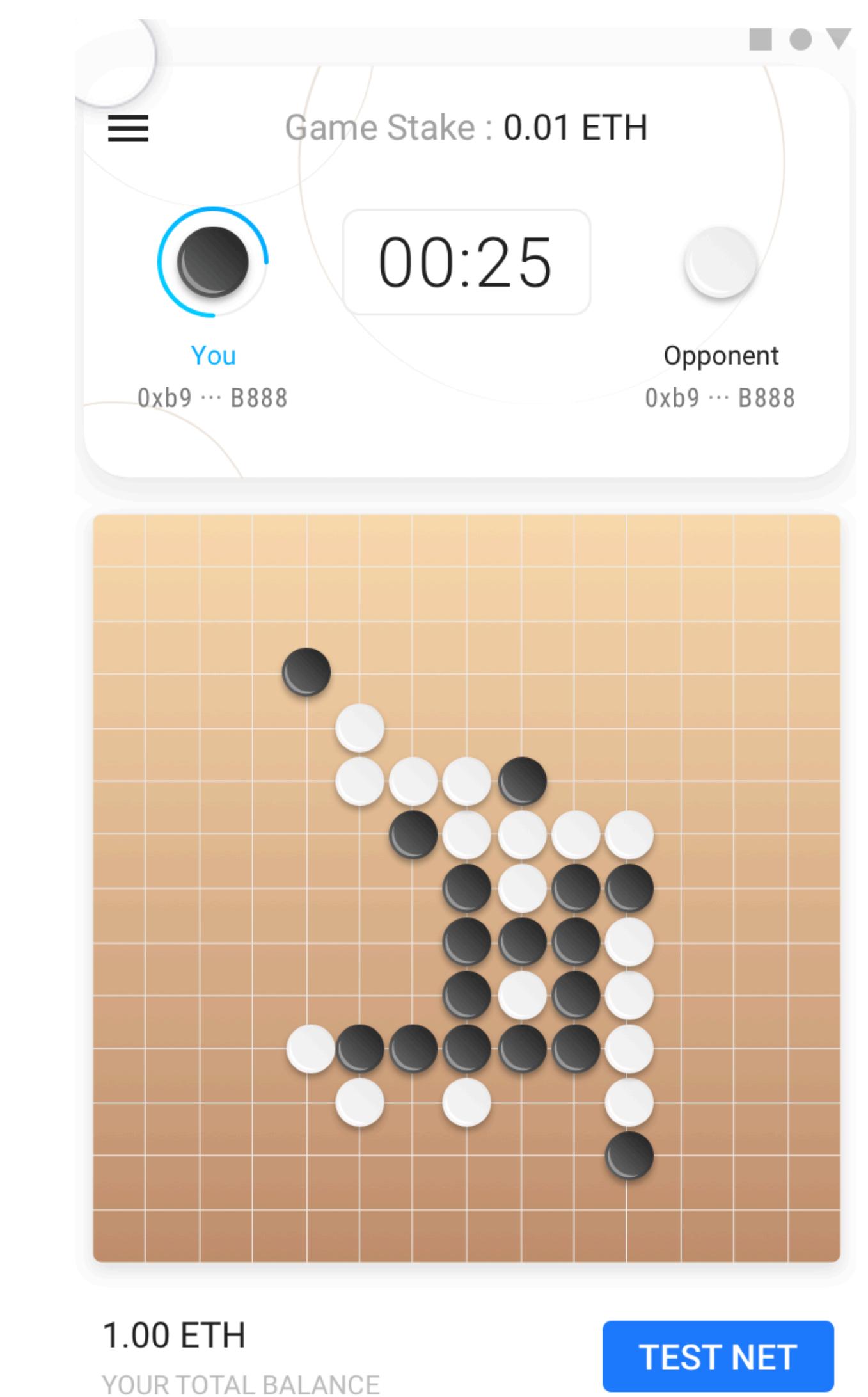
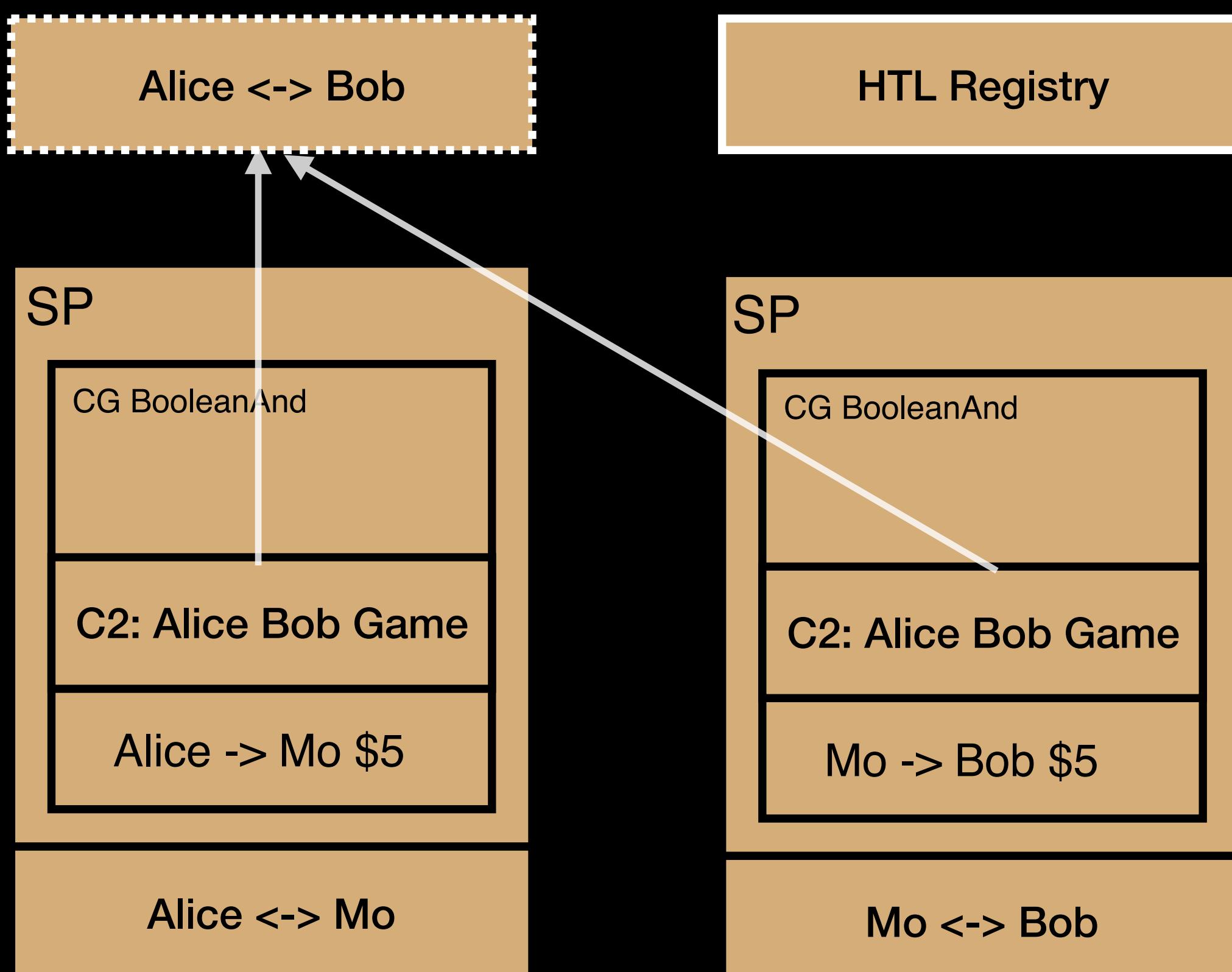
cChannel: Out-of-box features

Conditional multi-hop payment



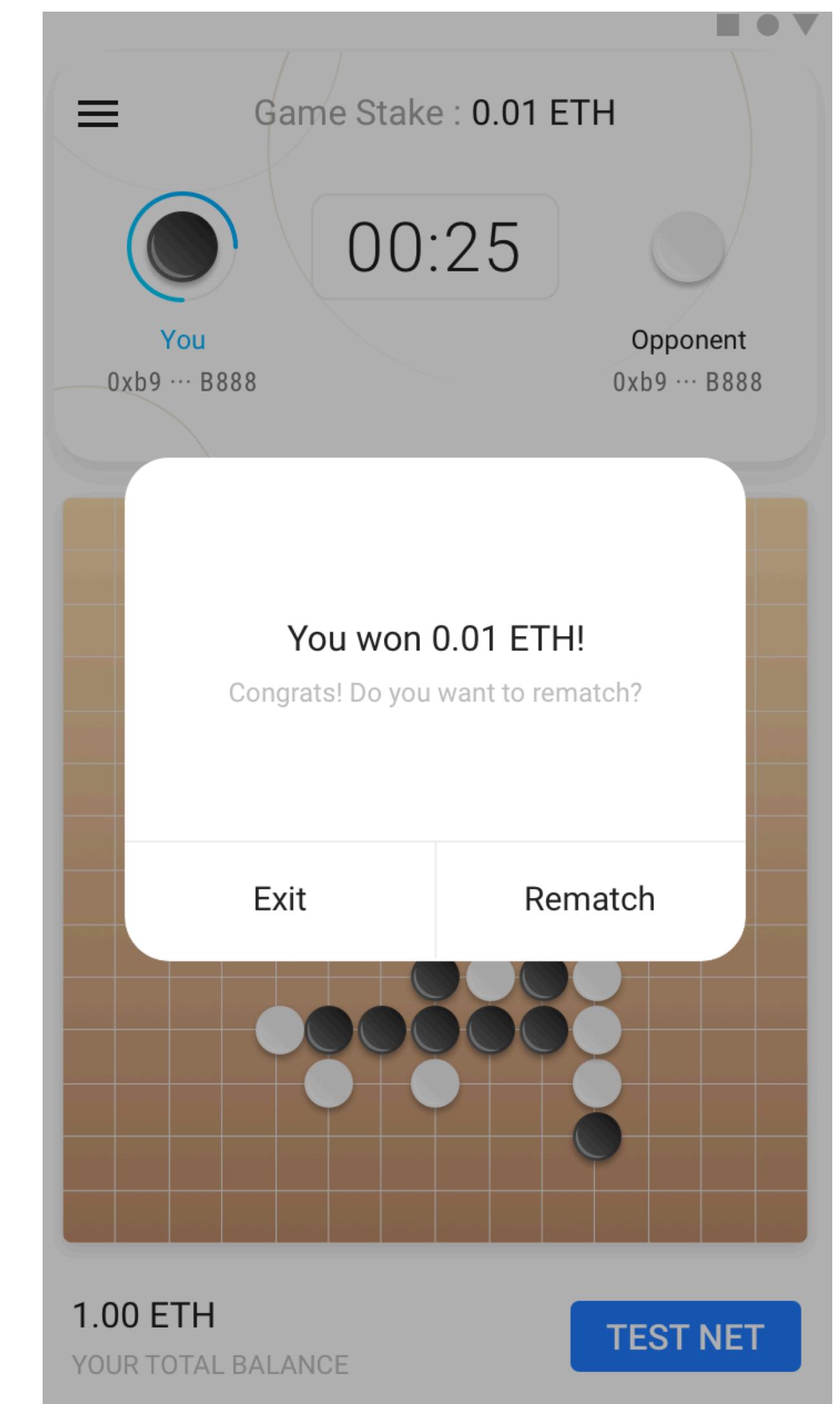
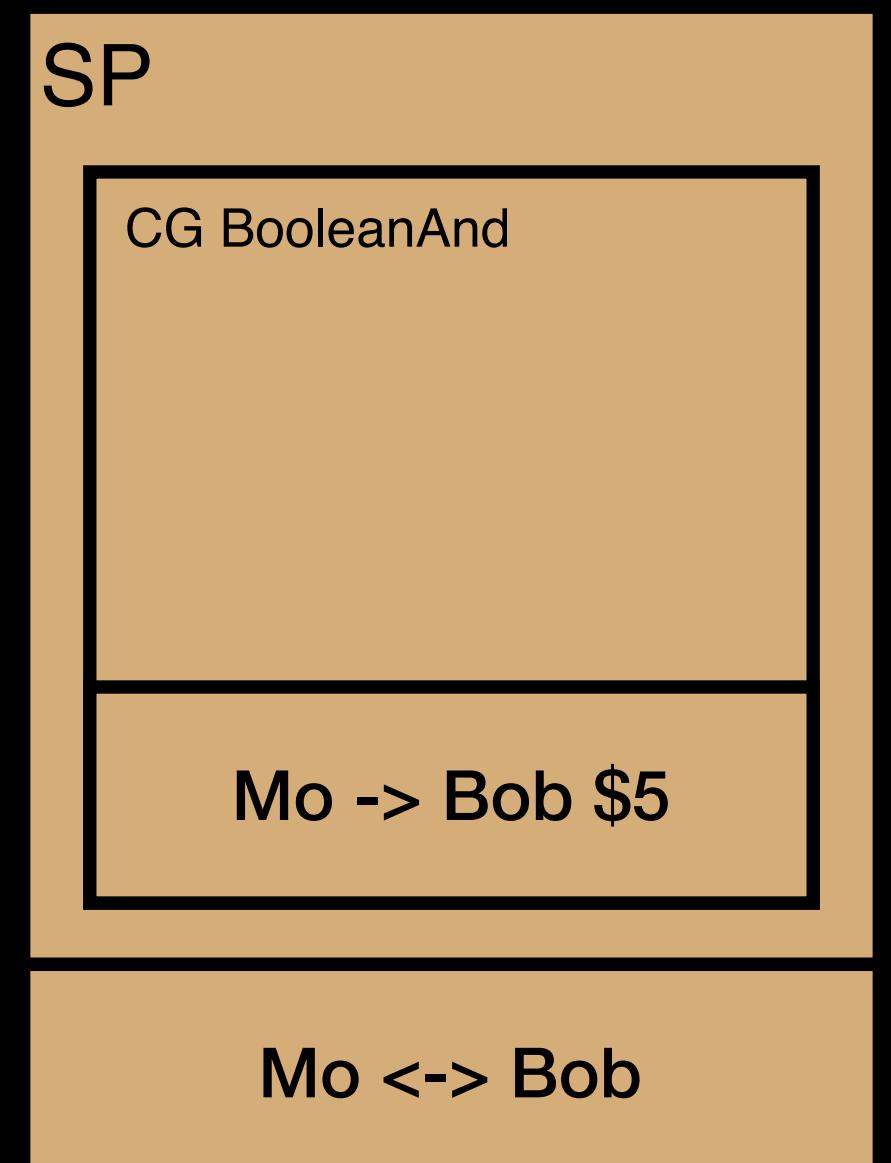
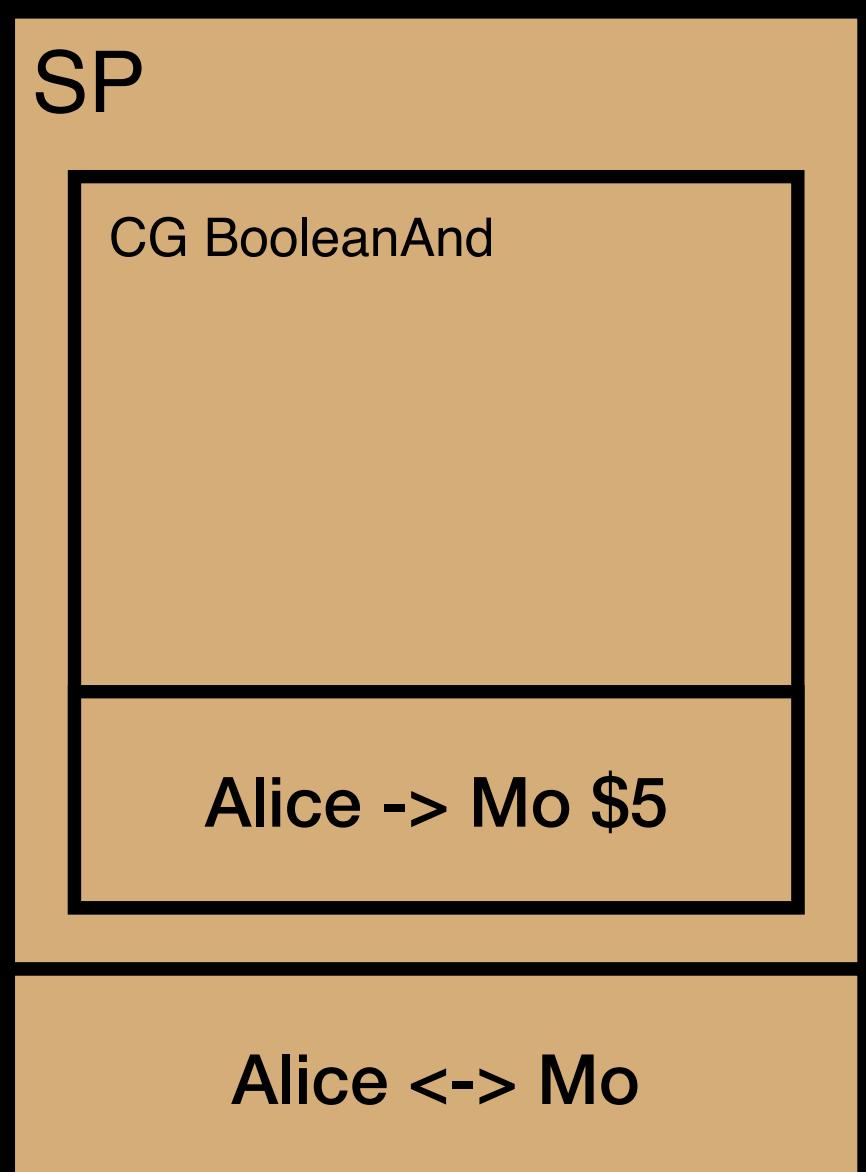
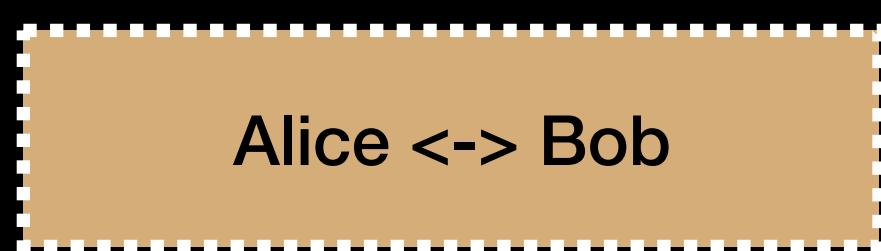
cChannel: Out-of-box features

Conditional multi-hop payment
+ Pure Off-chain Object



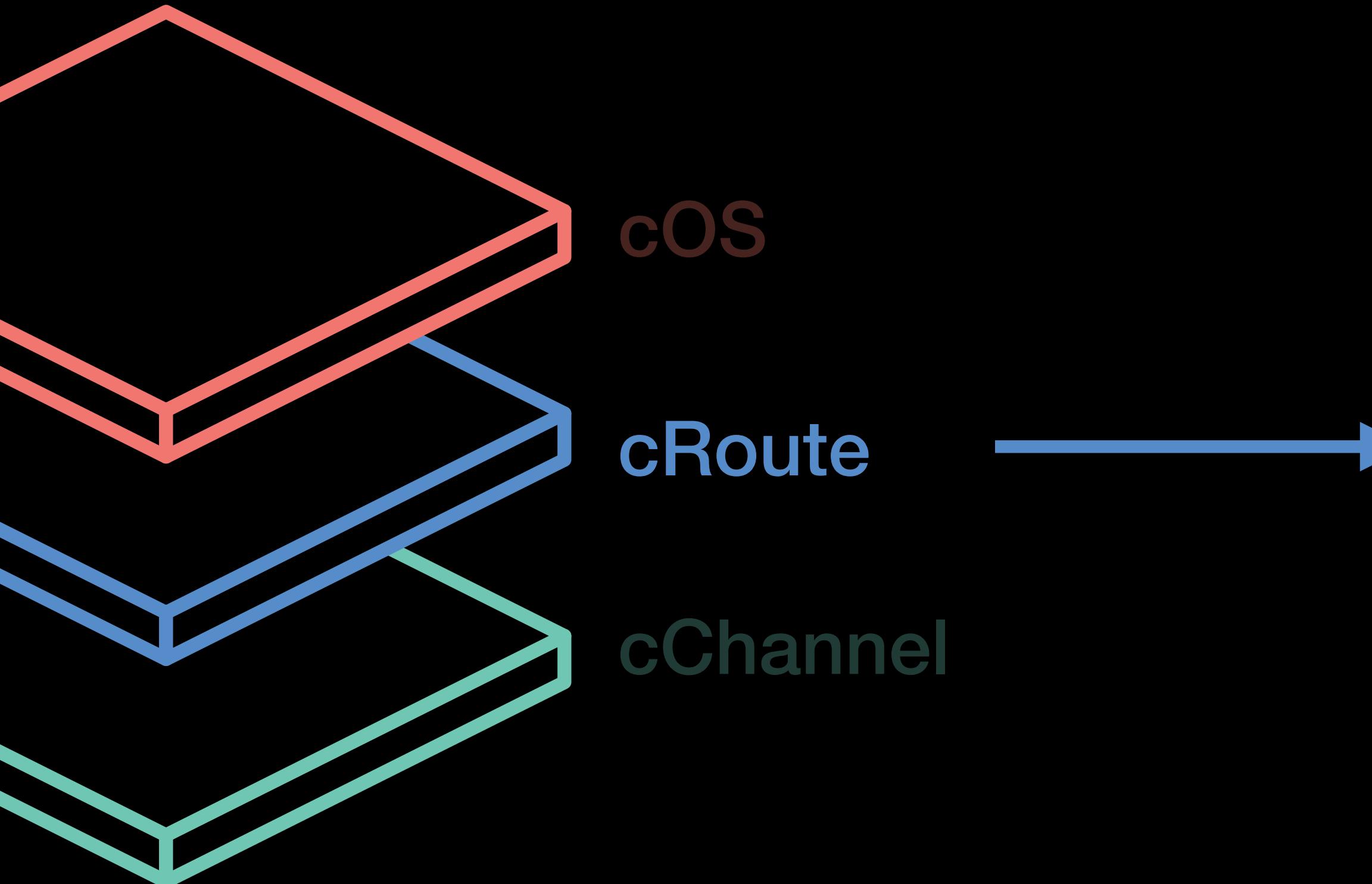
cChannel: Out-of-box features

Conditional multi-hop payment
+ Pure Off-chain Object



cChannel : Out-of-box features

- Off-chain Address Translator
- Cooperative settling
- Single-tx channel setup
- Direct final state claim
- Dynamic Deposit/Withdrawal
- Boolean circuit condition group
- Fund assignment condition group



cRoute

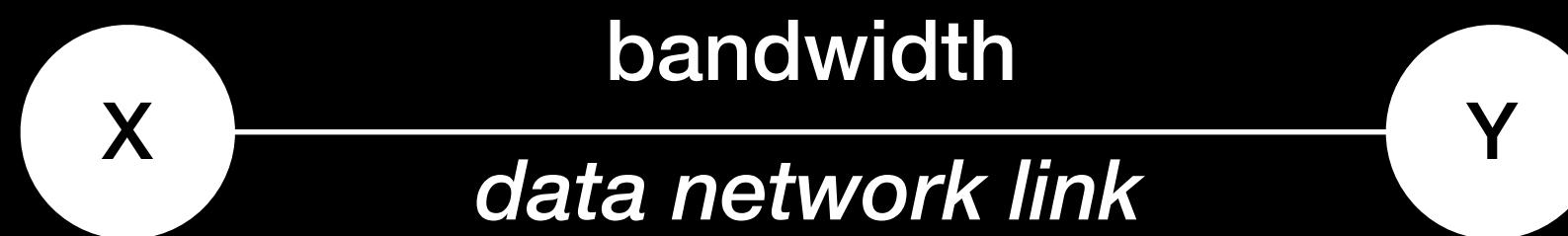
- “Connect once, use forever”
- Transparent channel balancing
- Fully decentralized
- Provably optimal payment routing
- High failure resilience

Why off-chain payment routing is challenging?

Off-chain payment networks are fundamentally different from data networks

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

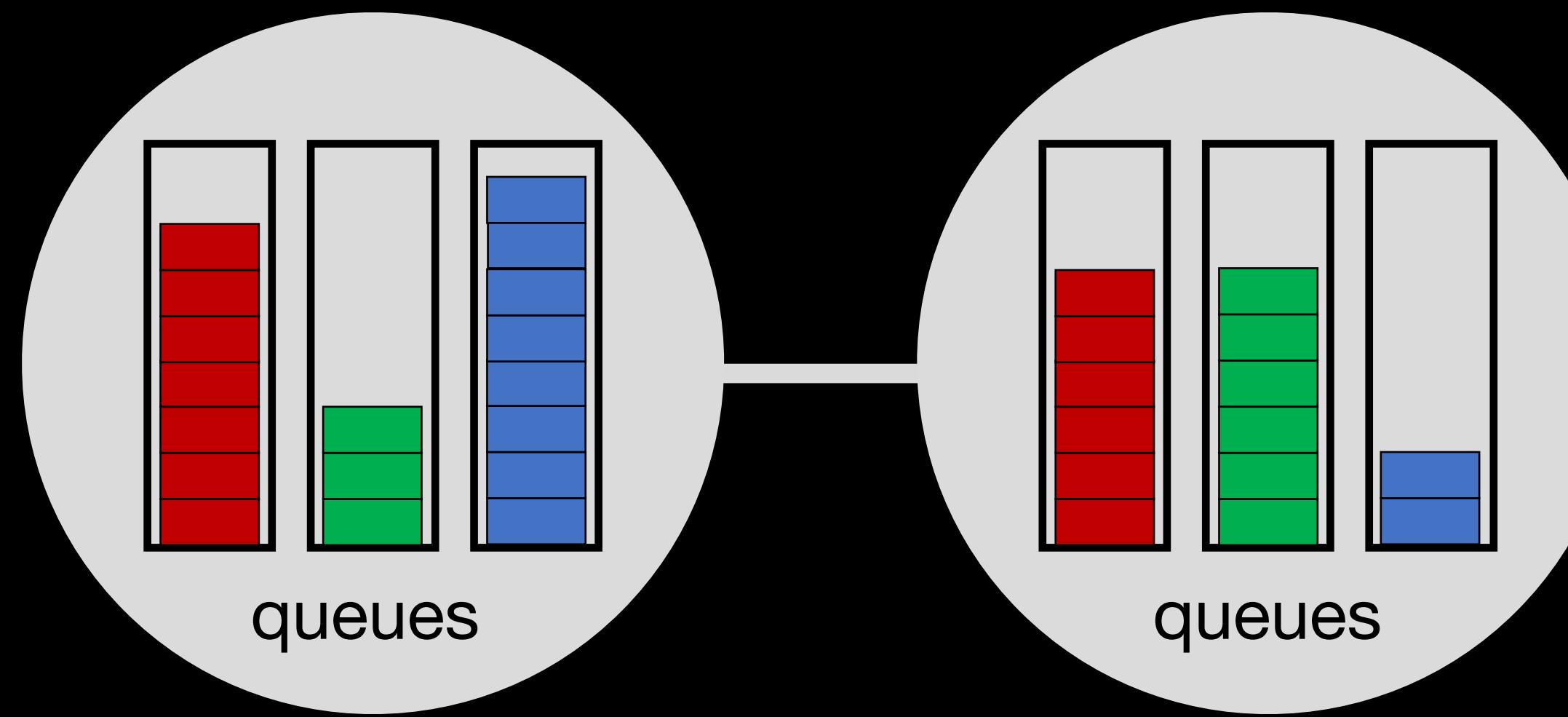
- Link state changed by every payment
- Max rate ranges from zero to *infi*
- Network is constantly changing

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

cRoute Principle 1: Follow Congestion Gradients

Route in directions that maximize the differential backlog between neighbors

Similar to water flowing through a network of pipes via pressure gradients



Node A
Balance = \$40

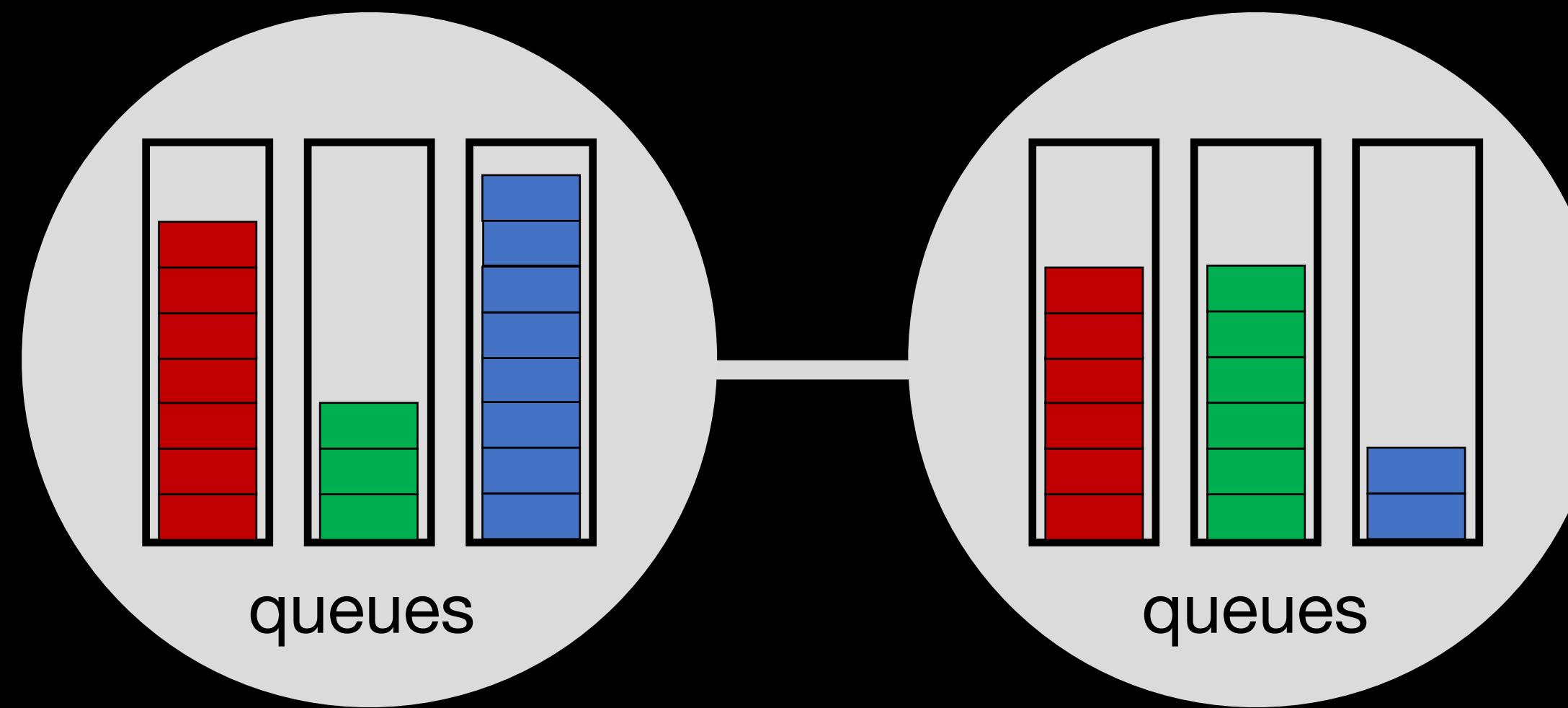
Node B
Balance = \$60

Route decision in current time slot:
Node A: send **BLUE** commodity over link (AB)
Node B: send **GREEN** commodity over link (BA)

cRoute Principle 2: Keep Channels Balanced

Take channel balance into consideration

Tunable parameter to adjust channel balance and differential backlog together



Node A
Balance = \$90

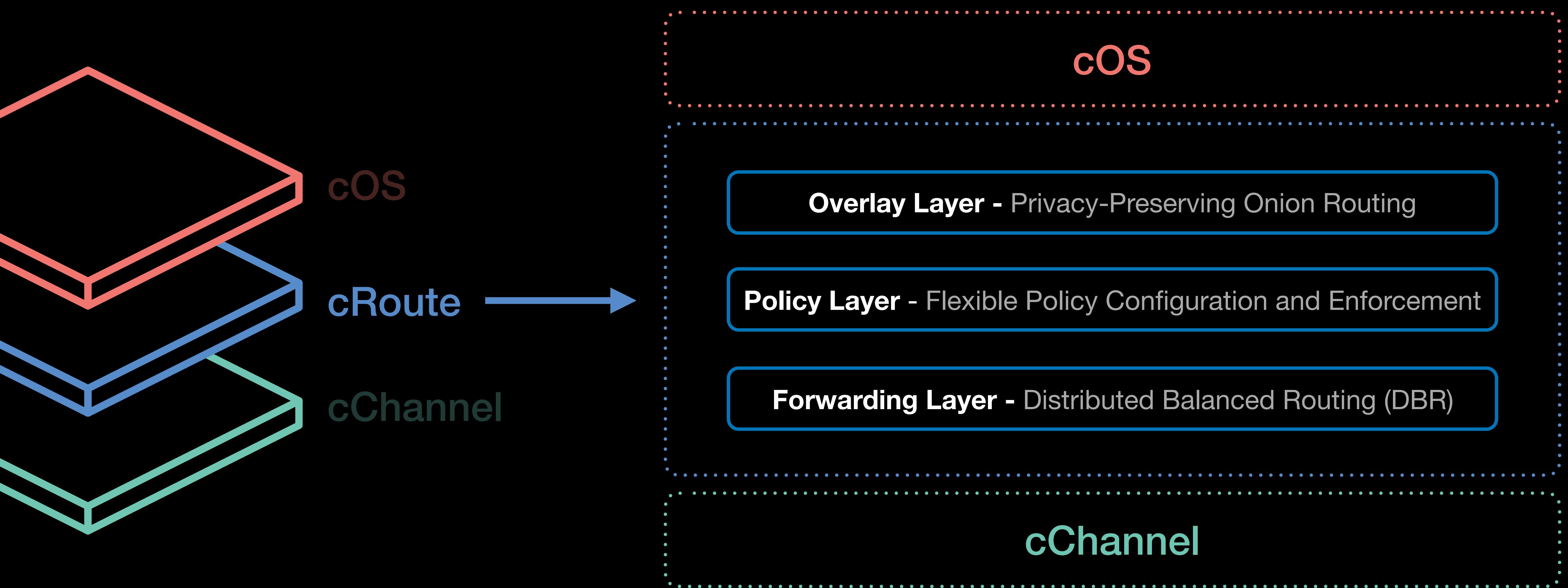
Node B
Balance = \$10

Route decision in current time slot:

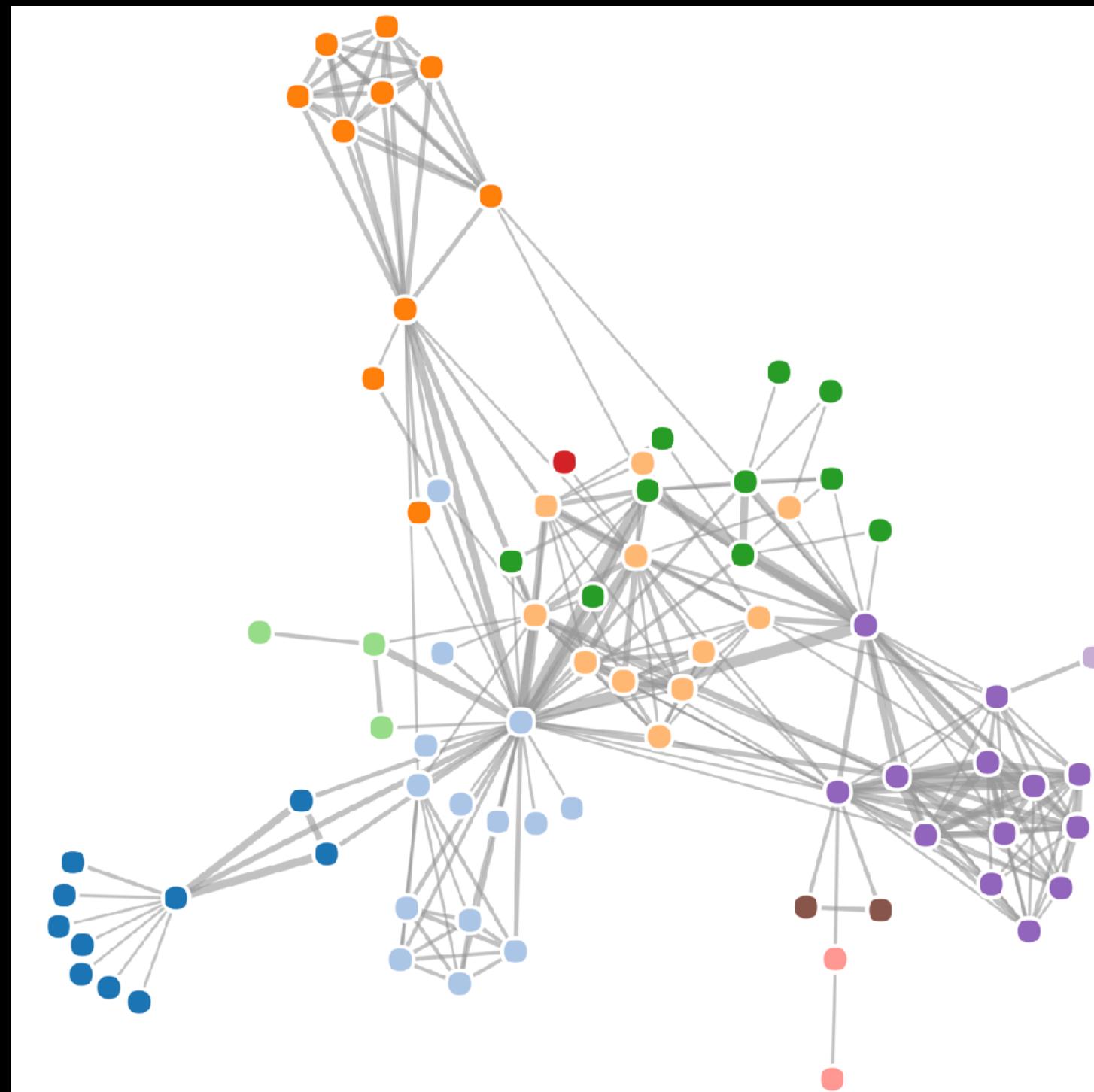
Node A: send **BLUE** commodity over link (AB)

Node B: wait till backlog gets larger or channel gets more balanced

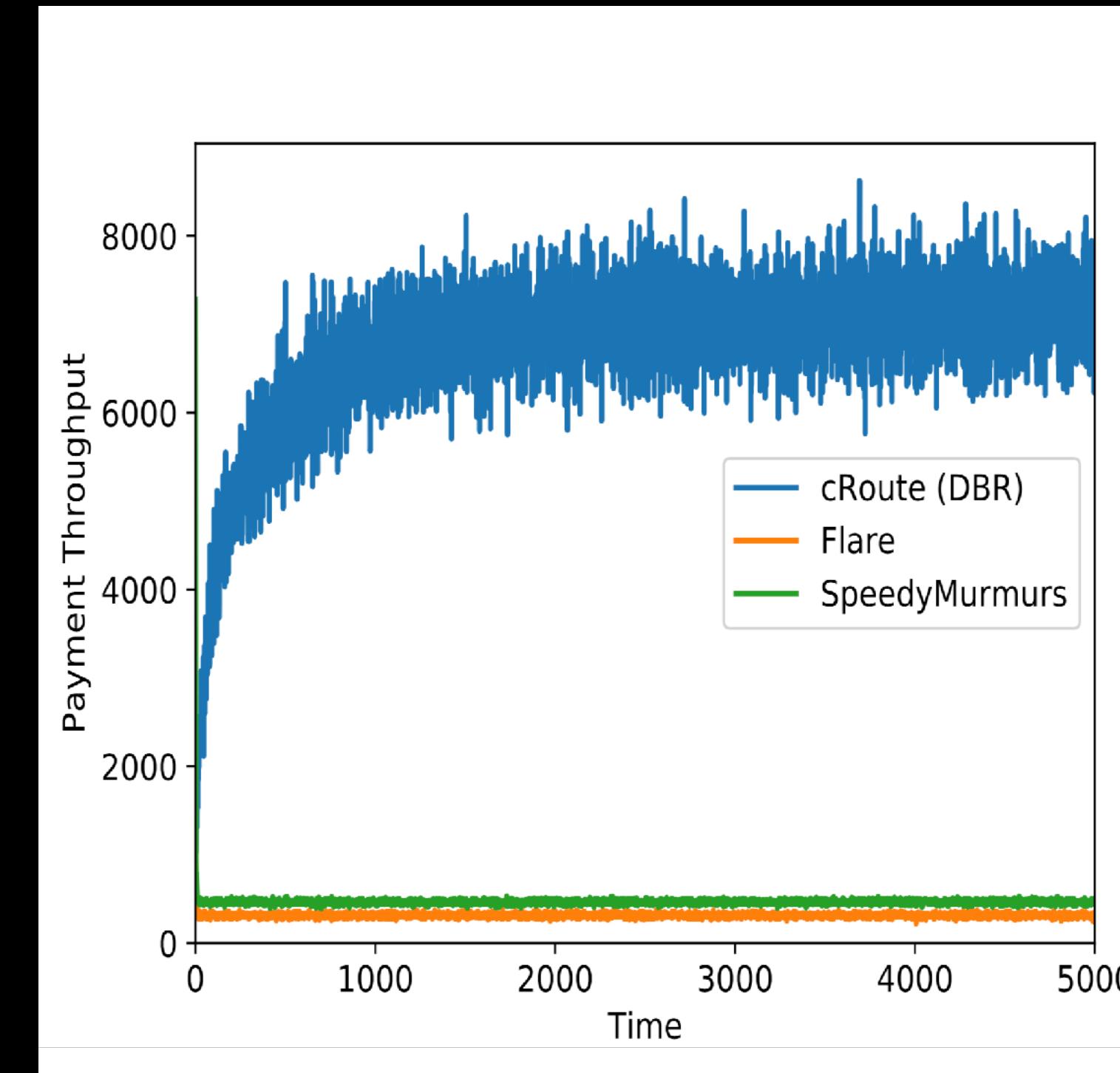
cRoute - Highly Efficient Payment Routing



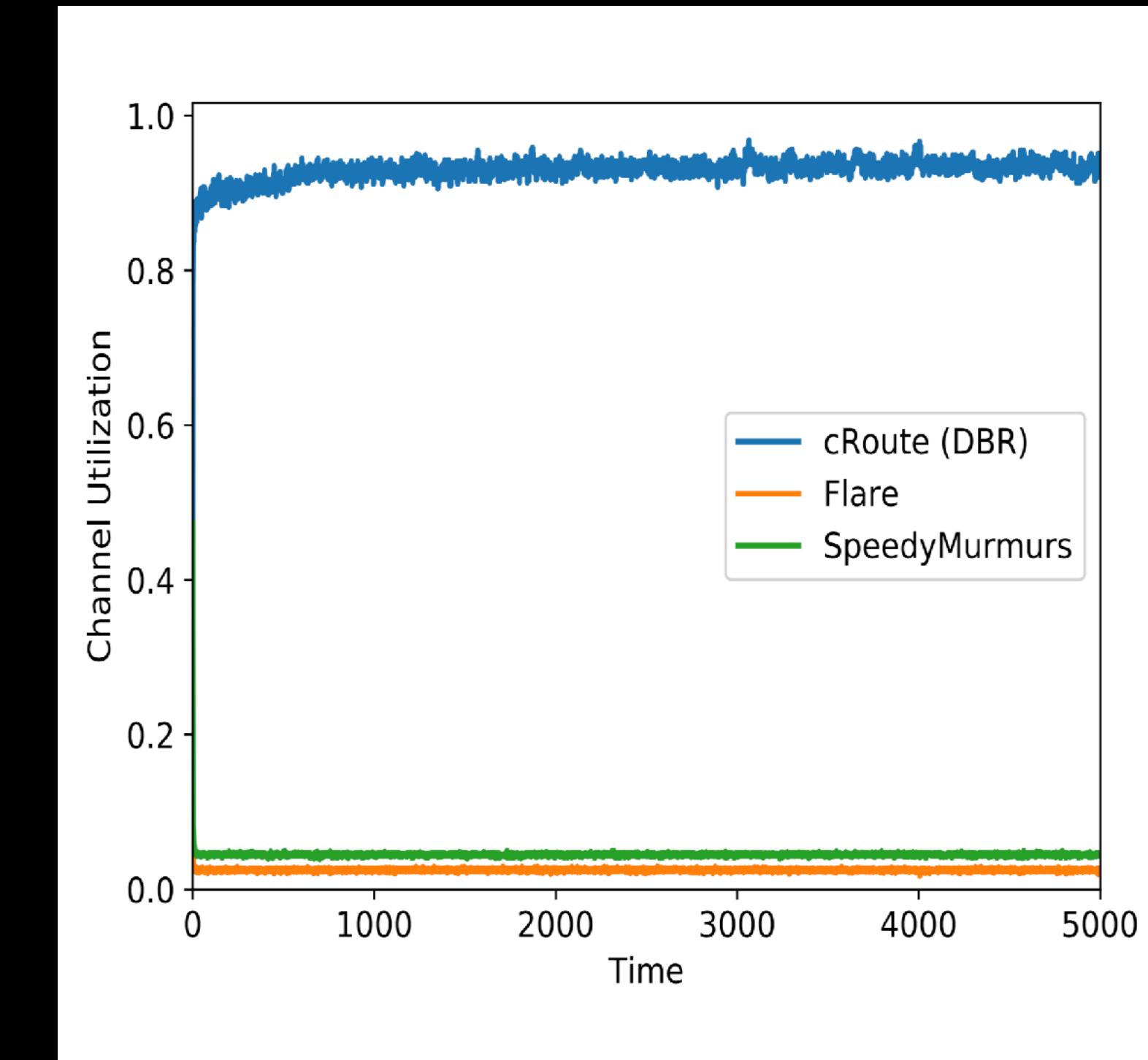
cRoute Simulation Results



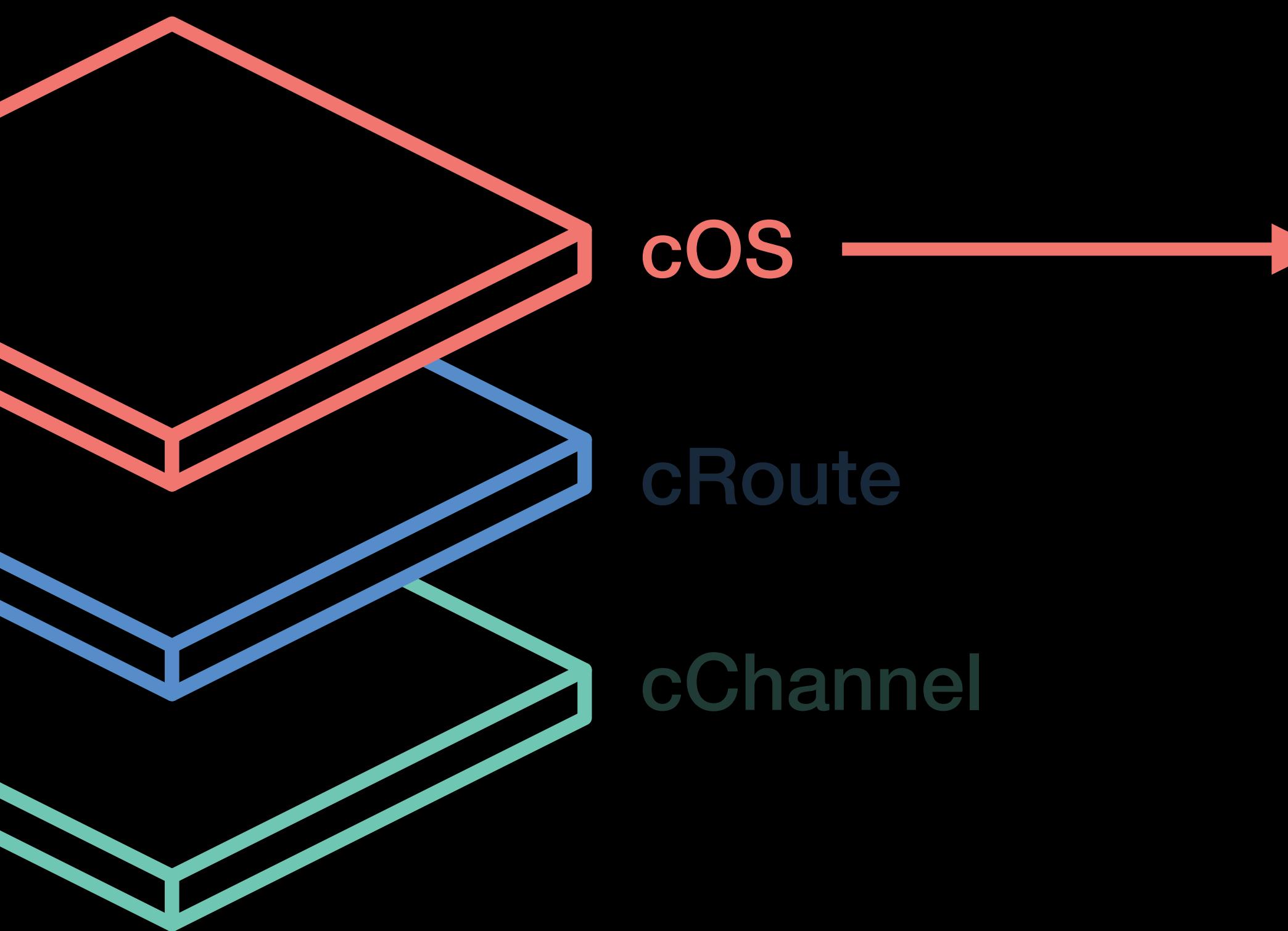
77 nodes
254 bi-directional payment channel
Poisson arrival with random src-dst



Payment throughput 20X of Lightning



Close to 100%
Channel utilization

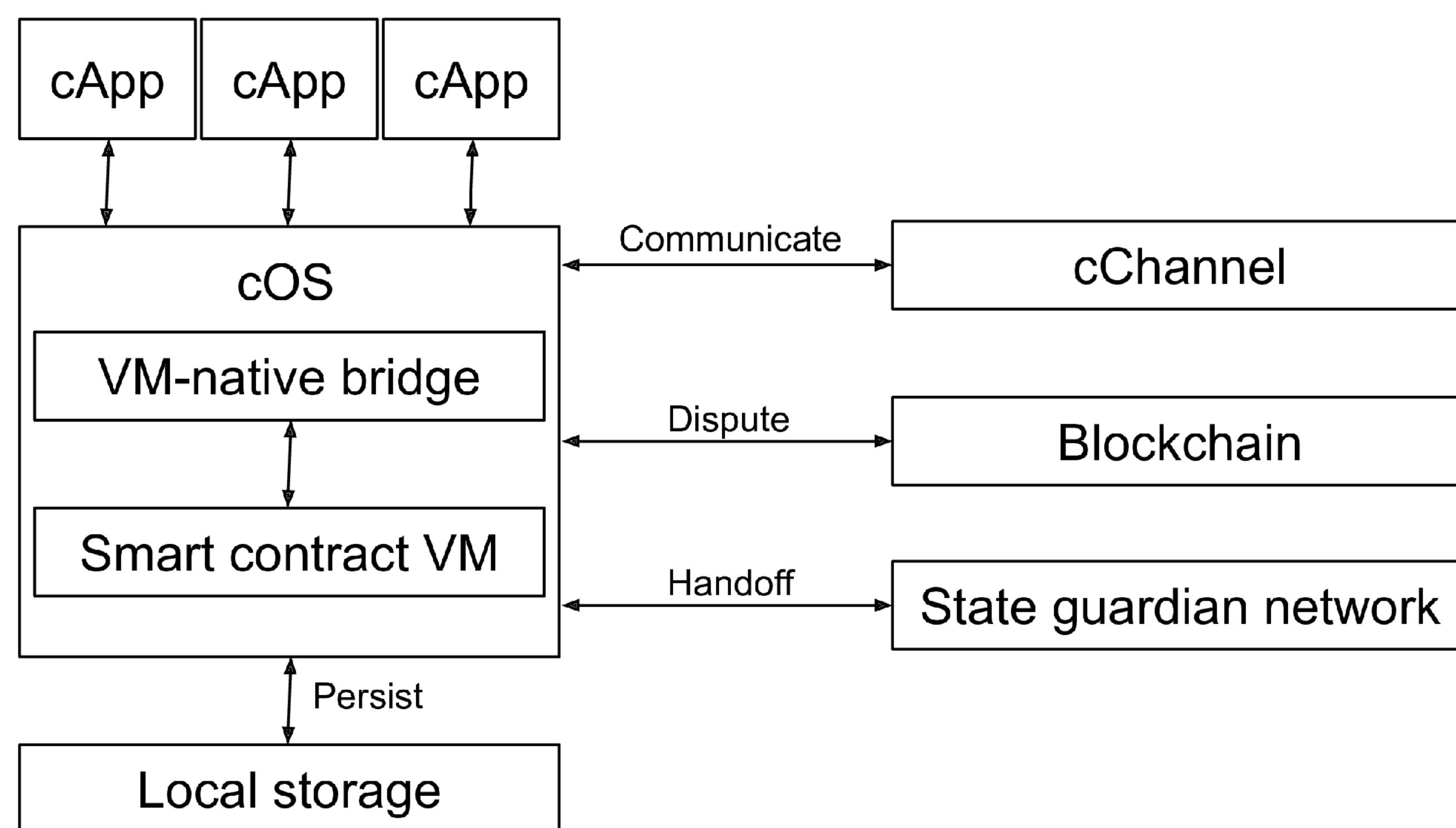


cOS

- Developer framework for cApps
- Bridging on-chain and off-chain byte codes
- Easy-to-use user runtime

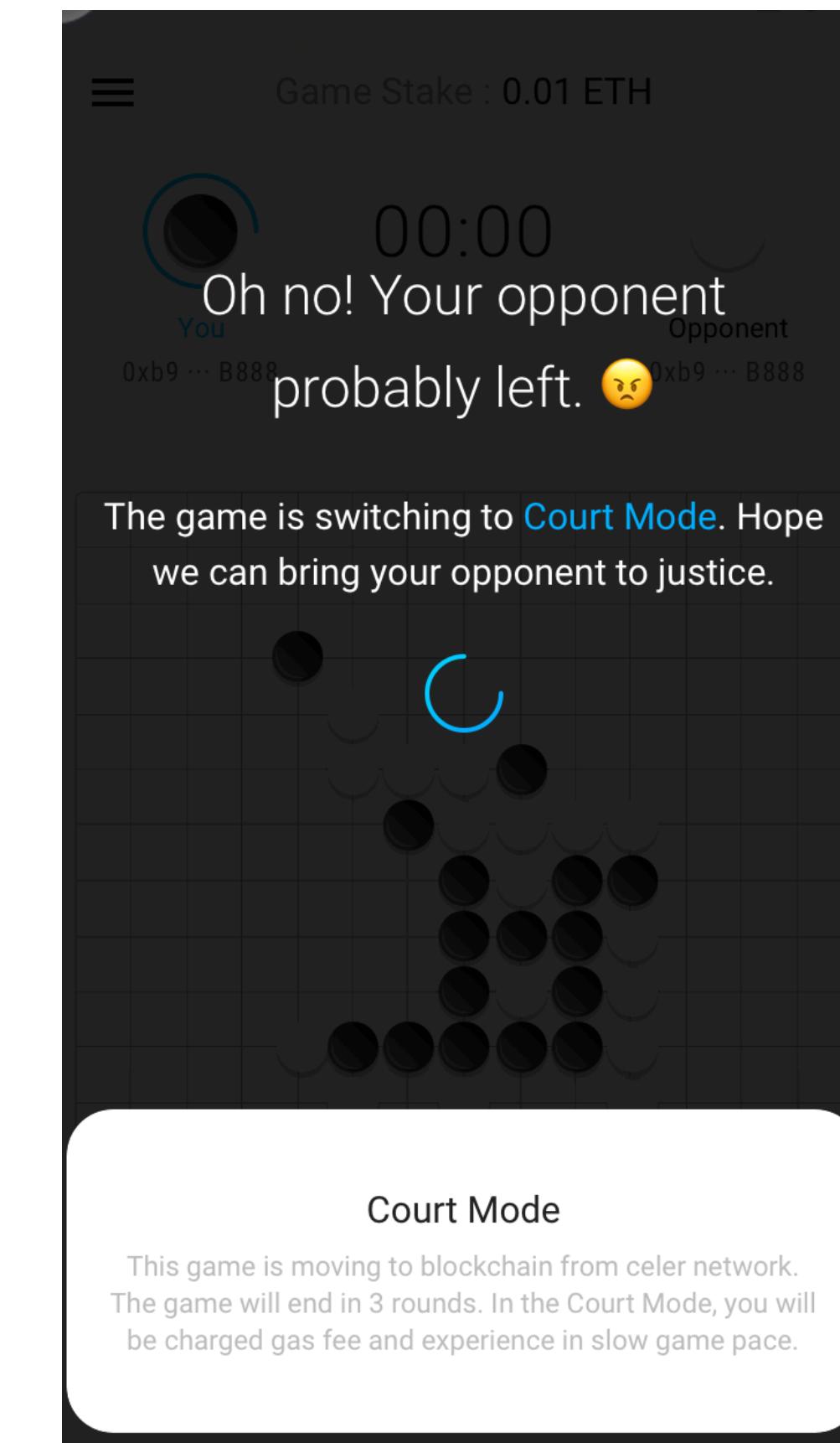
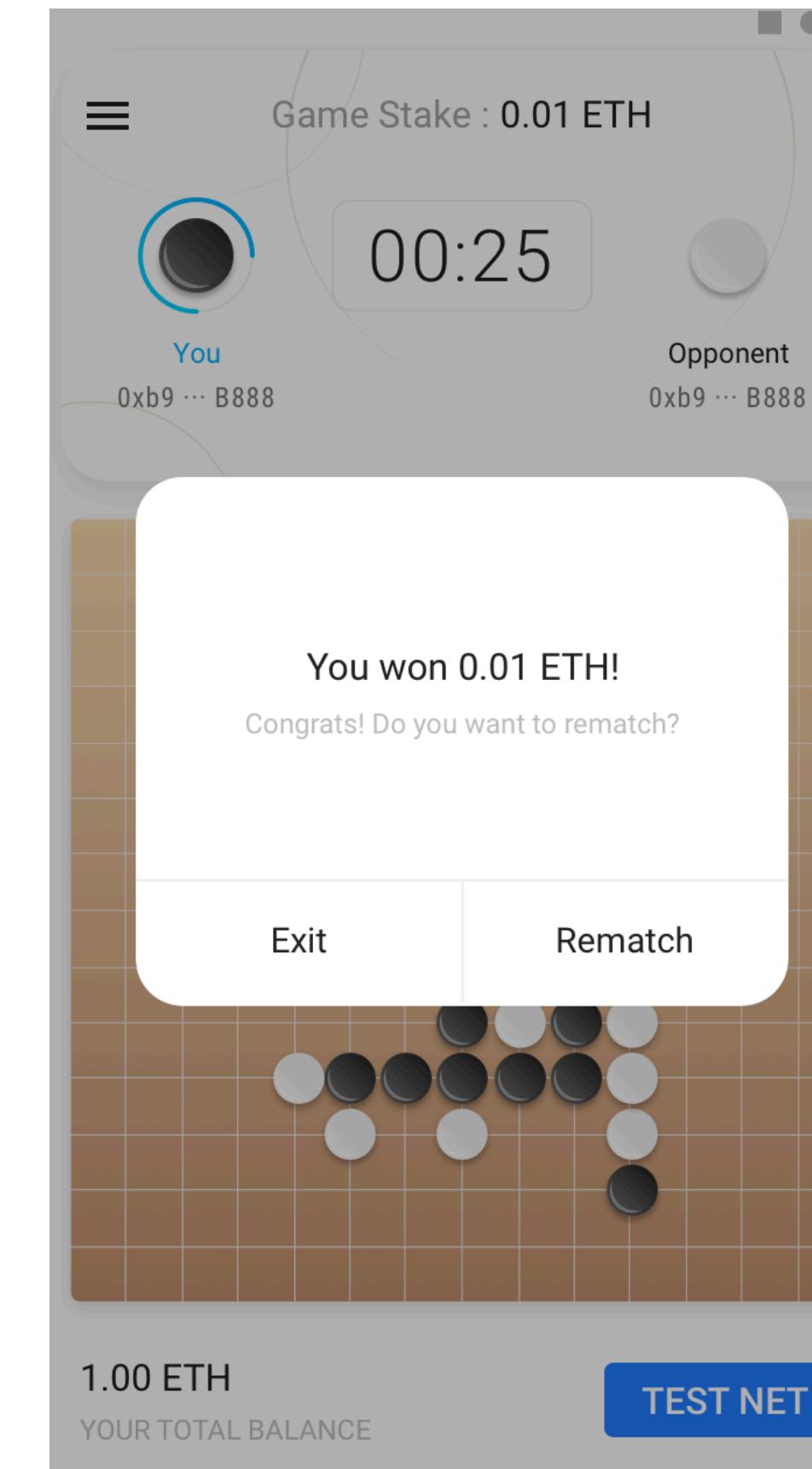
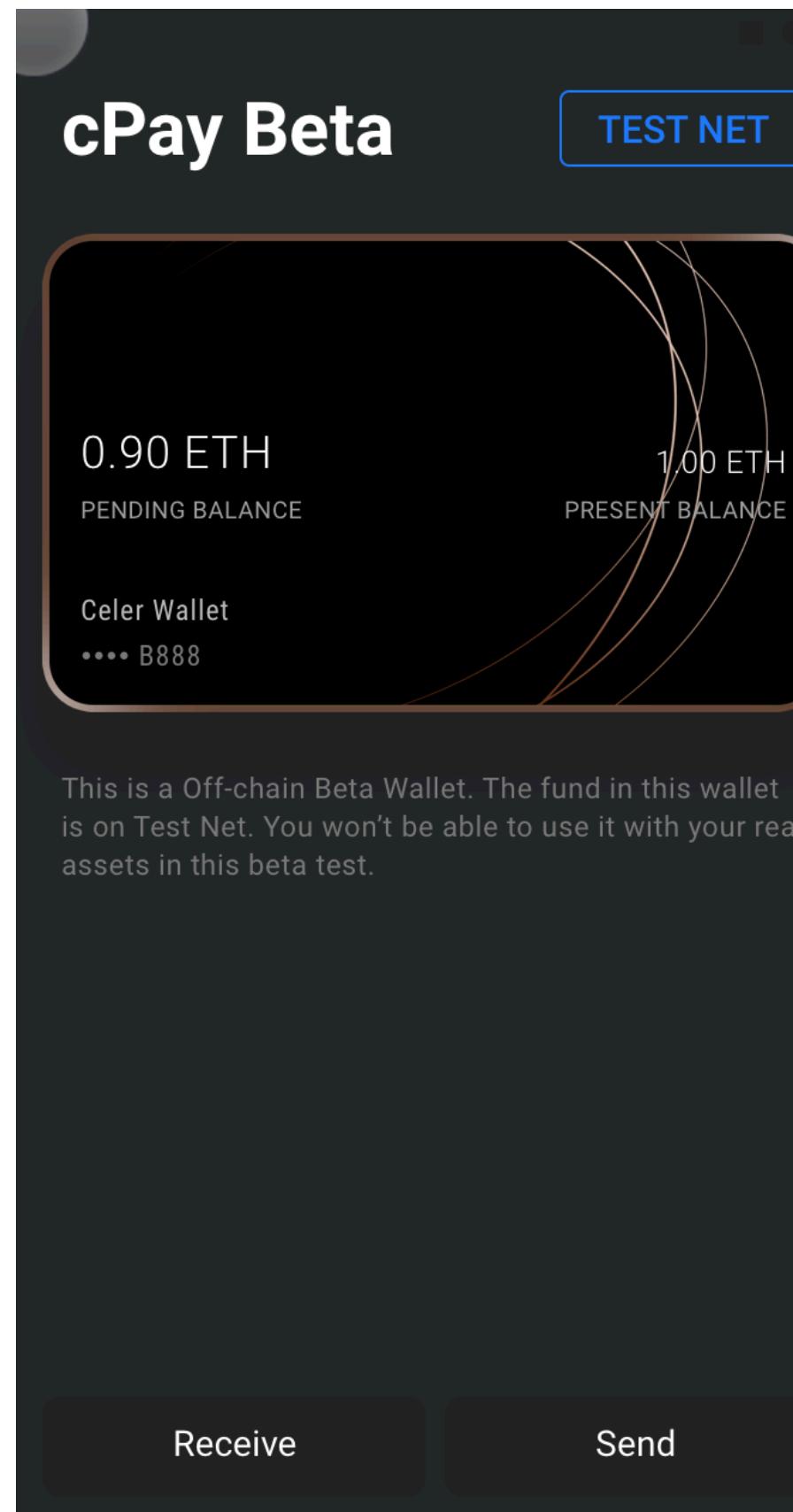
cOS

- Handling all of the complexity for off-chain scaling protocol
- Bridging on-chain and off-chain byte codes

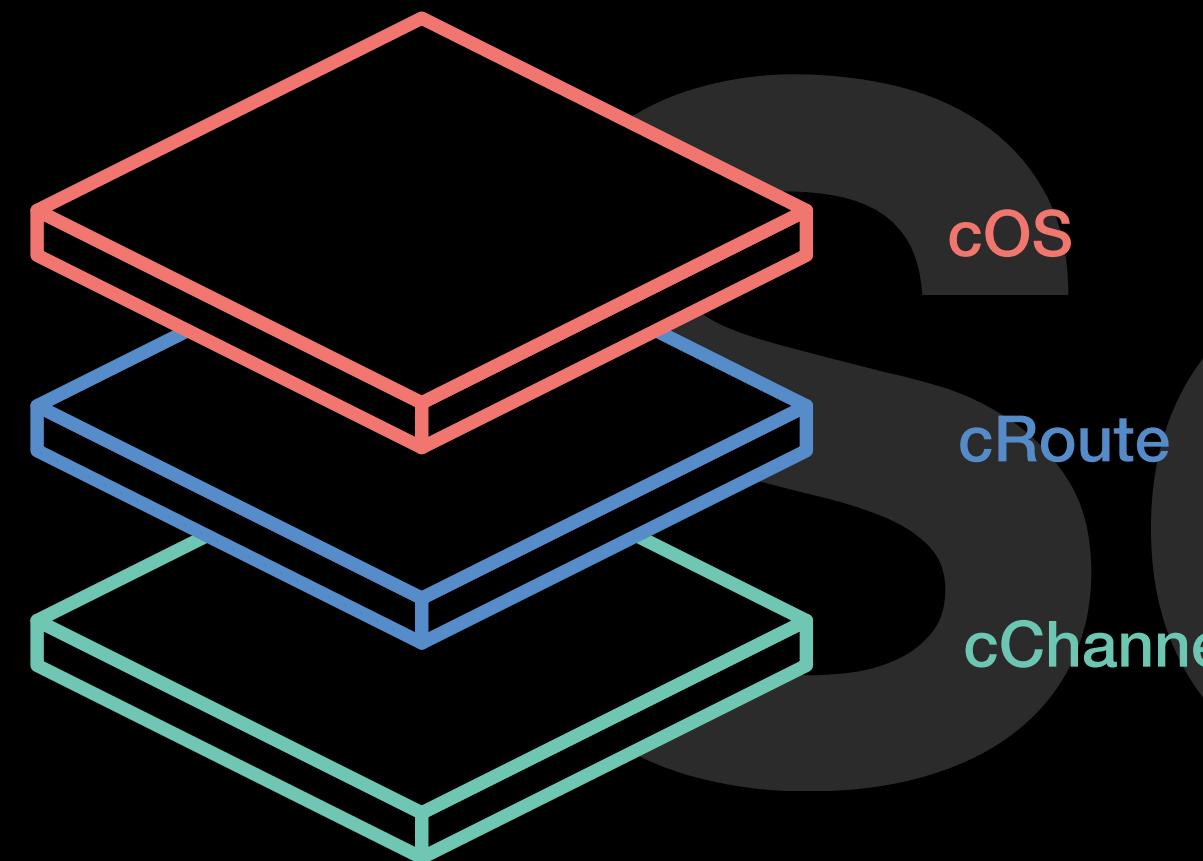


cOS

- Easy-to-use user runtime



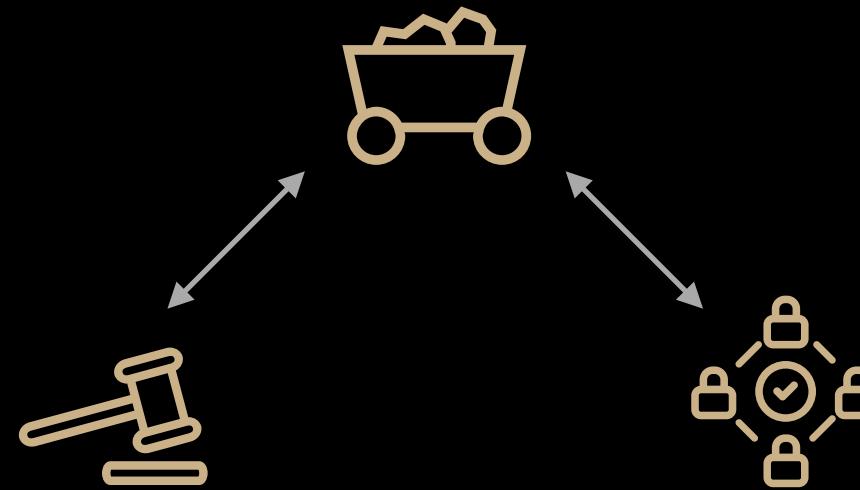
cStack



- How to support generic operations with minimal on-chain footprint?
- How to route value transfers efficiently in off-chain networks?
- How to bring mass adoption to off-chain dApps?

cEconomy

Proof of Liquidity
Commitment Mining



Liquidity Backing
Auction

State Guardian
Network

- How to make off-chain states always available for on-chain disputes?
- How to solve the state connectivity problem?
- How to obtain enough liquidity to run an off-chain service?

cEconomy



Liquidity Backing Auction

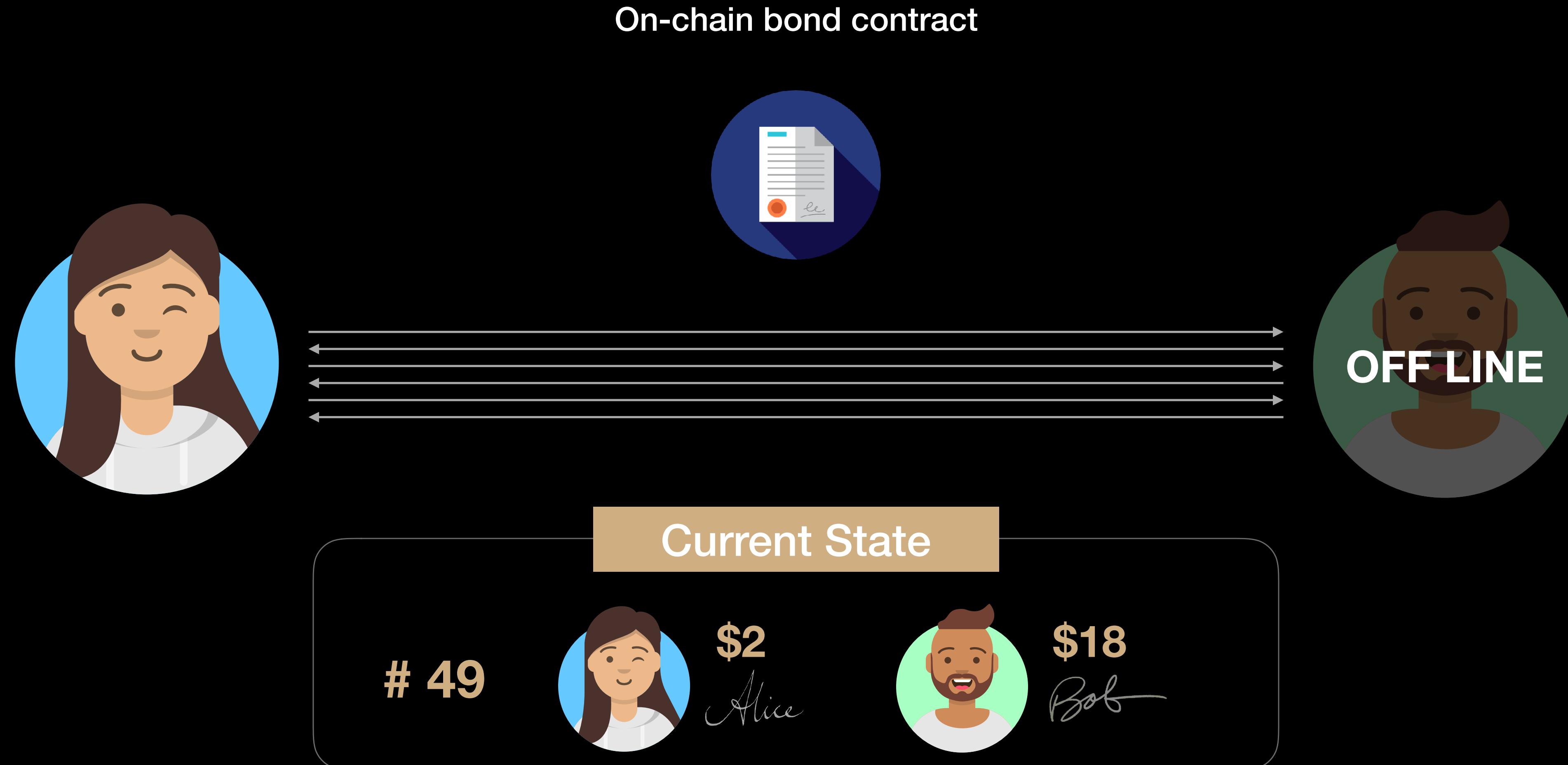
Proof of Liquidity Commitment Mining

State Guardian Network

1. State Availability Challenge

How to make off-chain states always available for on-chain disputes?

State Availability Challenge



Celer State Guardian Network

A Special Side Chain

Guardian #1



100 Celer Tokens

Guardian #2



100 Celer Tokens

Guardian #3



100 Celer Tokens

Guardian #4



100 Celer Tokens

Guardian #N

....



100 Celer Tokens



State Proof

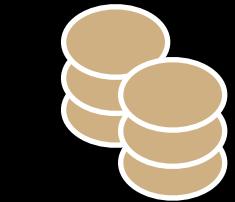
A State Proof can be anything, such as
game state, signed agreement, auction
ACK ...

Guardian #1



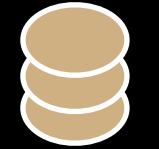
100 Celer Tokens

Guardian #2



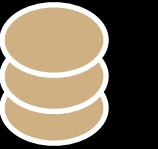
100 Celer Tokens

Guardian #3



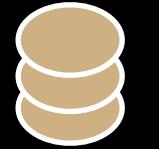
100 Celer Tokens

Guardian #4



100 Celer Tokens

Guardian #5



100 Celer Tokens

Guardian #6



100 Celer Tokens



Other State Proof
Pay \$0.5 for an hr

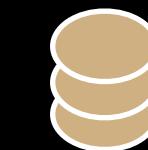


State Proof
Pay \$1 for an hr



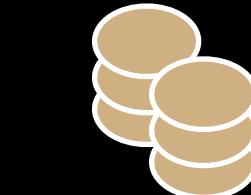
Other State Proof
Pay \$0.5 for an hr

#1



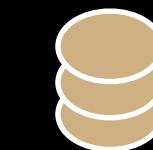
100 Celer Tokens

#2



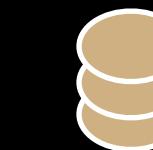
100 Celer Tokens

#3



100 Celer Tokens

#4



100 Celer Tokens

#5



100 Celer Tokens

#6



100 Celer Tokens

State Proof

State Proof

State Proof

Pay \$1 can get 3 guards (6/2) for an hour



Other State Proof
Pay \$0.5 for an hr

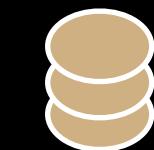


State Proof
Pay \$1 for an hr



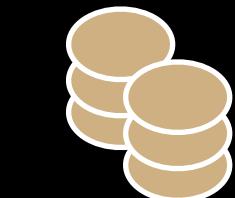
Other State Proof
Pay \$0.5 for an hr

#1



100 Celer Tokens

#2



200 Celer Tokens

#3



100 Celer Tokens

#4



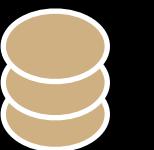
100 Celer Tokens

#5



100 Celer Tokens

#6



100 Celer Tokens

State Proof
State Proof
State Proof

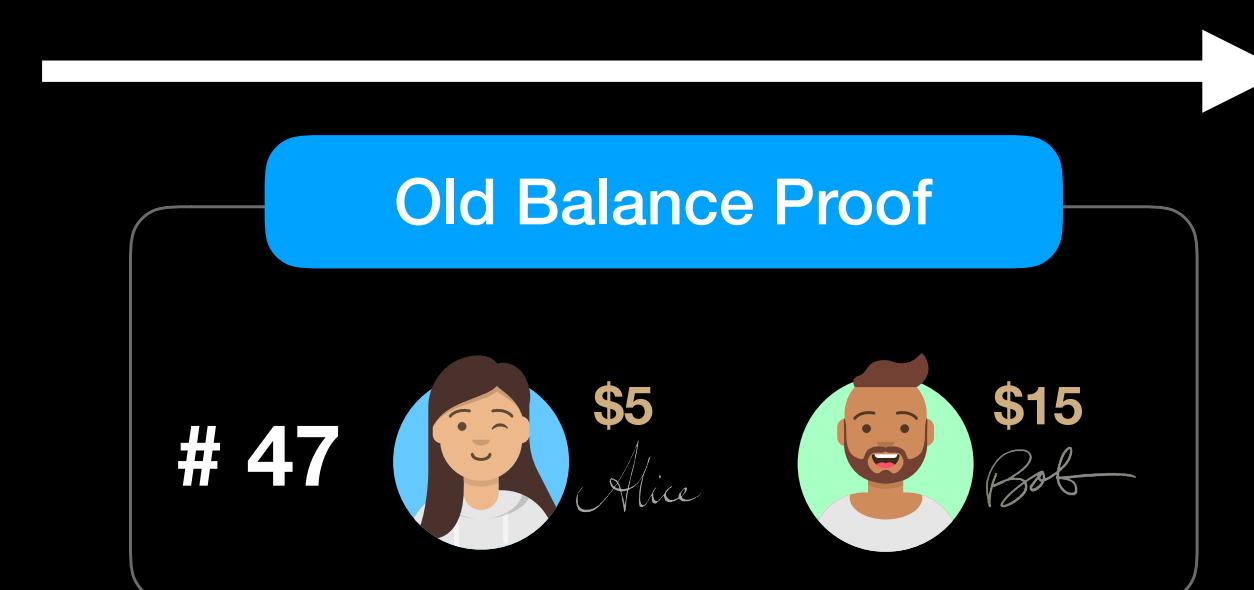
State Proof
State Proof

State Proof
State Proof

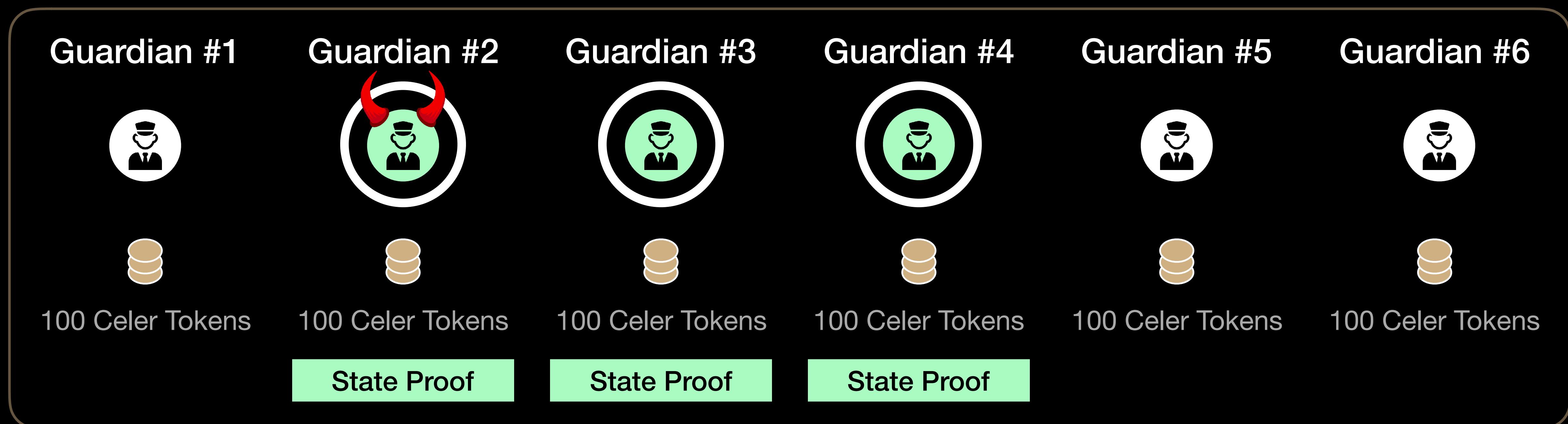
More Celer Tokens = More work = More Money



Malicious

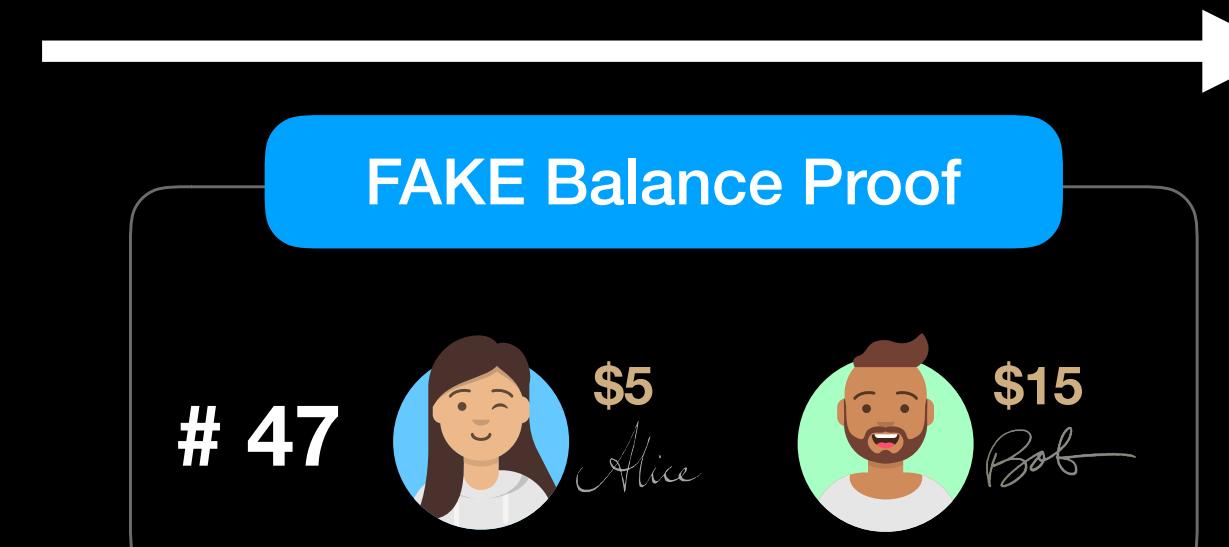


State Proof

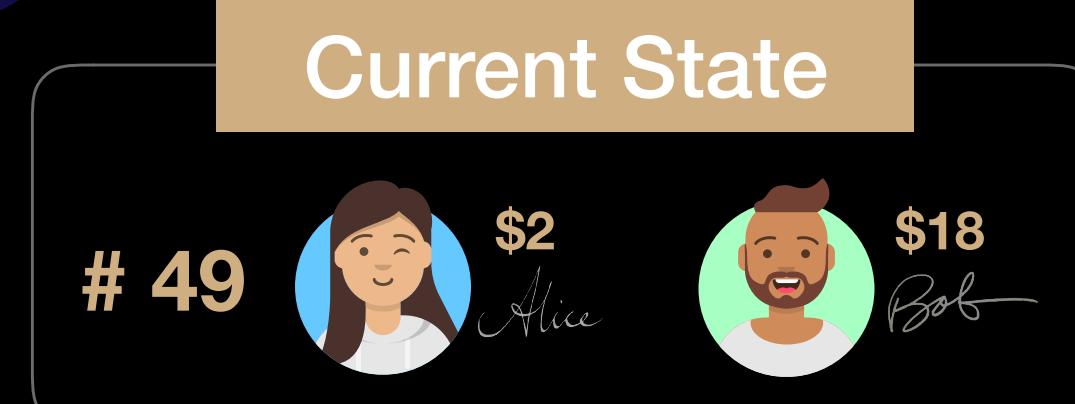




Malicious



Current State



OFF LINE

State Proof

Guardian #1



100 Celer Tokens

Guardian #2



100 Celer Tokens

Guardian #3



100 Celer Tokens

Guardian #4



100 Celer Tokens

Guardian #5



100 Celer Tokens

Guardian #6



100 Celer Tokens

State Proof

State Proof

State Proof

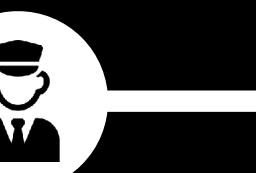
Dispute Timeline



Guardian #2



Guardian #3

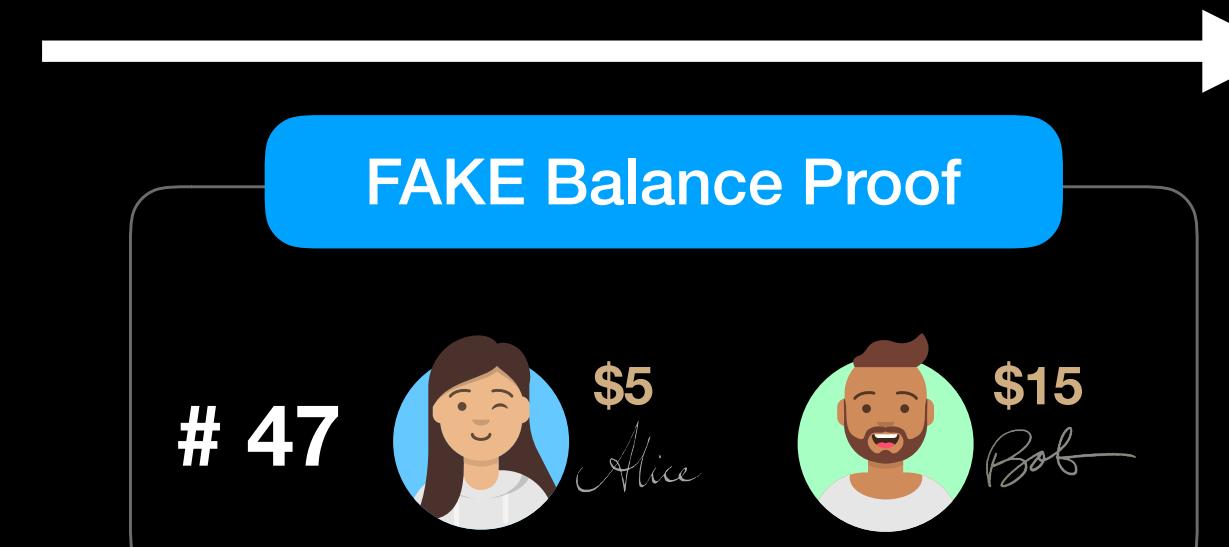


Guardian #4





Malicious



Current State



OFF LINE

State Proof

Guardian #1

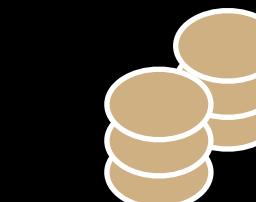


Celer Stake slashed & kicked out of SGN



100 Celer Tokens

Guardian #3



200 Celer Tokens

Guardian #4



100 Celer Tokens

Guardian #5



100 Celer Tokens

Guardian #6



100 Celer Tokens

Dispute Timeline



Guardian #2



Guardian #3

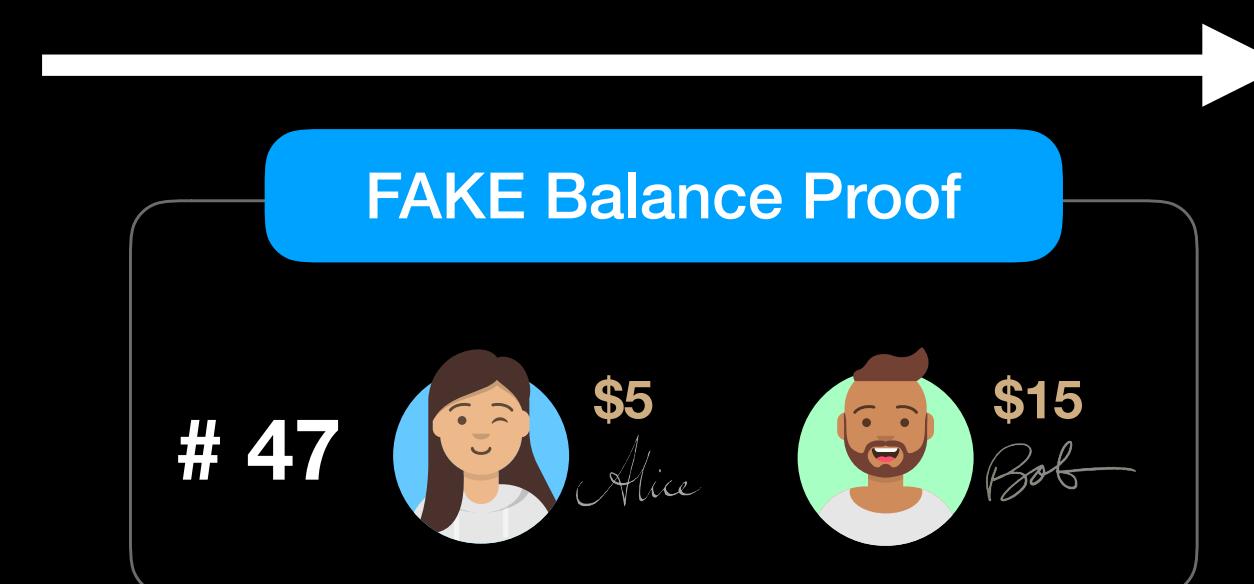


Guardian #4





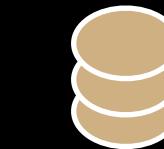
Malicious



OFF LINE

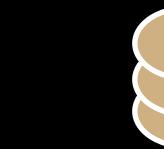


Guardian #1



100 Celer Tokens

Guardian #2



100 Celer Tokens

Guardian #3



100 Celer Tokens

Guardian #4



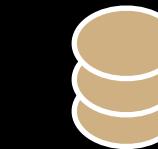
100 Celer Tokens

Guardian #5



100 Celer Tokens

Guardian #6



100 Celer Tokens

State Proof

State Proof

State Proof

Dispute Timeline



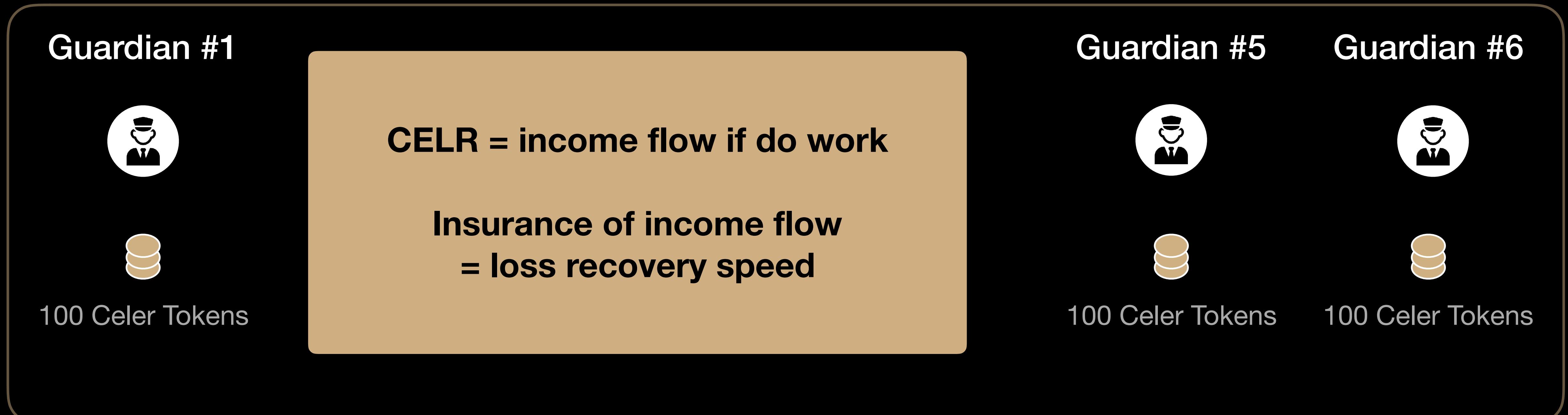
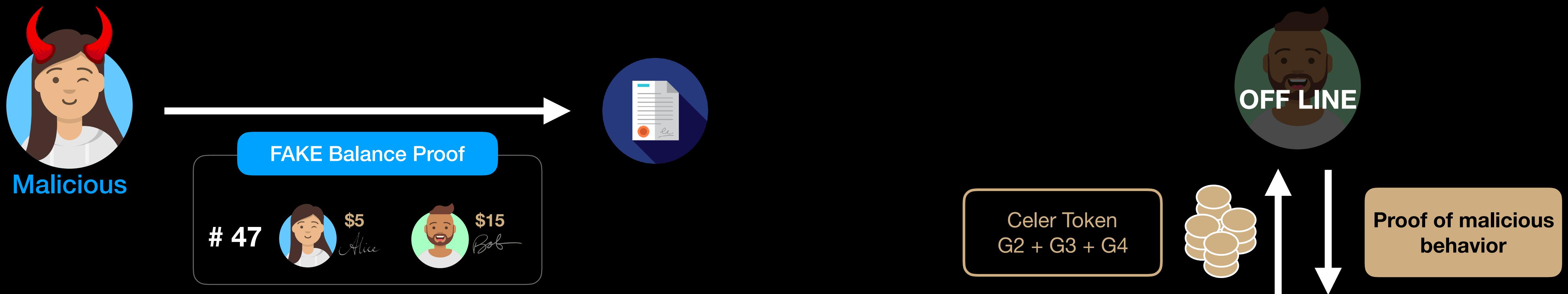
Guardian #2



Guardian #3



Guardian #4



2. Connectivity Challenge

How to solve the state connectivity problem?

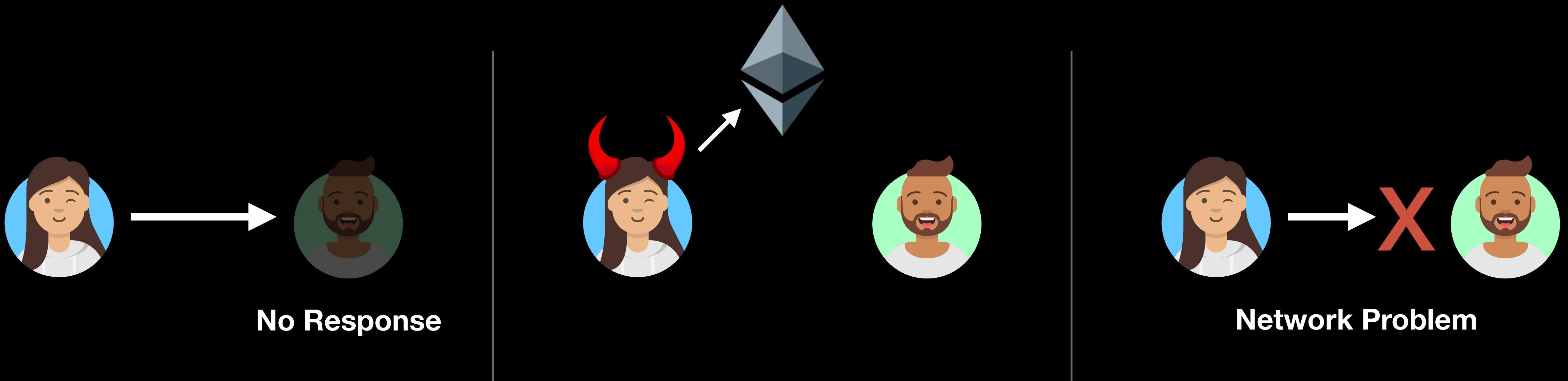
State Availability Challenge



State Availability Challenge



But blockchain cannot differentiate



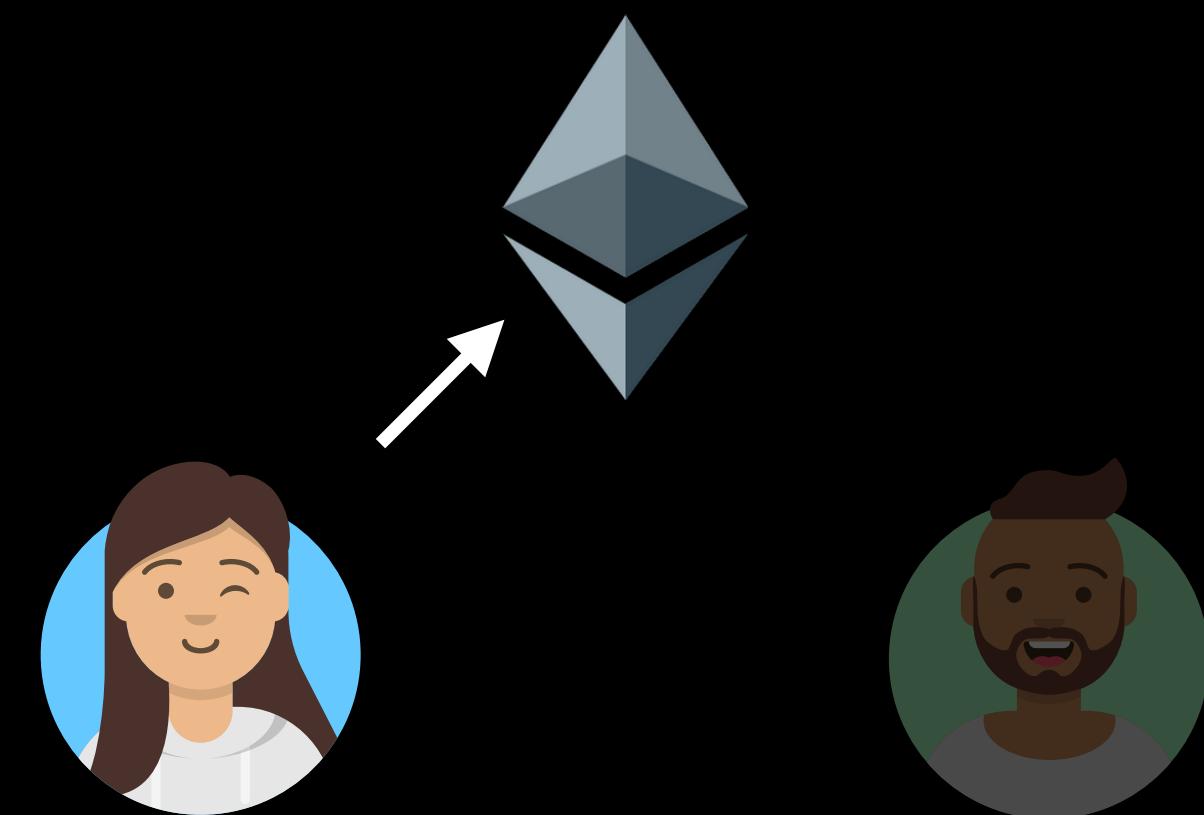
Connectivity Challenge

We need a “fall-back data exchange fabric” that

- is reasonably available
- has attribution of availability time
- is cost effective
- does not require large amount of resources from end users

Blockchain as a data availability service

- ✓ reasonably available
- ✓ attribution of availability time
- ✗ on-chain state storage - expensive
- ✗ require $O(n)$ monitoring*

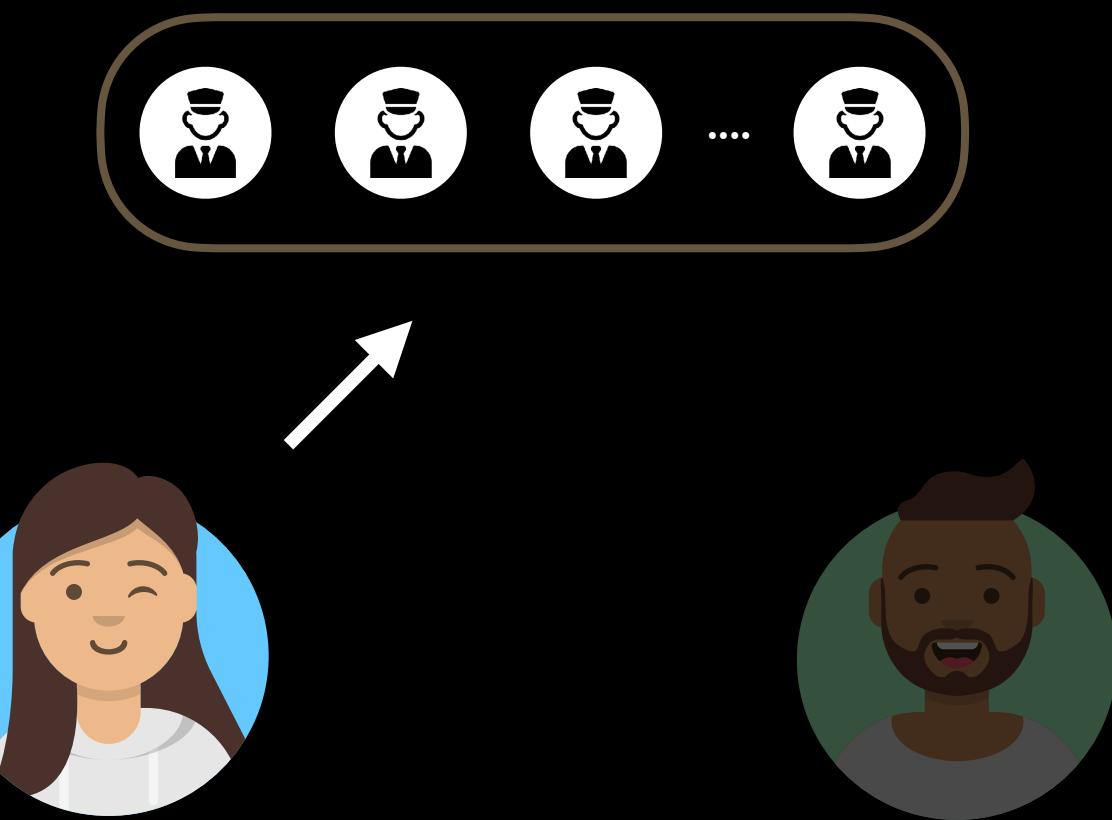


2. Connectivity Solution

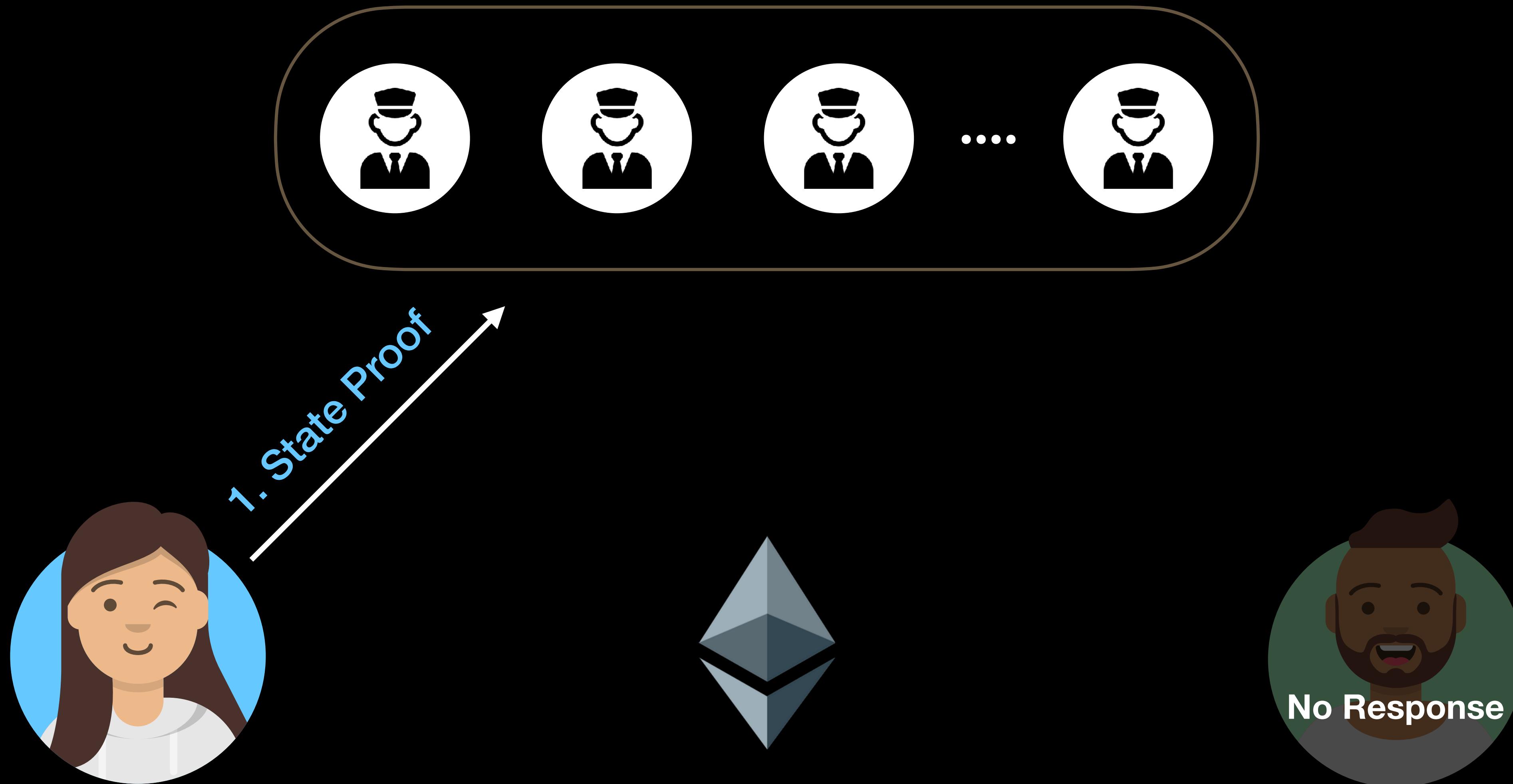
State Guardian Network (SGN) as a data availability service

SGN as a data availability service

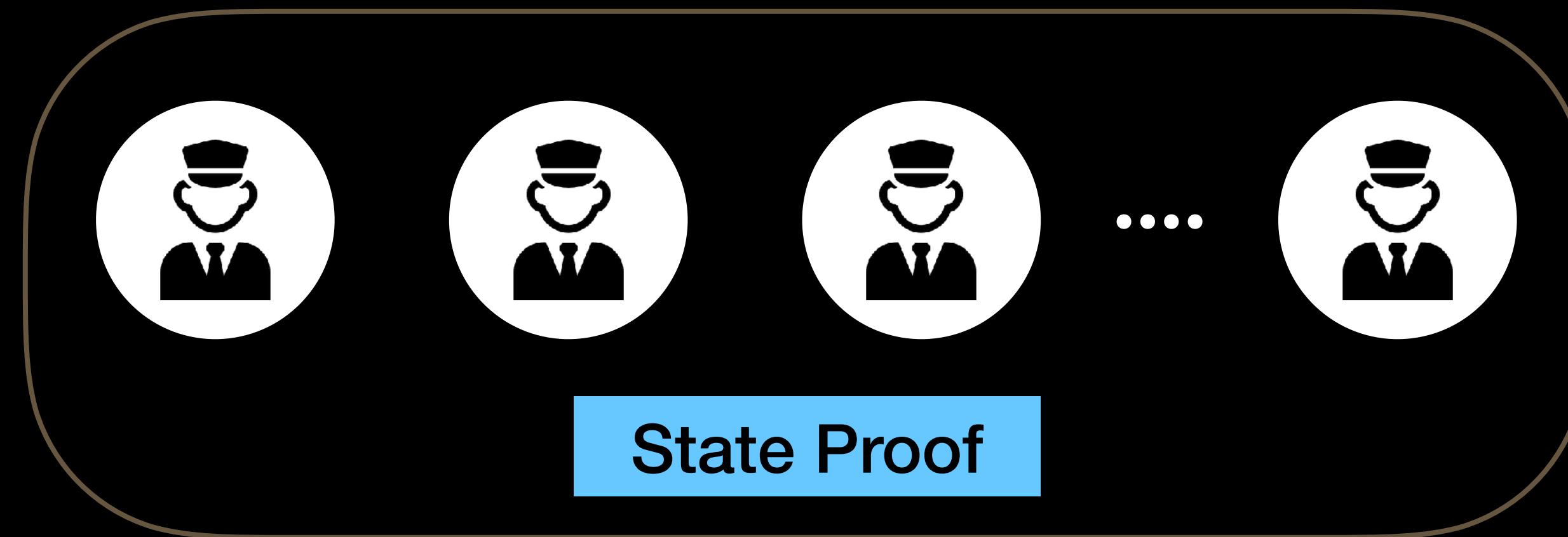
- ✓ reasonably available
- ✓ attribution of availability time
- ✓ on-chain state storage - expensive
- ✓ require $O(n)$ monitoring*



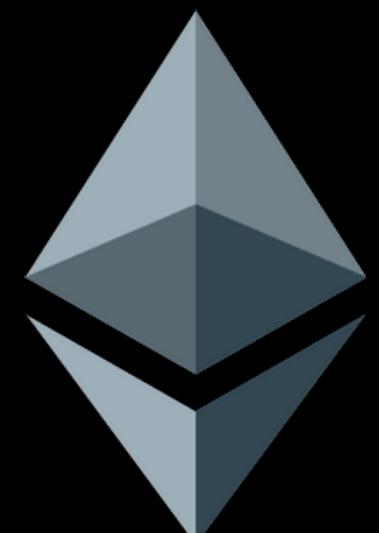
SGN as a data availability service



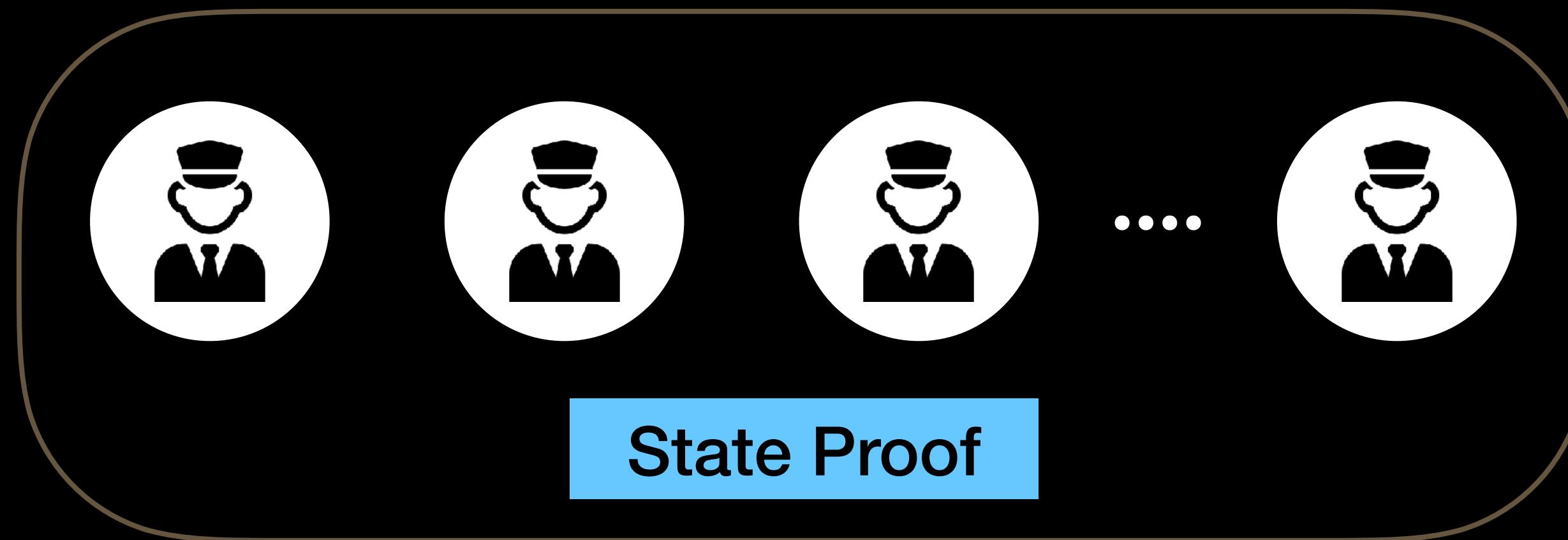
SGN as a data availability service



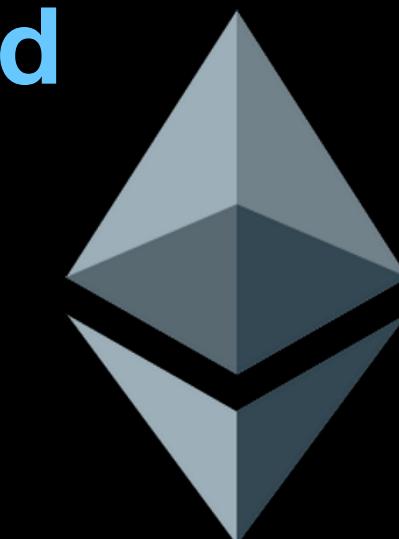
2. Waiting for Timeout



SGN as a data availability service



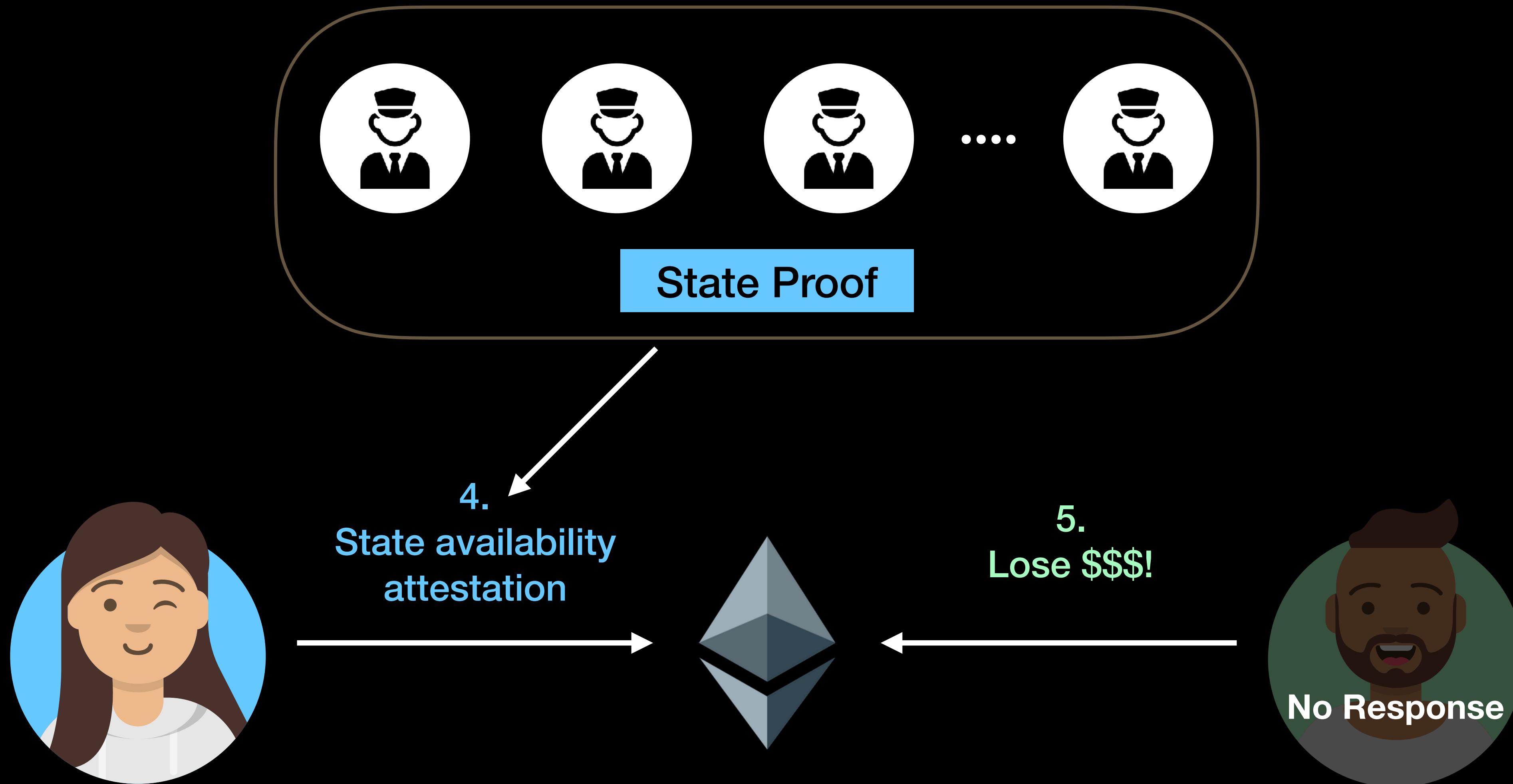
3. “1-bit” claim + challenge with \$ bond



3. challenge response with \$ bond



SGN as a data availability service





State Guardian Network

- Compact sidechain
- Decentralized trust
- Collusion resistance
- Simple unified interface
- No additional liquidity lockup
- Flexible economic dynamics
- Efficient service pricing
- A solution for data connectivity

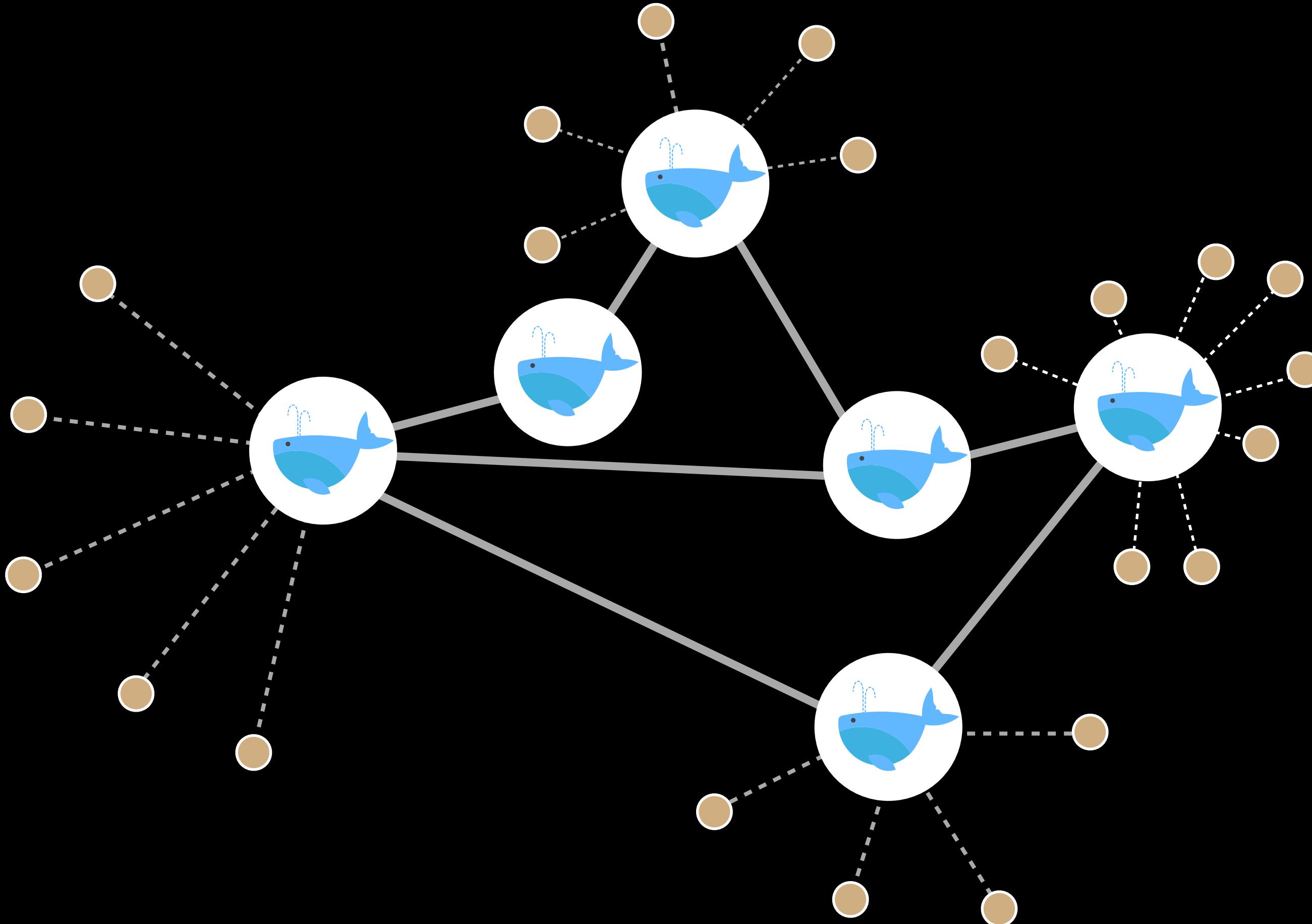
Stake CELR into SGN to guard user states and earn service fees

3. Network Liquidity Challenge

How to obtain enough liquidity to run an off-chain service?

Centralization Risk

- Poor service
- Expensive
- No neutrality
- Closed system



3. Network Liquidity Solution

Proof of Liquidity Commitment

+

Liquidity Backing Auction

Proof of Liquidity Commitment



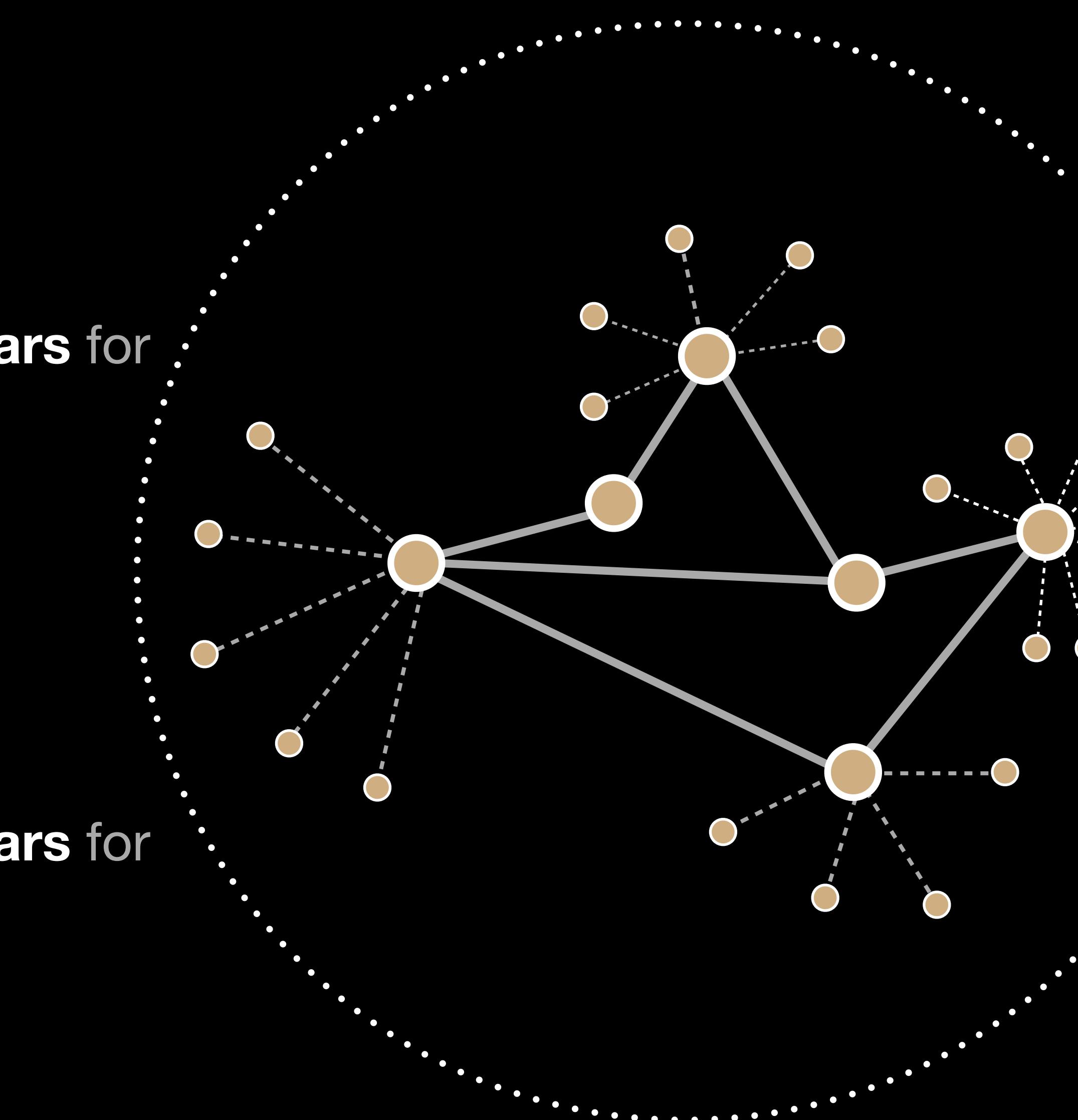
Liquidity Baker

I am willing to lock up **10 ETH for 3 years** for nothing but backing of Celer Networks



Liquidity Baker

I am willing to lock up **30 ETH for 2 years** for nothing but backing of Celer Networks



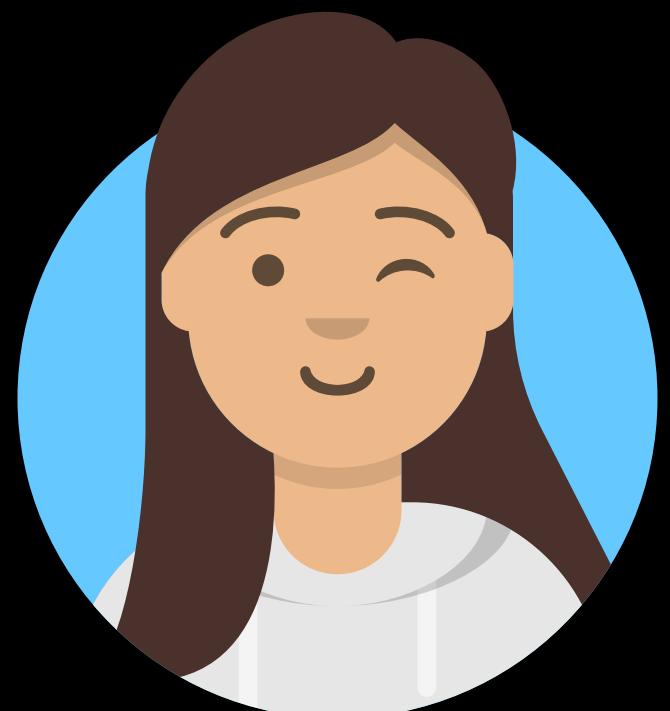
Proof of Liquidity Commitment



$10 \text{ ETH} \times 3 \text{ years} = 30 \text{ Virtual Mining Power}$



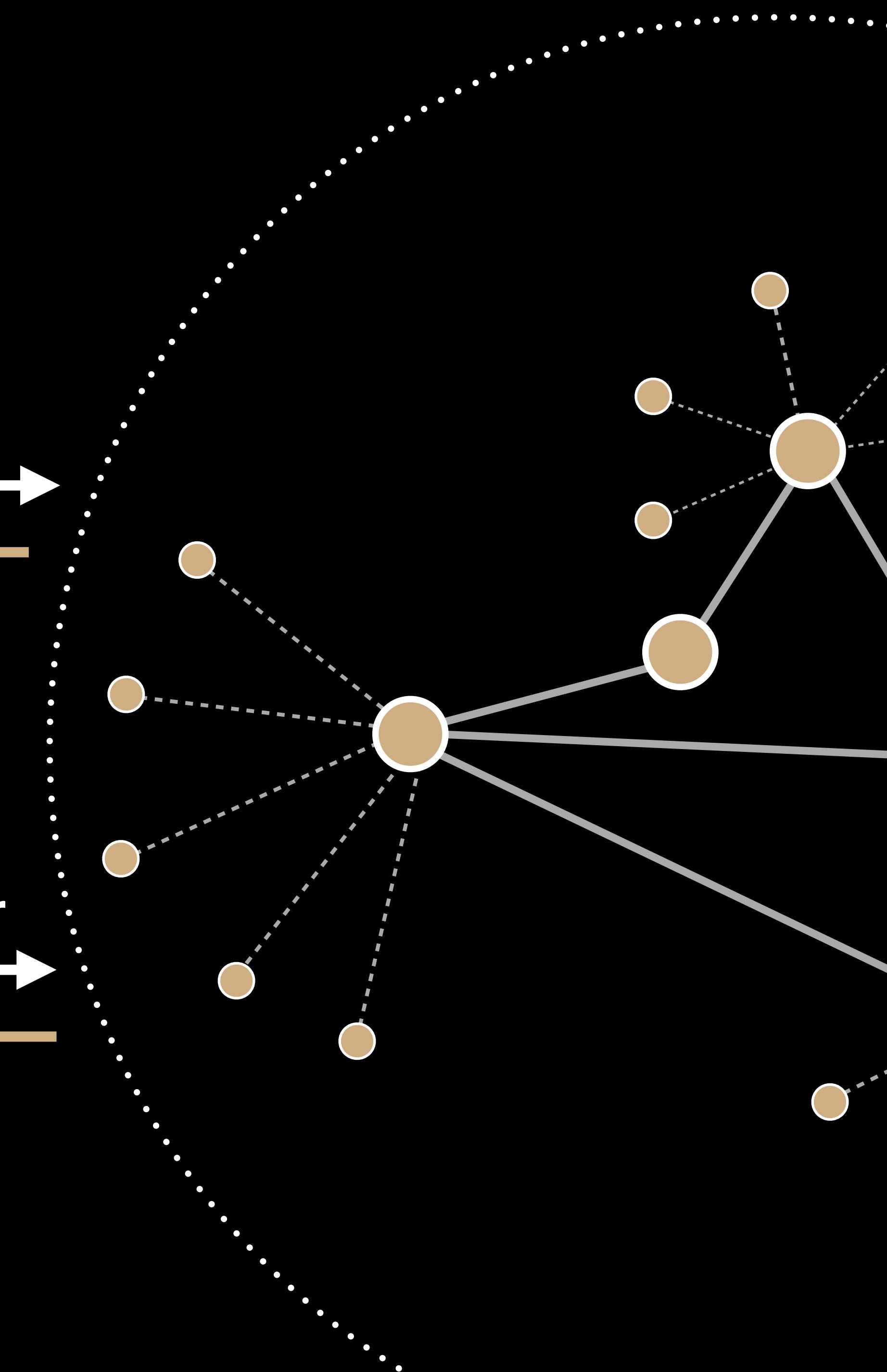
$1/3 \times \text{Newly generated CELR}$



$30 \text{ ETH} \times 2 \text{ years} = 60 \text{ Virtual Mining Power}$

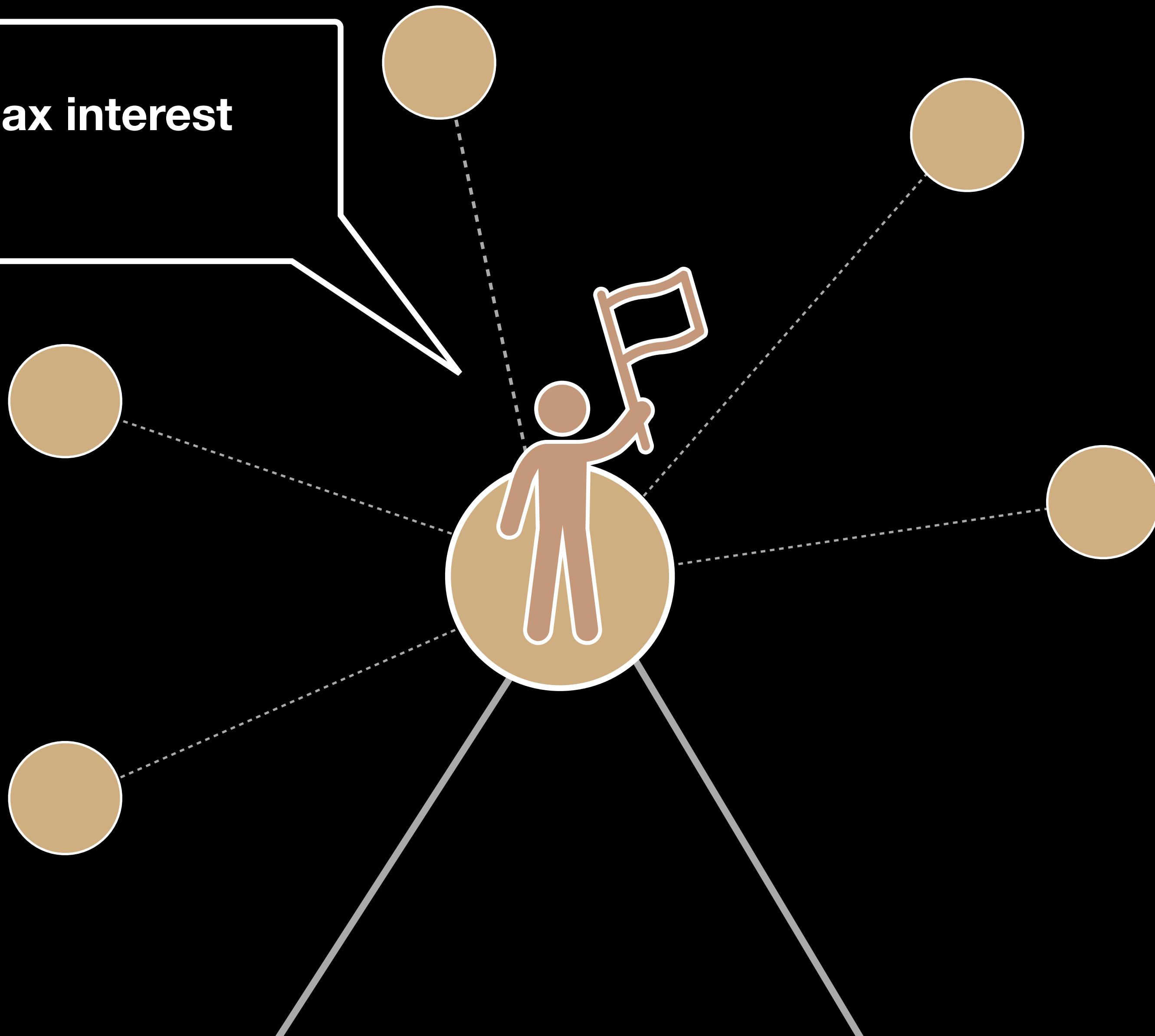


$2/3 \times \text{Newly generated CELR}$

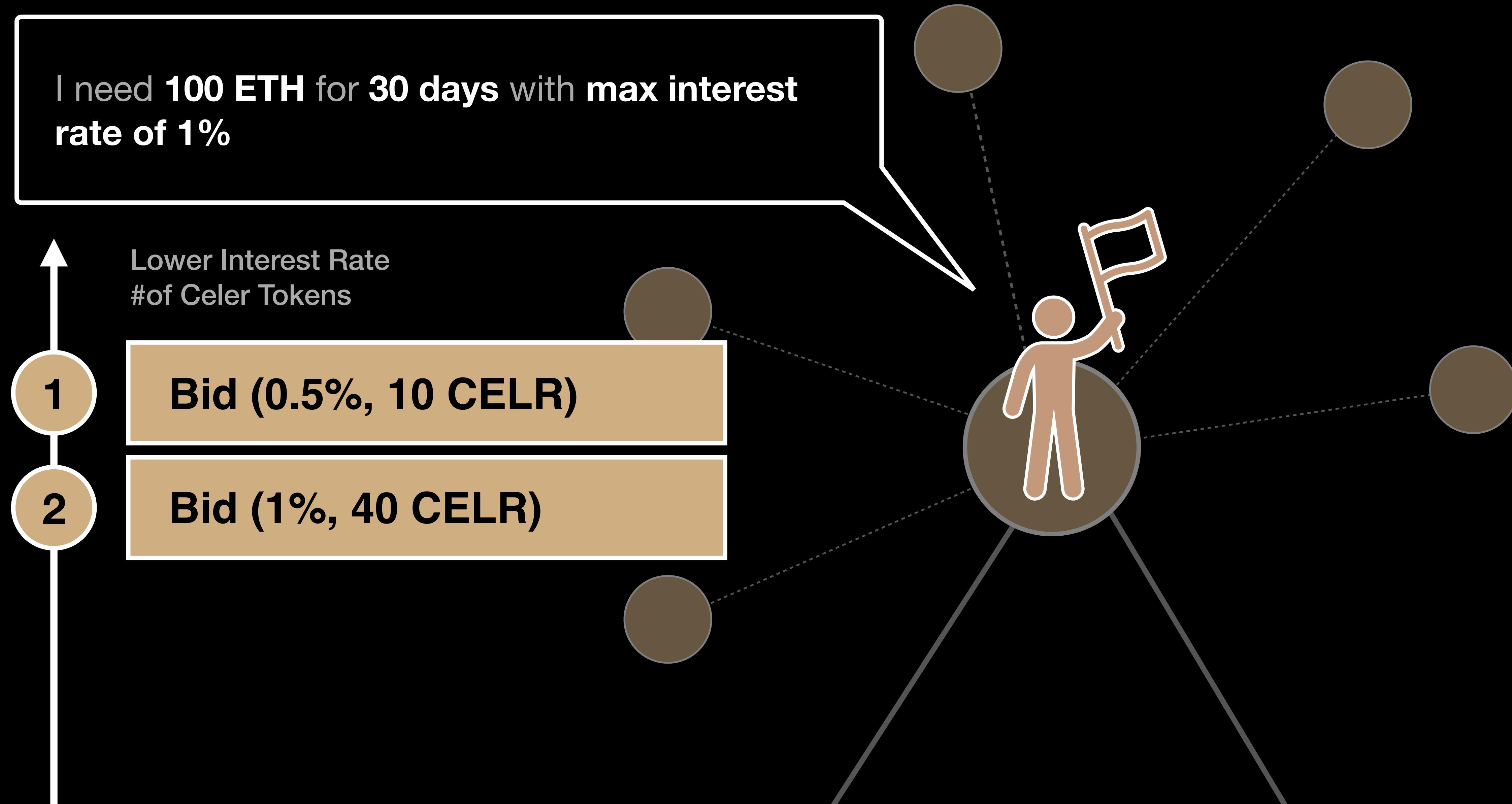


Liquidity Backing Auction

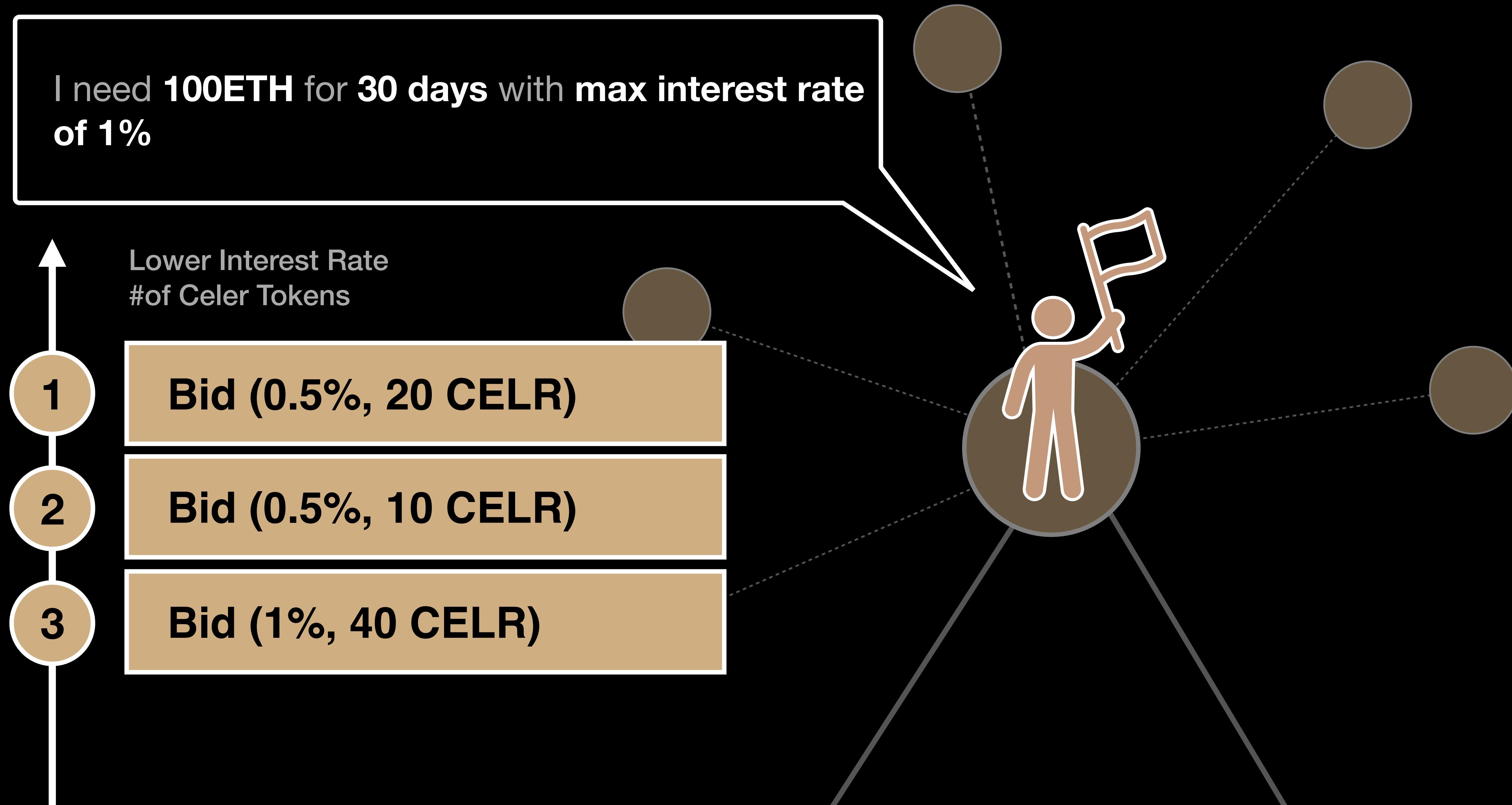
I need **100 ETH** for **30 days** with **max interest rate of 1%**



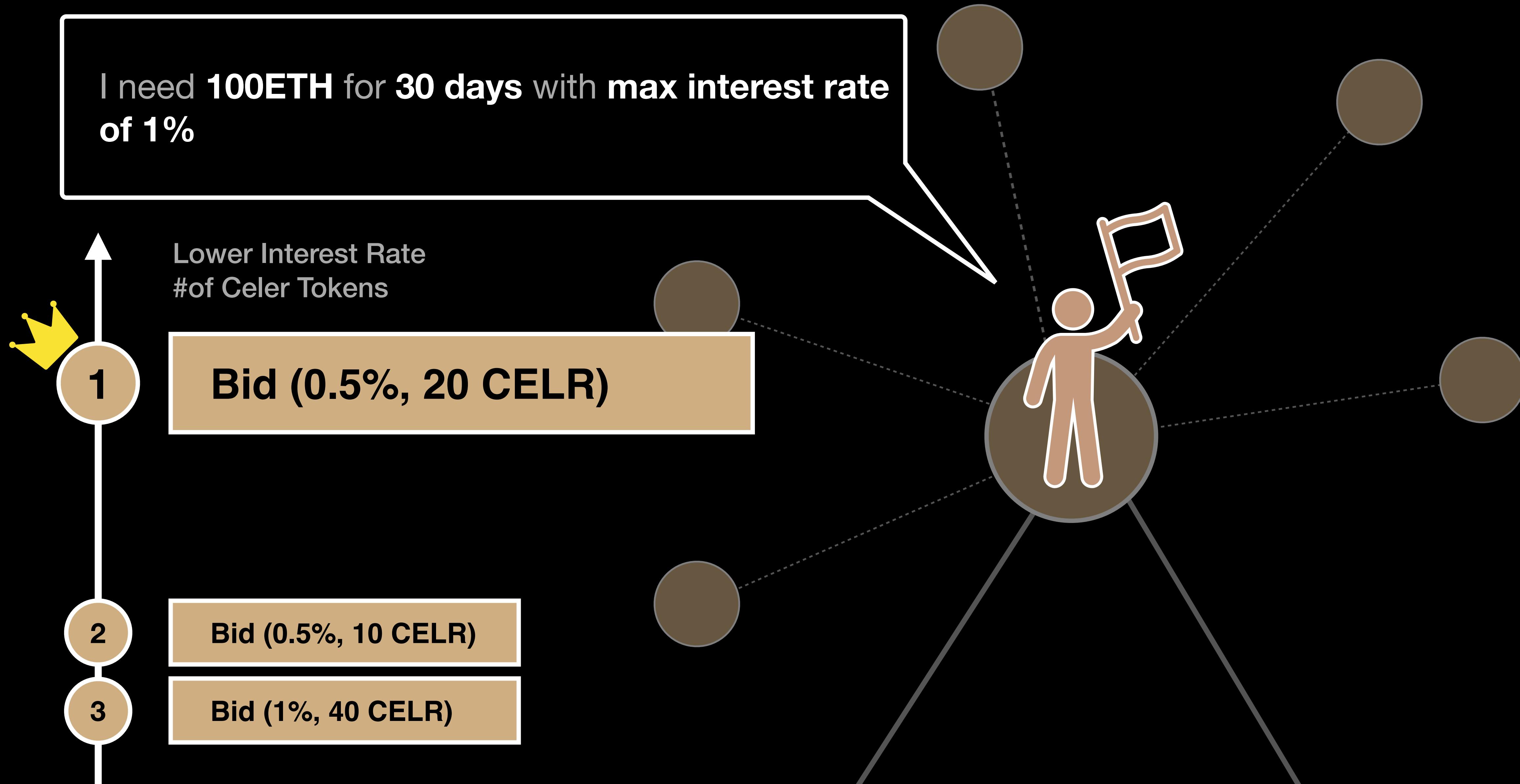
Liquidity Backing Auction



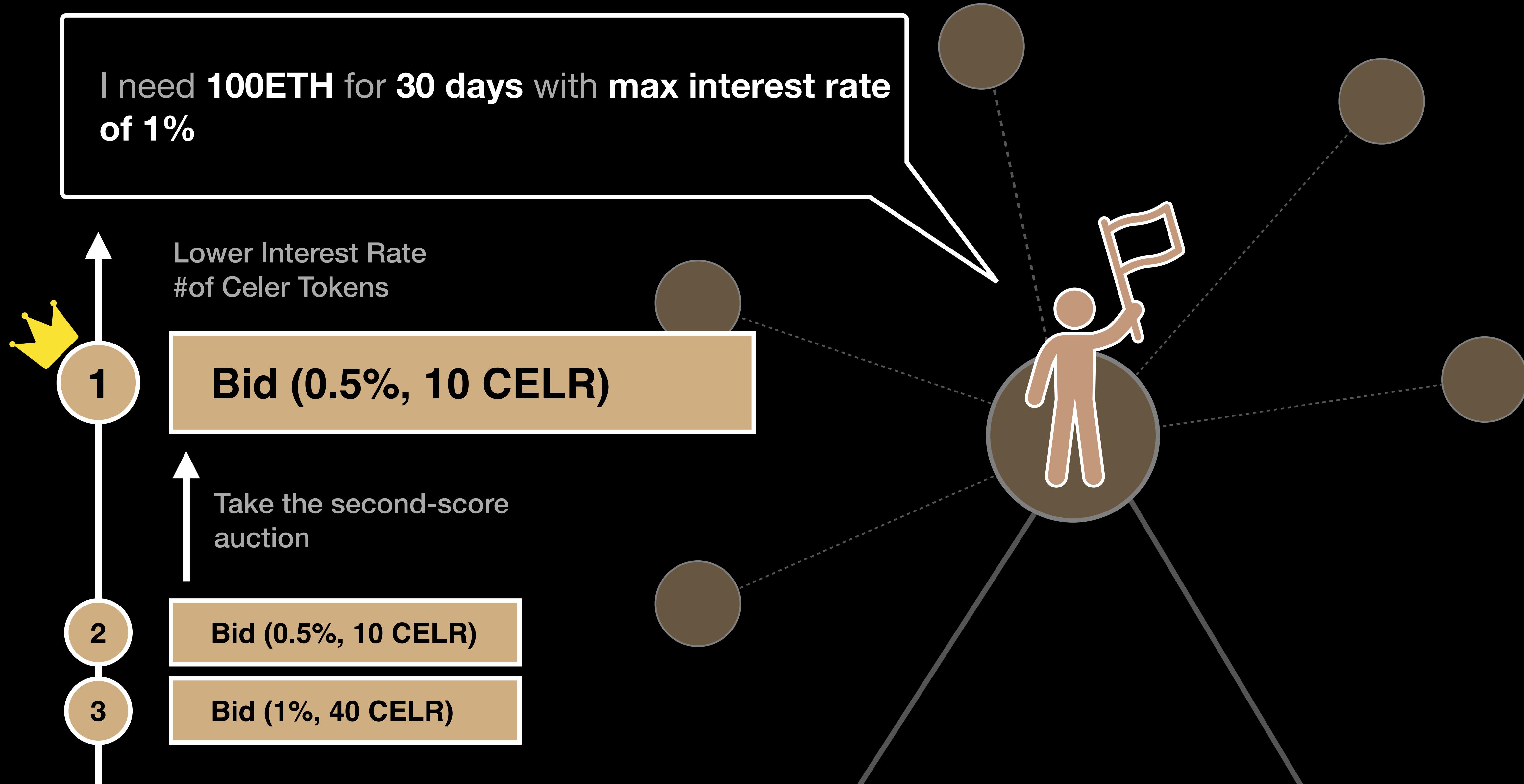
Liquidity Backing Auction



Liquidity Backing Auction



Liquidity Backing Auction



cEconomy completes the off-chain ecosystem



Liquidity Backing Auction

Proof of Liquidity Commitment Mining

State Guardian Network



- Mine CELR by locking idle liquidity in the off-chain platform
- Incentivize an abundant and stable liquidity pool

Proof of Liquidity Commitment Mining



Liquidity Backing Auction

- off-chain service providers crowdsource liquidity
- Reward committed liquidity backers
- Select backers based on interest rate and CELR staking
- Ensure 100% safety for end users

cStack

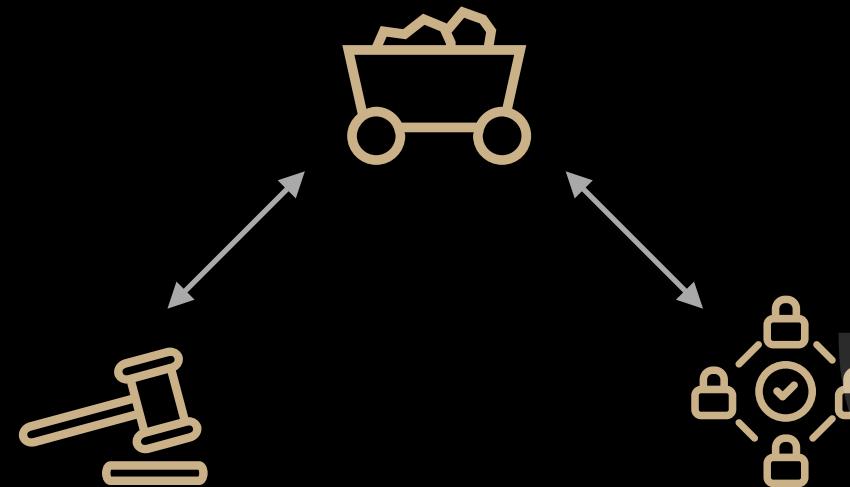


cOS
cRoute
cChannel

- How to support generic operations with minimal on-chain footprint?
- How to route value transfers efficiently in off-chain networks?
- How to bring mass adoption to off-chain dApps?

cEconomy

Proof of Liquidity
Commitment Mining



Liquidity Backing
Auction

State Guardian
Network

- How to make off-chain states always available for on-chain disputes?
- How to solve the state connectivity problem?
- How to obtain enough liquidity to run an off-chain service?

Wechat



CelerNetwork技术社群



Telegram



t.me/celernetwork



cWallet Product Page



get.celer.app



Follow Us

