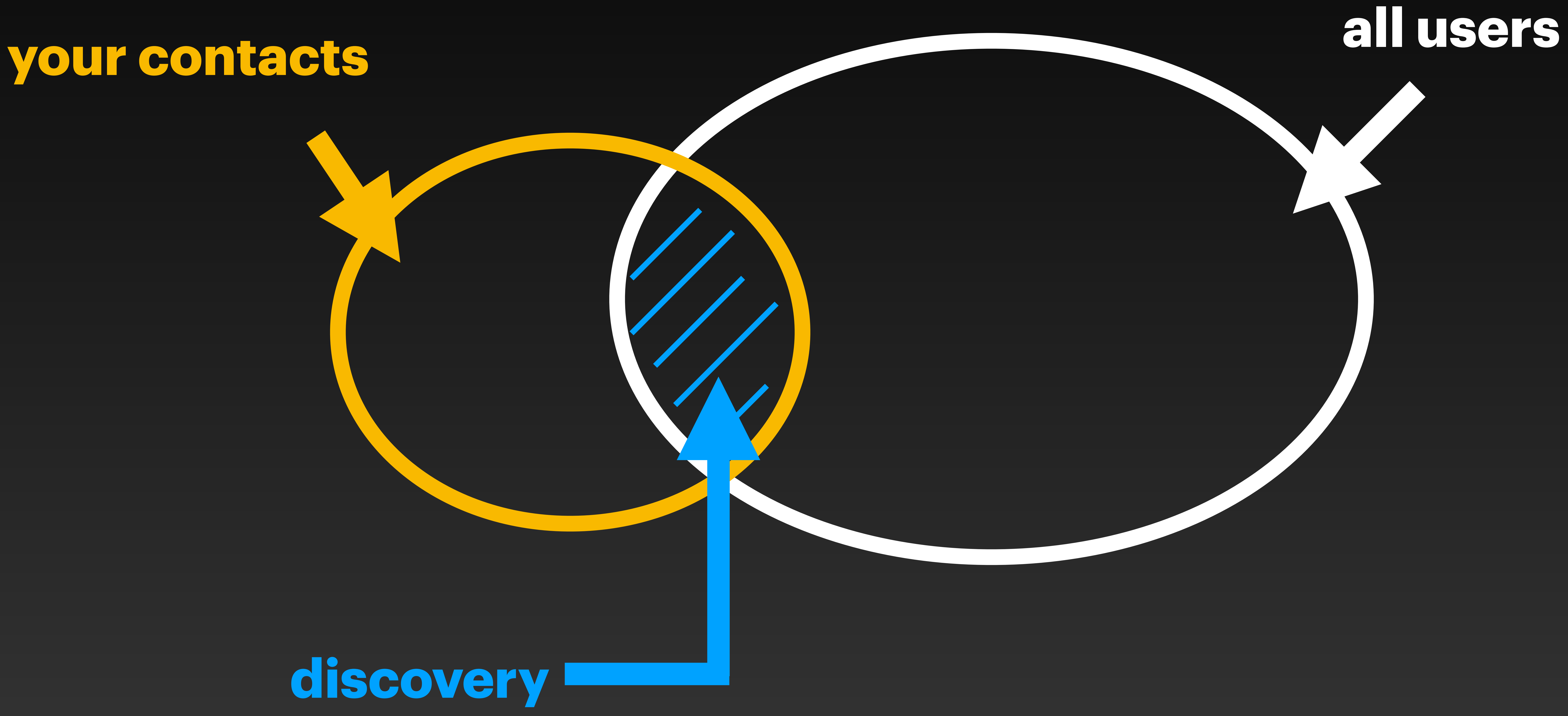
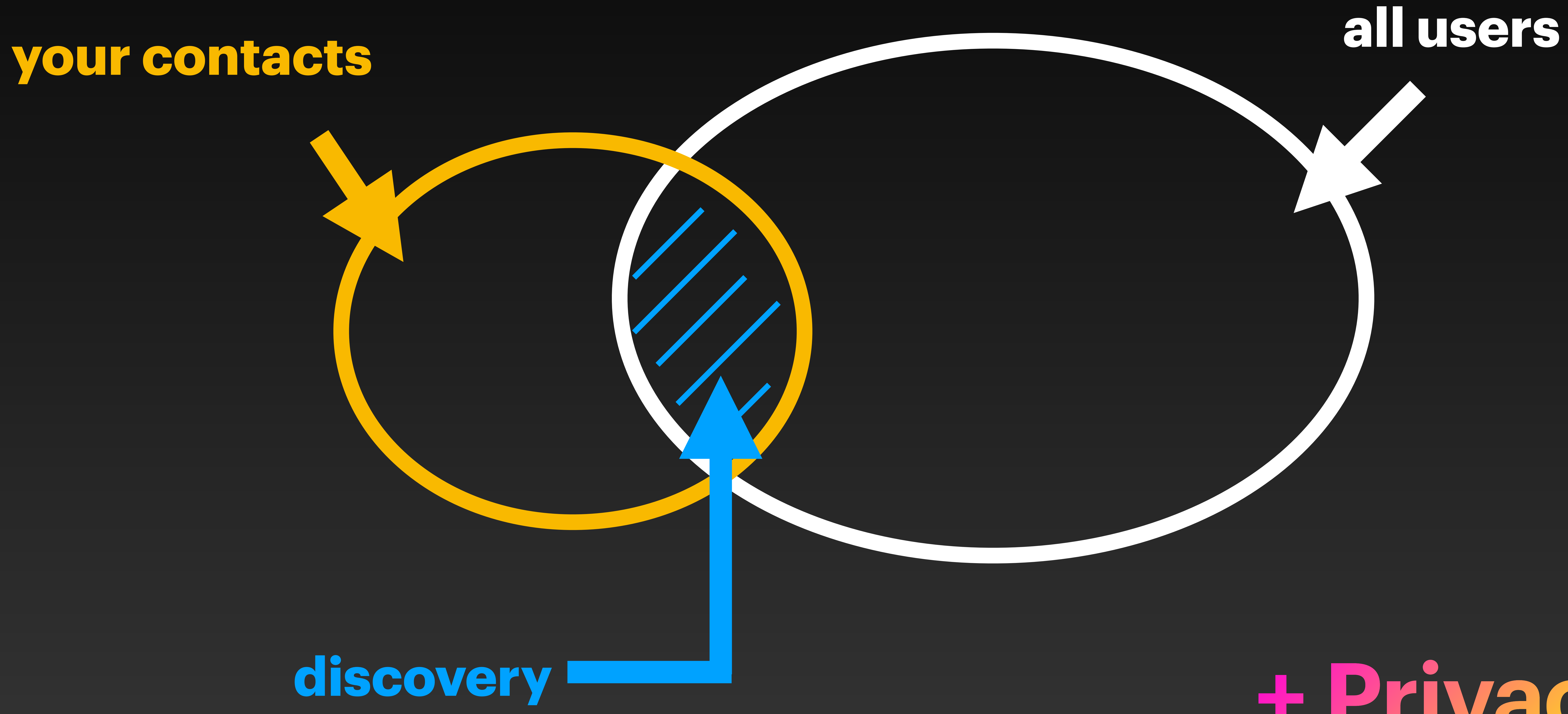
Arke

Scalable and Byzantine Fault Tolerant Privacy-Preserving Contact Discovery

Alberto Sonnino


your contacts
all users
discovery



+ Privacy

Web2 Needs

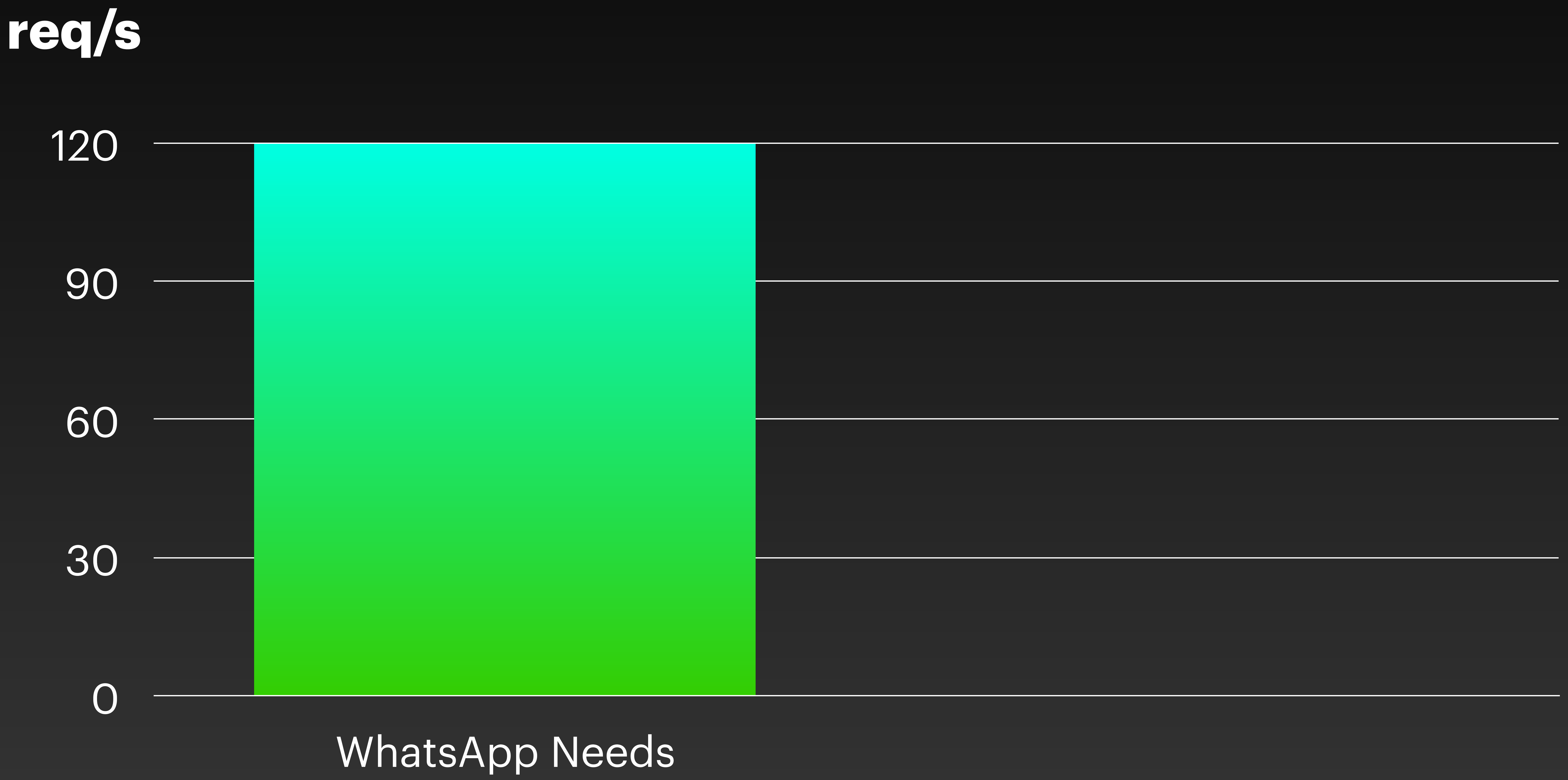
- 10 Million requests / day

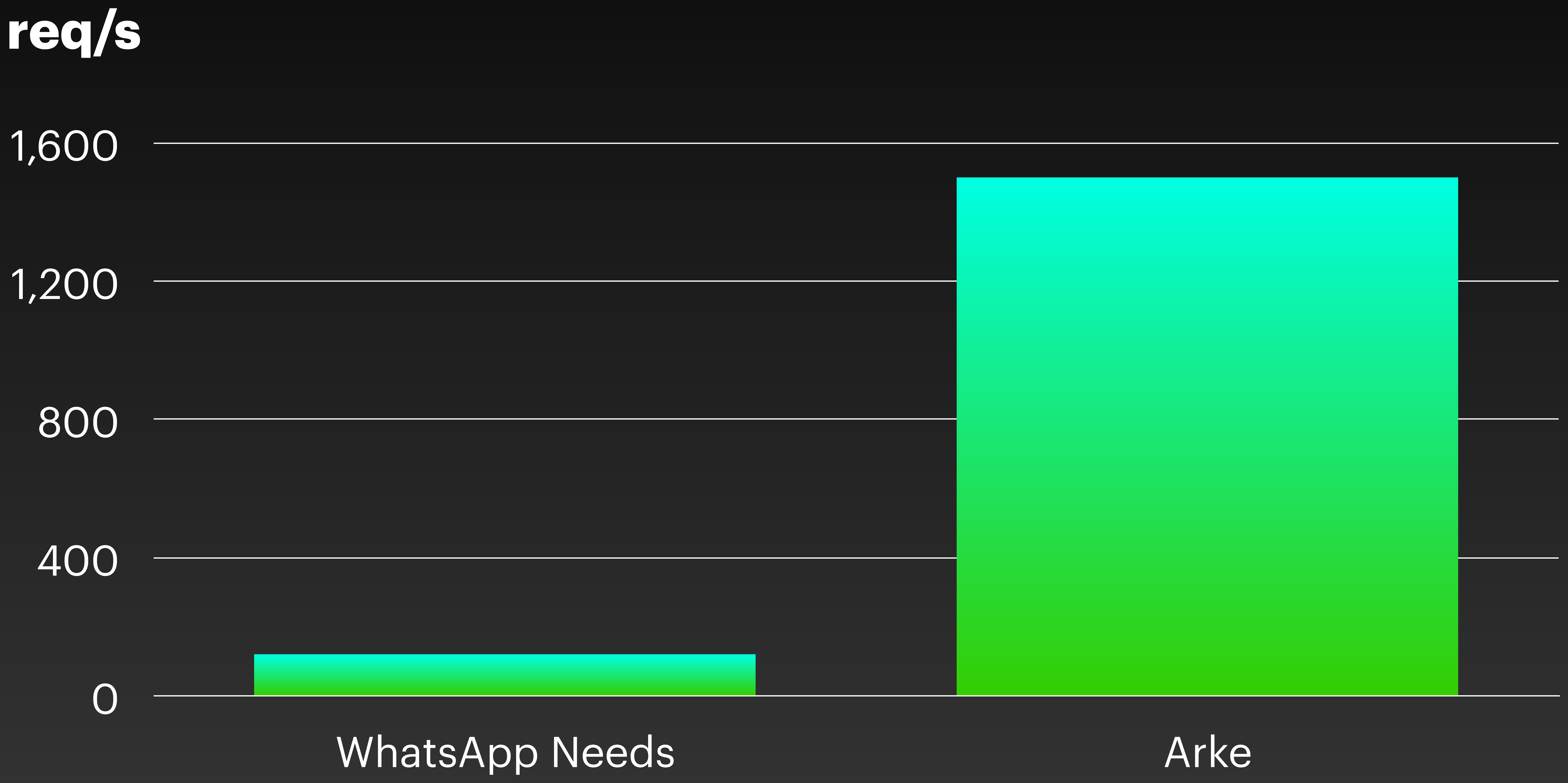
Web3 Needs

- Decentralisation

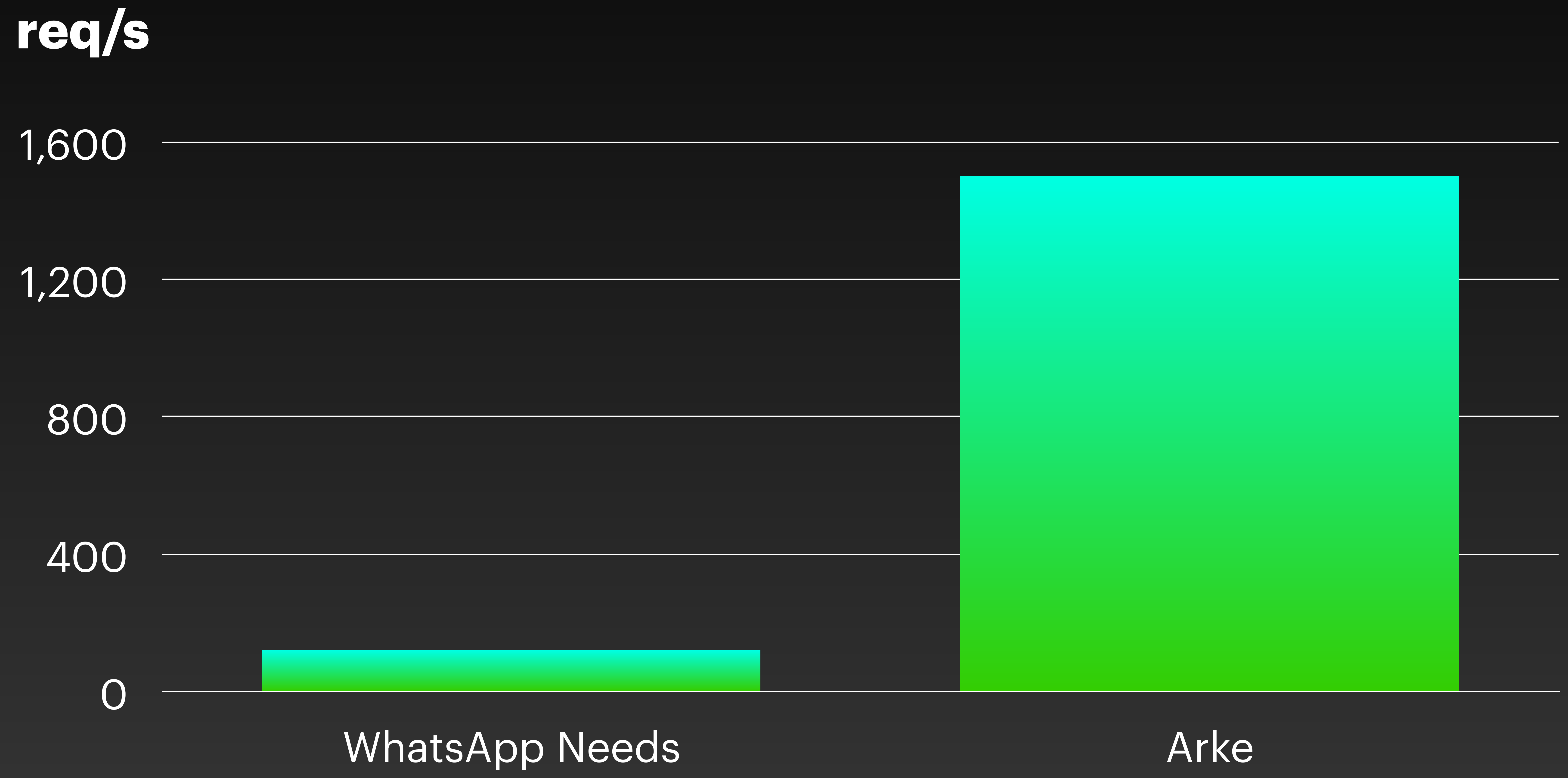
Arke

- O(1) — independent of the total number of users
- Byzantine Fault Tolerant

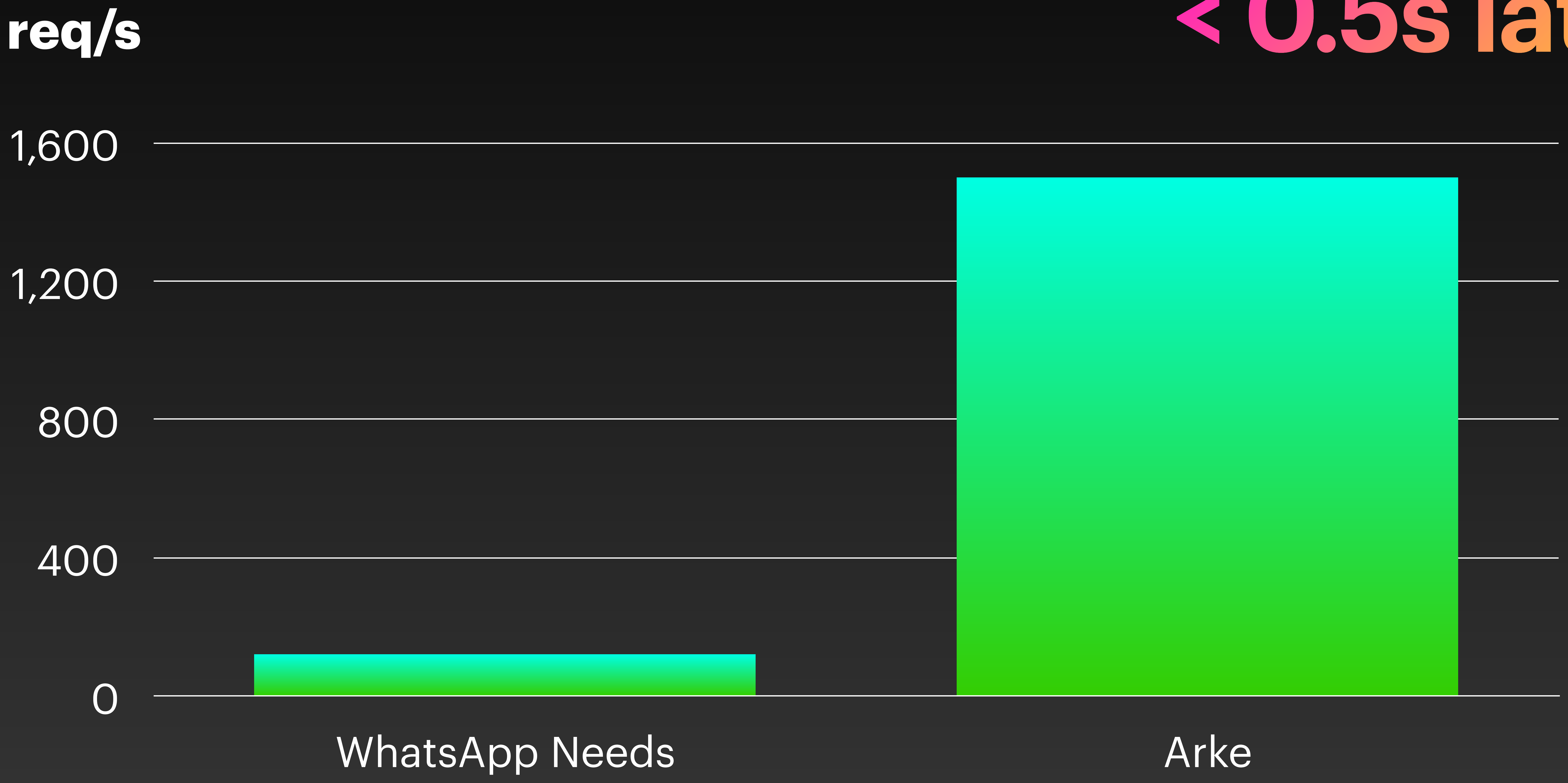

req/s
120
90
60
30
0
WhatsApp Needs


req/s
1,600
1,200
800
400
0
WhatsApp Needs
Arke

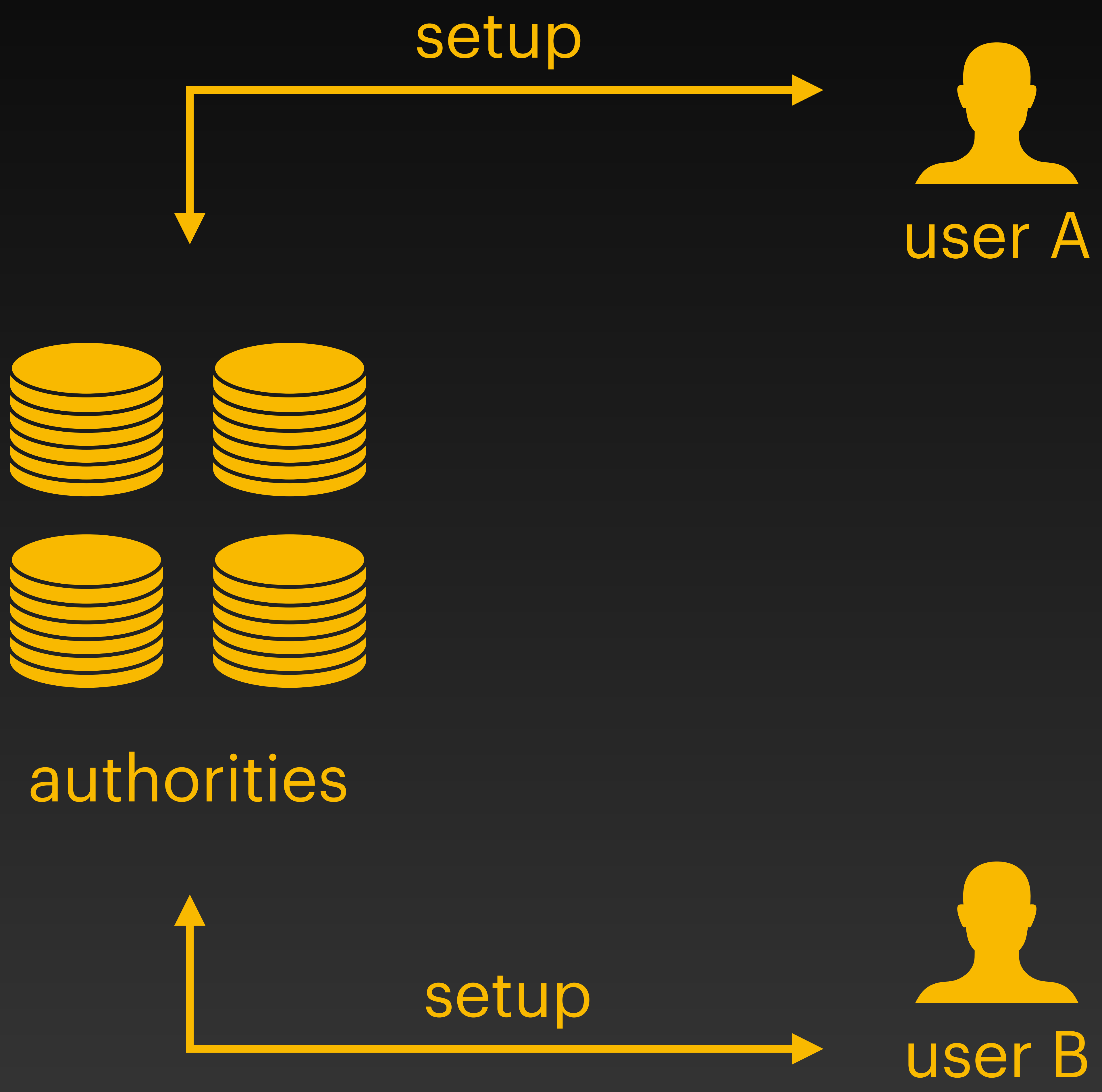
50 nodes

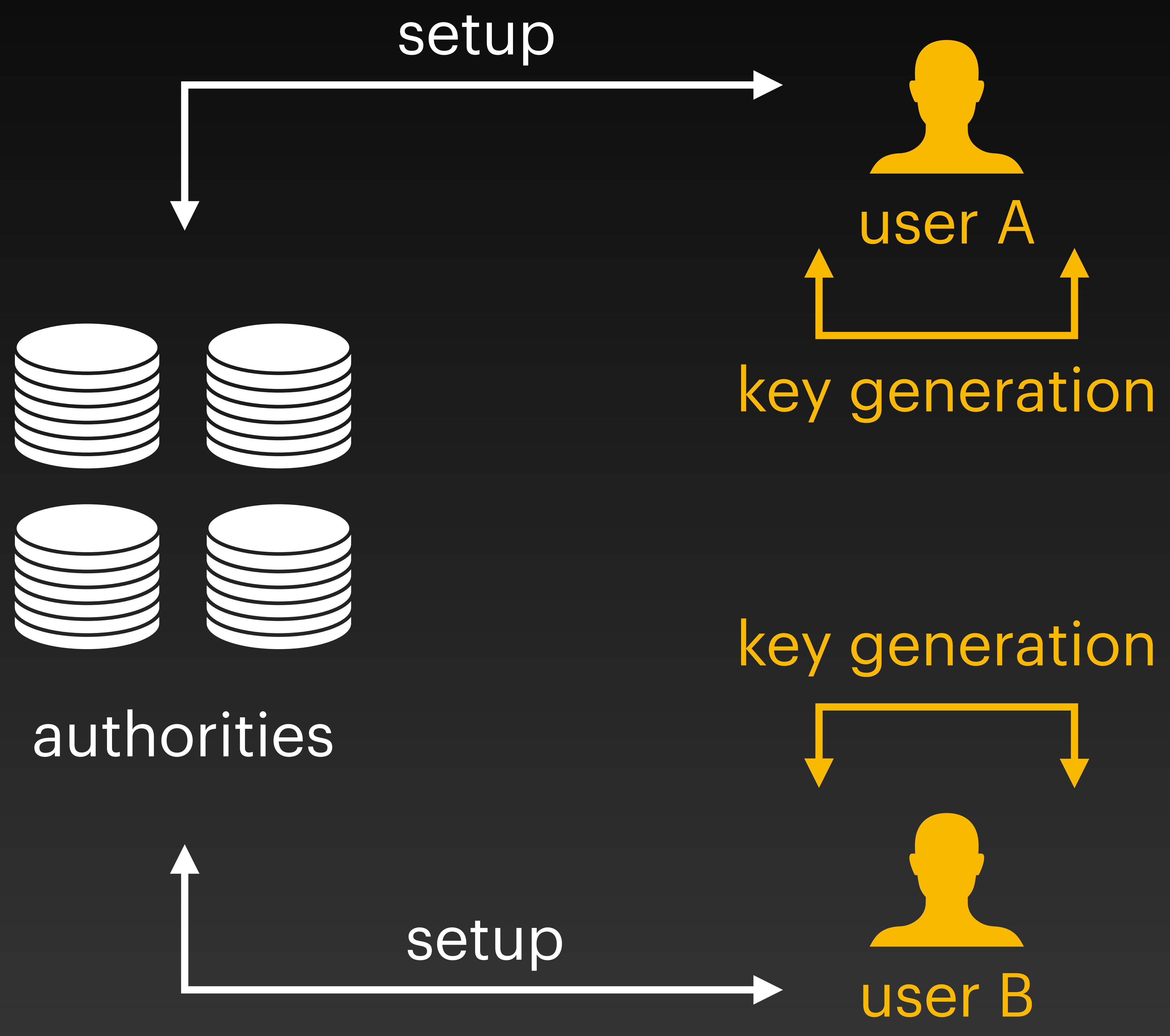


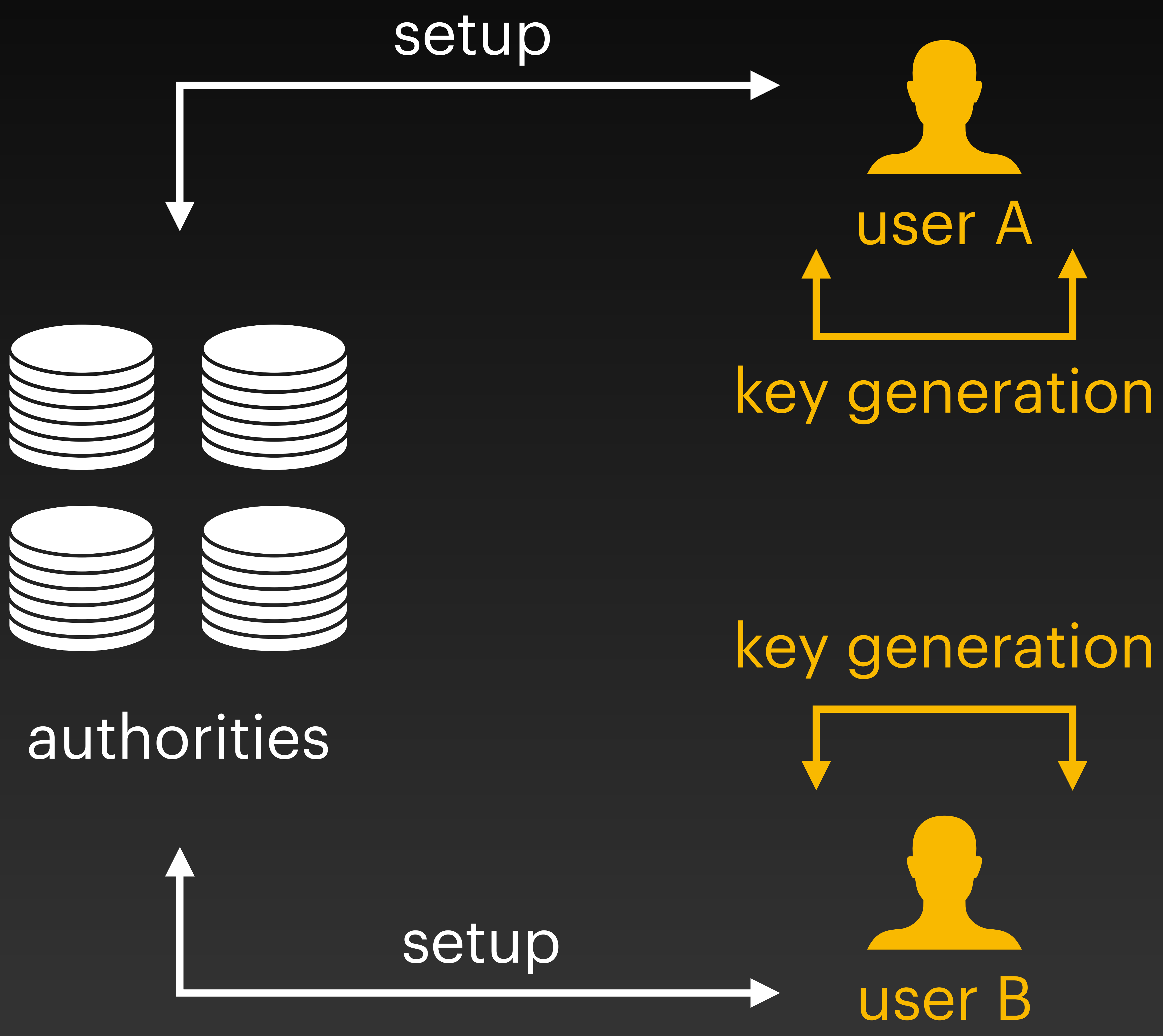
50 nodes
< 0.5s latency



How does it work?


setup
user A
authorities
setup
user B


setup
user A
key generation
authorities
key generation
setup
user B



setup

user A

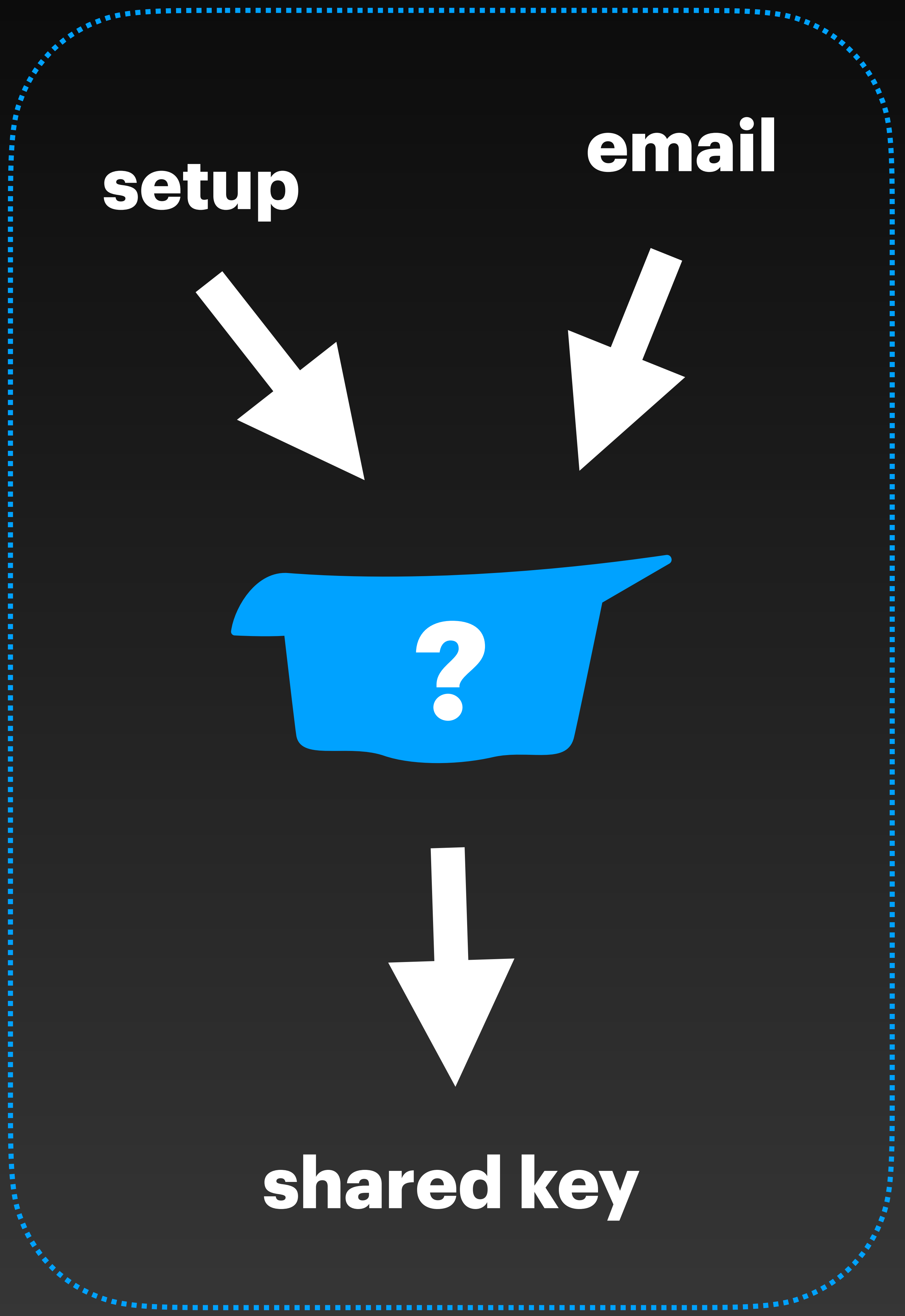
key generation

authorities

key generation

setup

user B

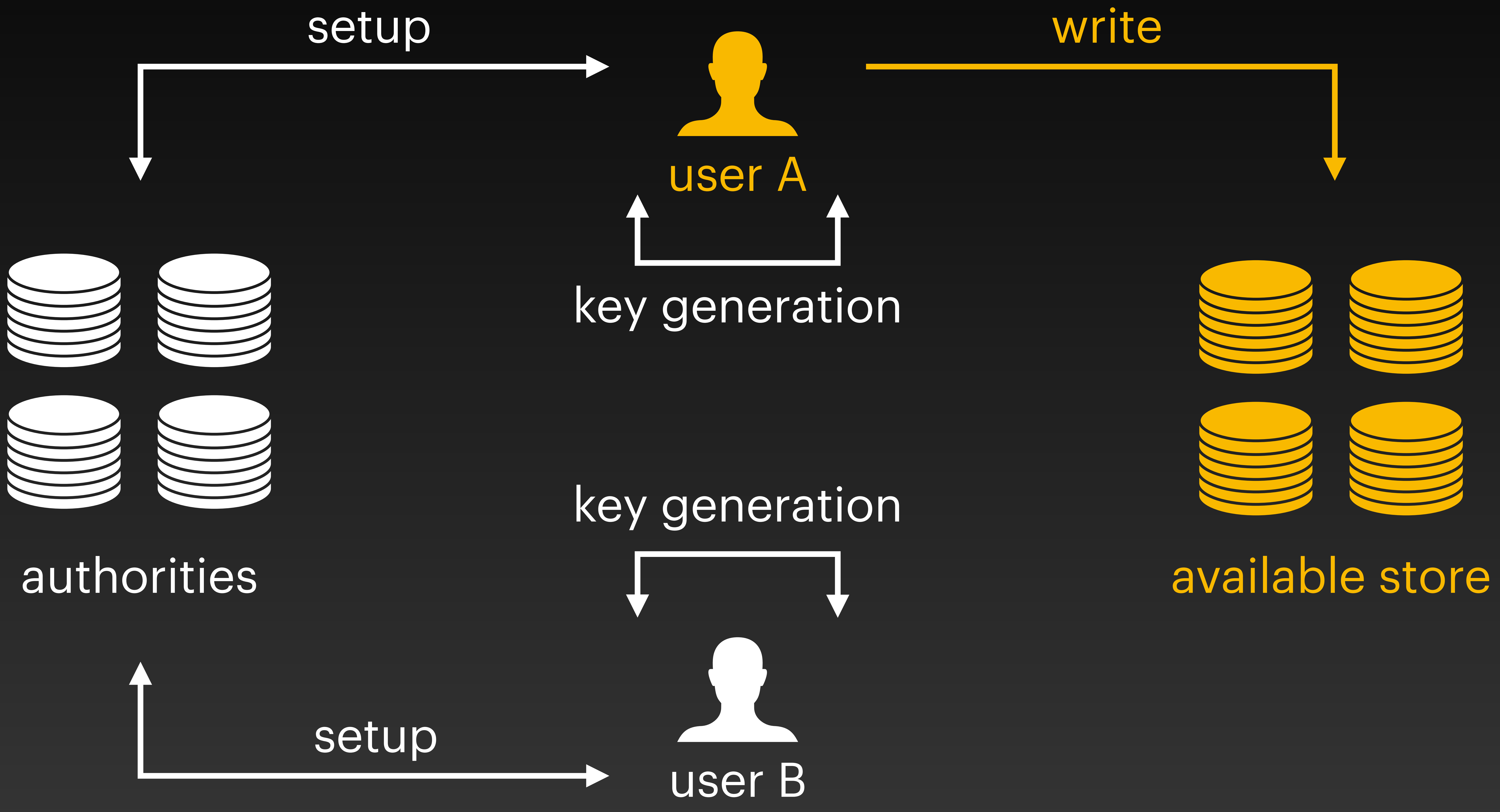


setup

email

?

shared key


setup

write

user A

key generation

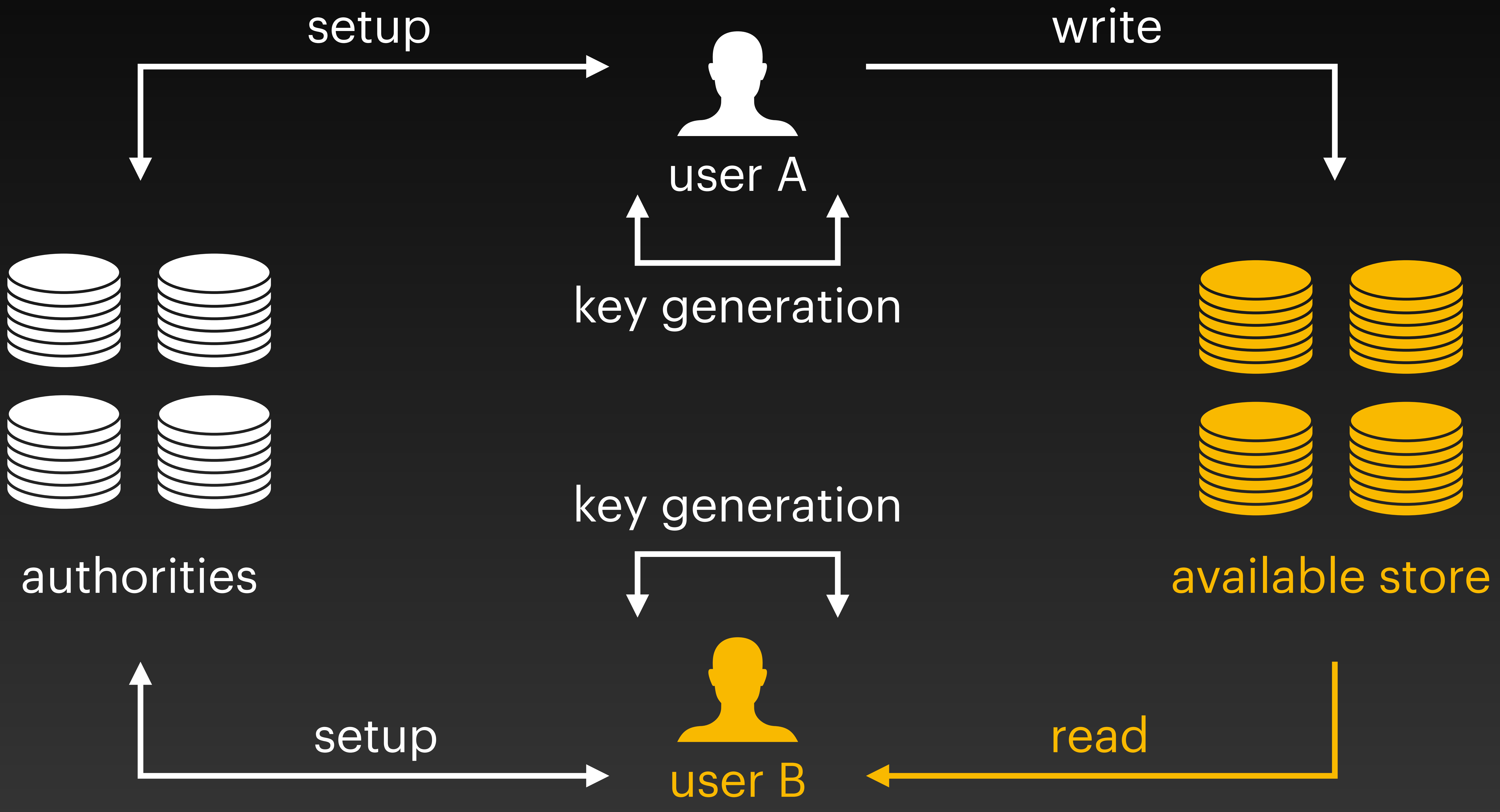
authorities

key generation

available store

setup

user B



setup

user A

write

key generation

authorities

key generation

available store

setup

user B

read

What about blockchains?

MystenLabs

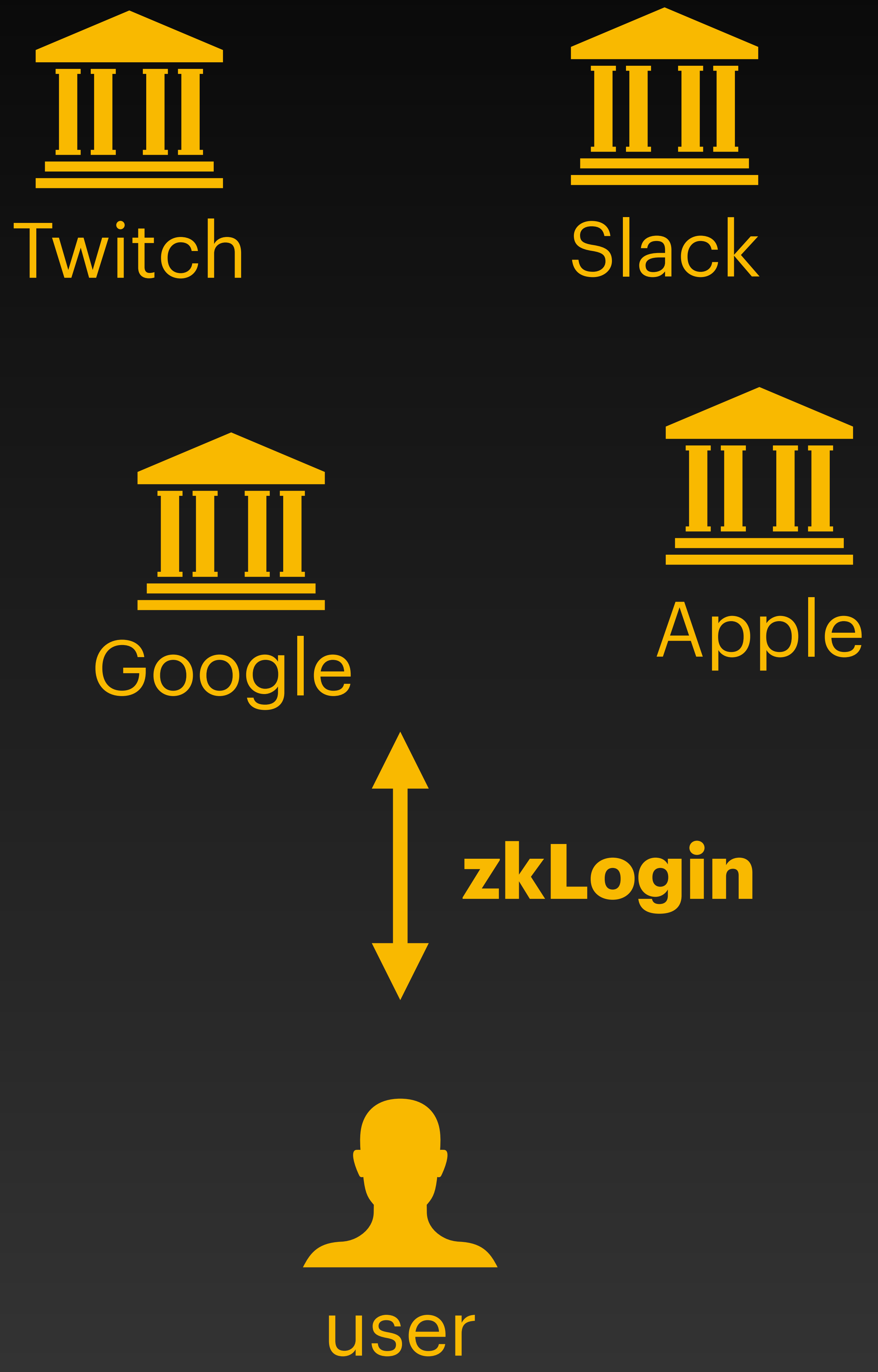
Sui

zkLogin

How to SNARK sign-in with Google, Apple, FB

Kostas Kryptos Chalkias | Yan Ji | Jonas Lindstrøm | Deepak Maram | Ben Riva
Arnab Roy | Mahdi Sedaghat | Joy Wang

TUM Blockchain, Sep 2024



Twitch

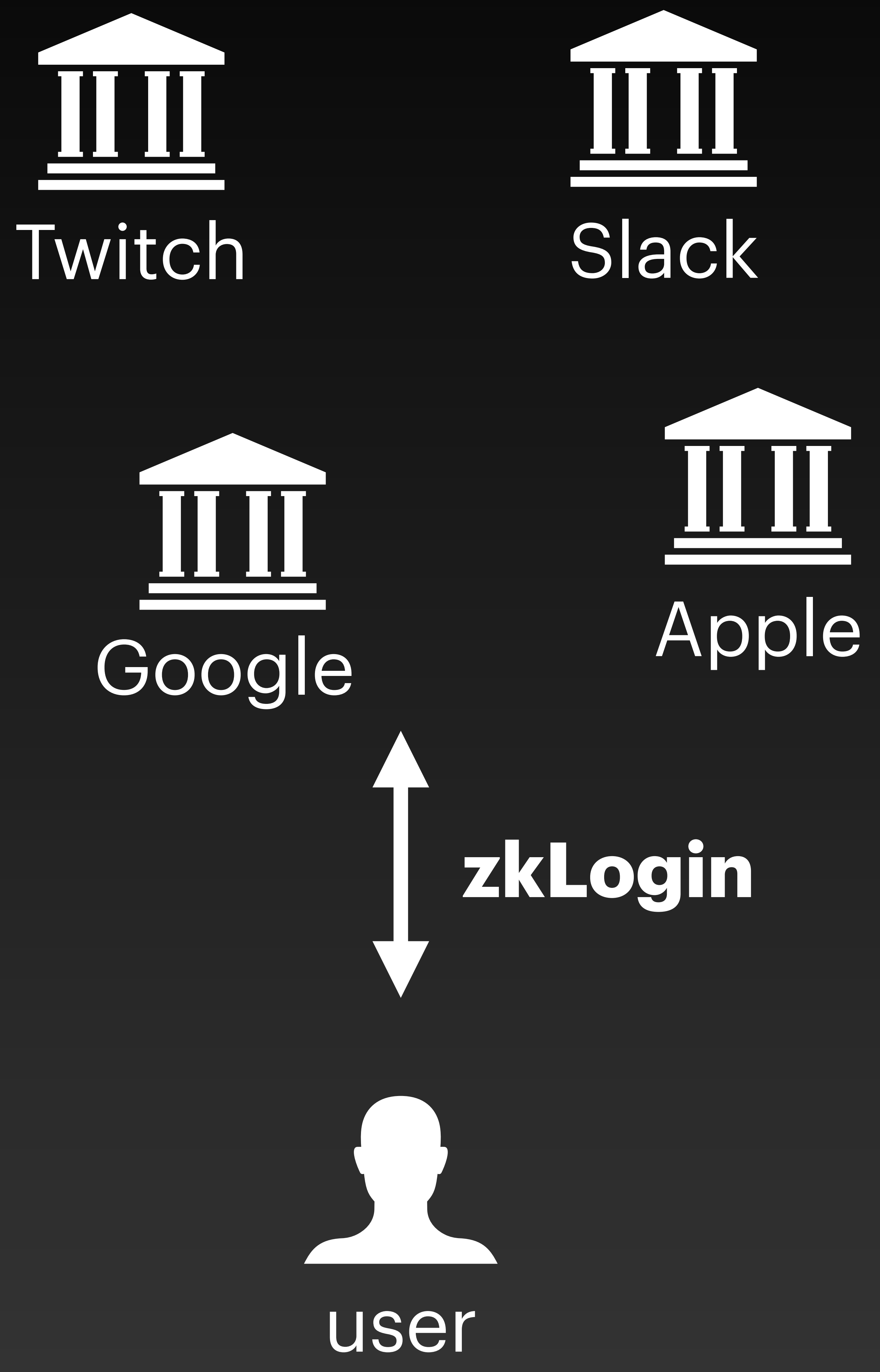
Slack

Google

Apple

zkLogin

user



Twitch

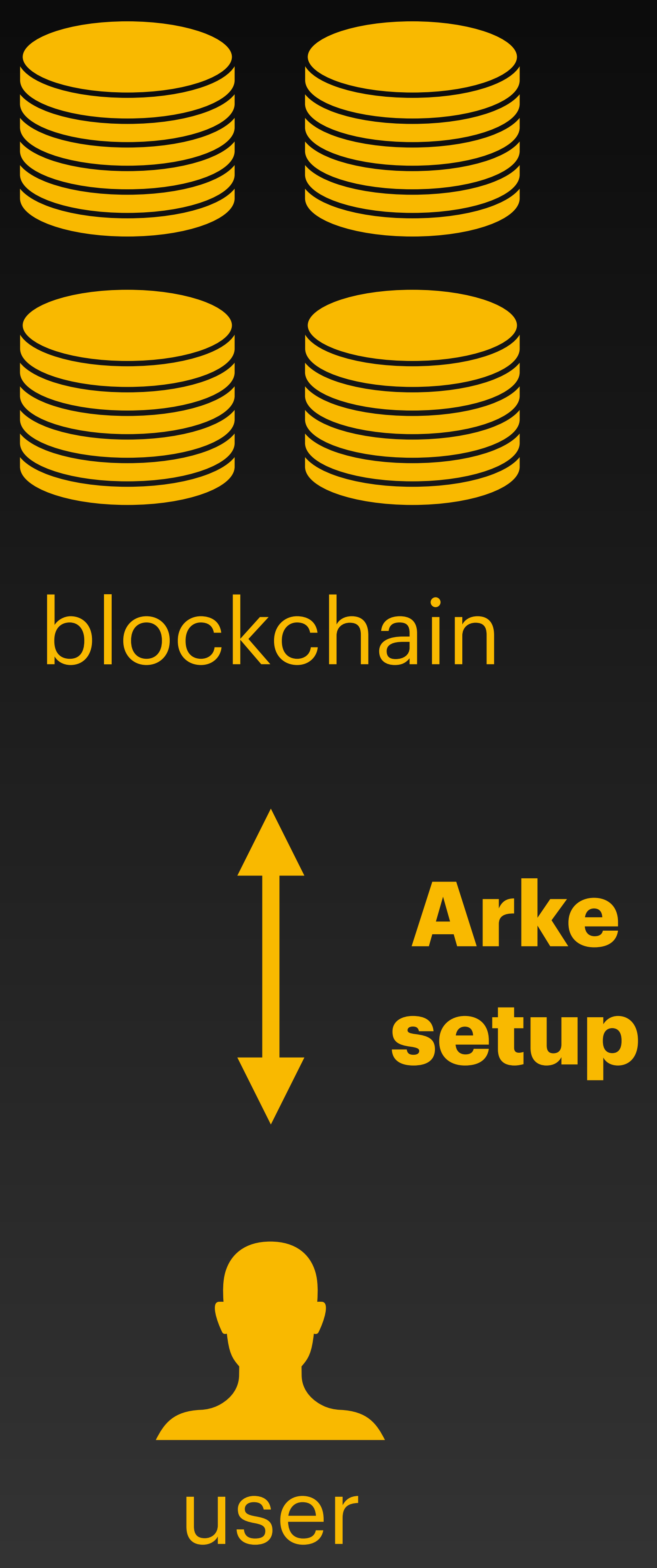
Slack

Google

Apple

zkLogin

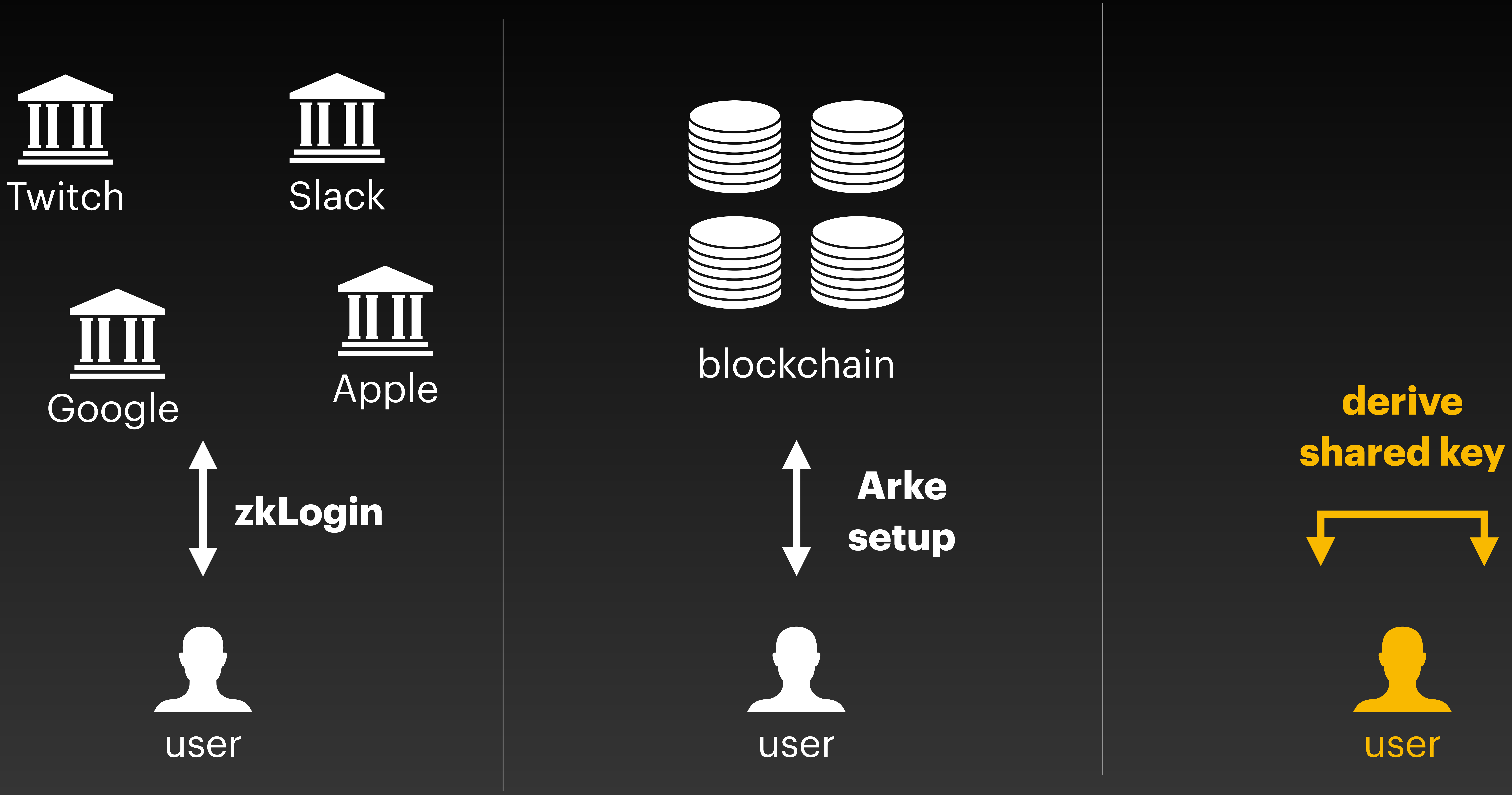
user



blockchain

Arke setup

user



Twitch

Slack

Google

Apple

zkLogin

user

blockchain

Arke setup

user

derive shared key

user

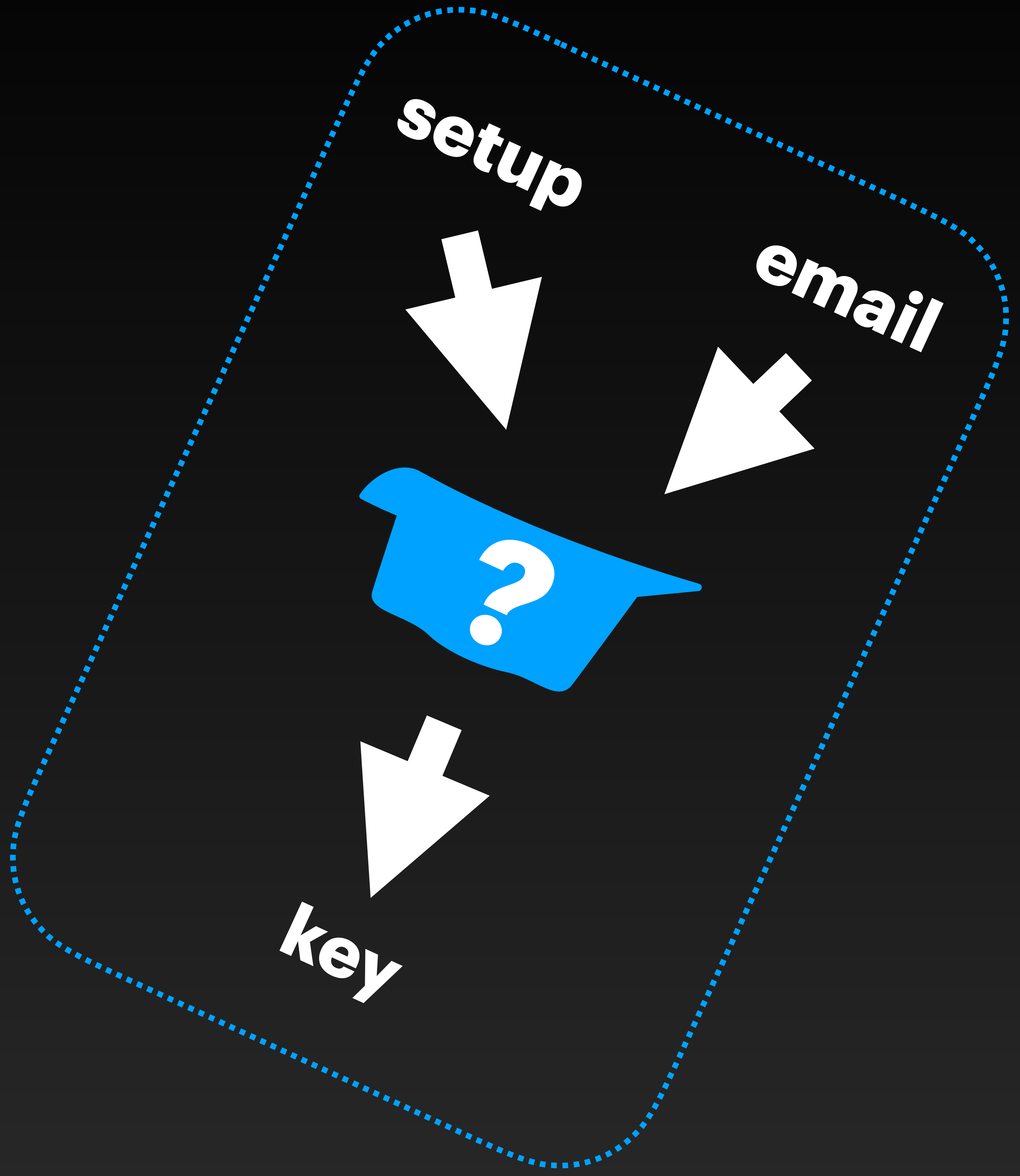
More than private chats

- Decentralised messaging
- Bootstrap multi-user gaming sessions
- Airdrops / payments even before recipient has an account

Paper

alberto@mystenlabs.com

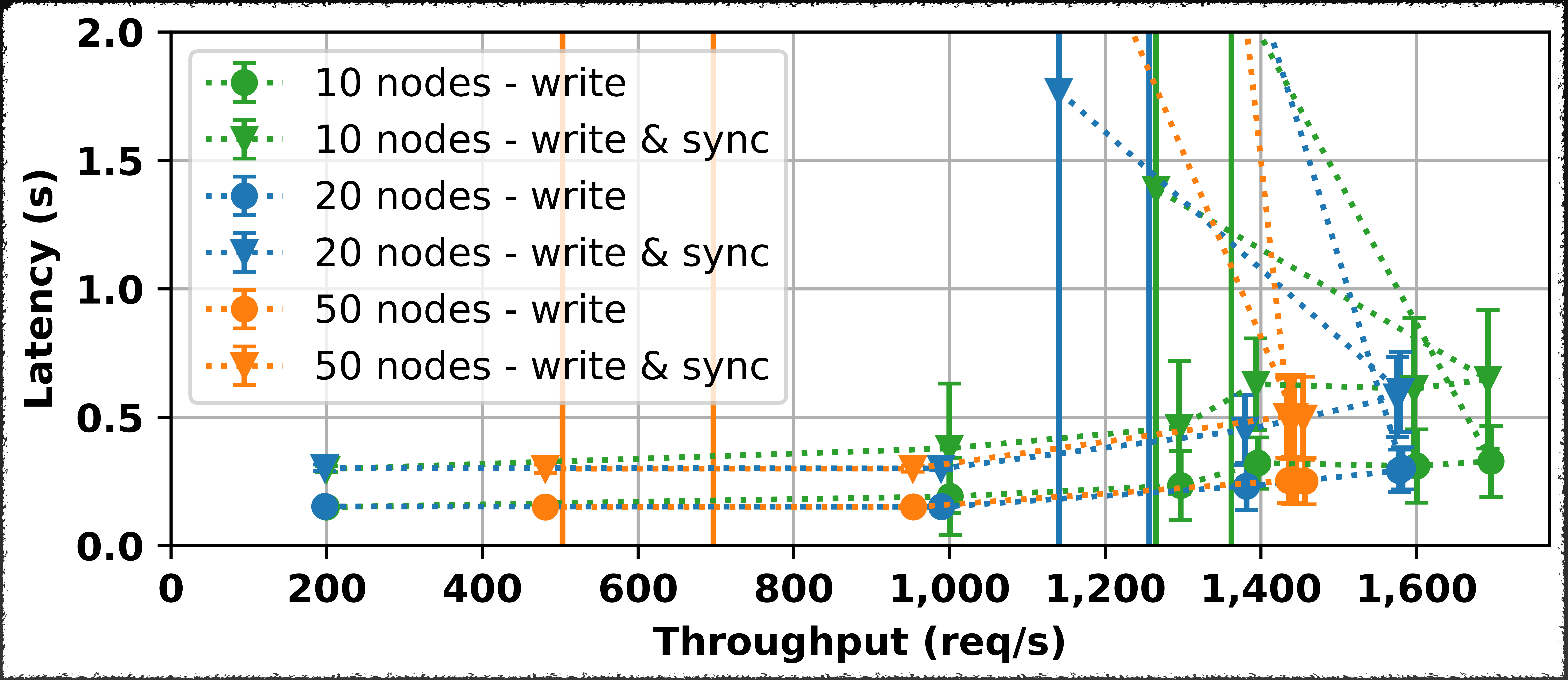
Paper



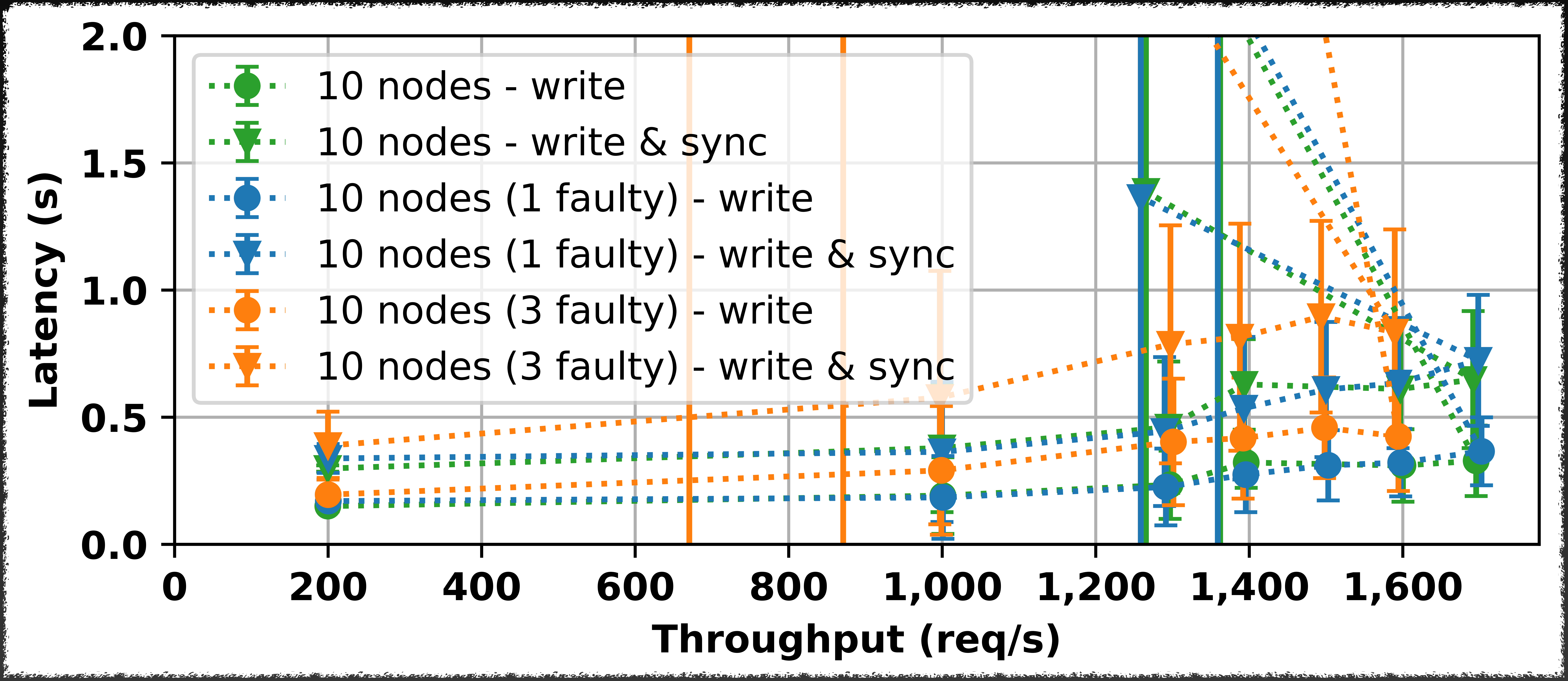
alberto@mystenlabs.com

L-Graphs

Performance

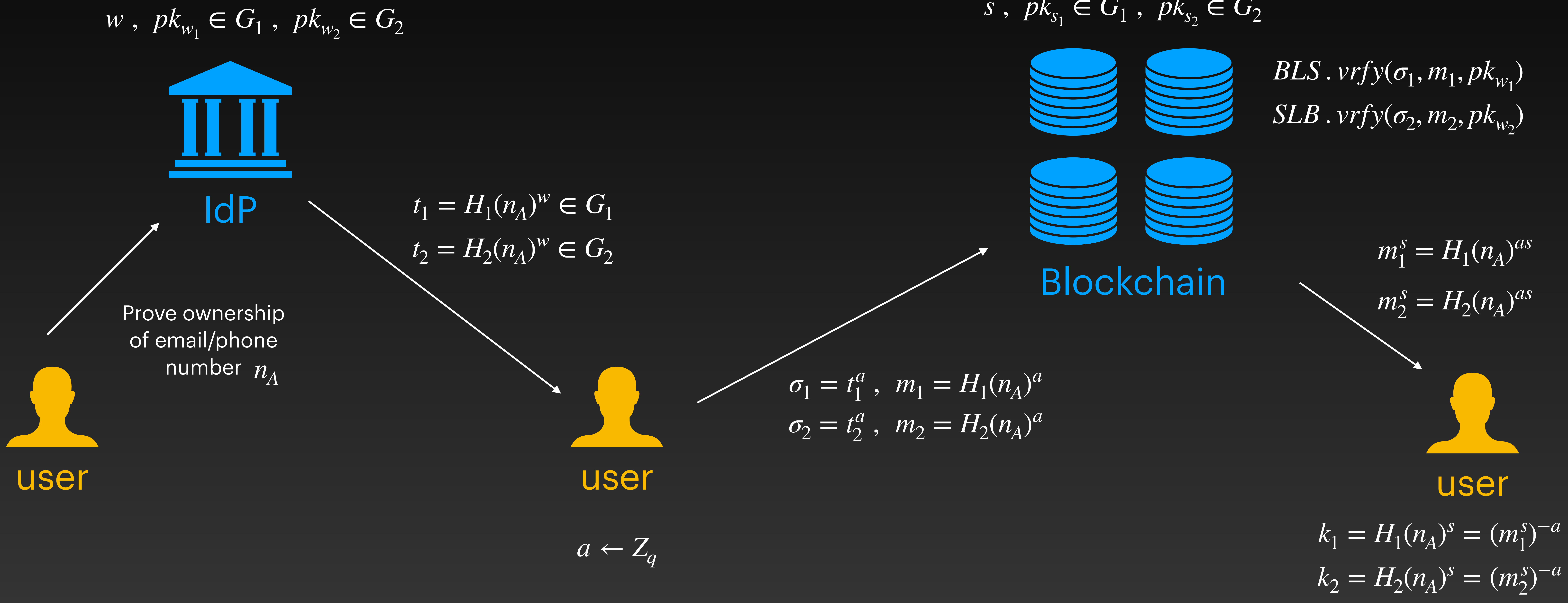


Performance



The Crypto

Setup



$w \ , \ pk_{w_1} \in G_1 \ , \ pk_{w_2} \in G_2$

IdP

user

Prove ownership of email/phone number n_A

$t_1 = H_1(n_A)^w \in G_1$

$t_2 = H_2(n_A)^w \in G_2$

user

$a \leftarrow Z_q$

$\sigma_1 = t_1^a \ , \ m_1 = H_1(n_A)^a$

$\sigma_2 = t_2^a \ , \ m_2 = H_2(n_A)^a$

$s \ , \ pk_{s_1} \in G_1 \ , \ pk_{s_2} \in G_2$

Blockchain

$BLS\,.\,vrfy(\sigma_1, m_1, pk_{w_1})$

$SLB\,.\,vrfy(\sigma_2, m_2, pk_{w_2})$

$m_1^s = H_1(n_A)^{as}$

$m_2^s = H_2(n_A)^{as}$

user

$k_1 = H_1(n_A)^s = (m_1^s)^{-a}$

$k_2 = H_2(n_A)^s = (m_2^s)^{-a}$

Key Derivation

user A

$$S_{AB} = e(k_1, H_2(n_B)) = e(H_1(n_A)^s, H_2(n_B))$$
$$S_{BA} = e(H_1(n_B), k_2) = e(H_1(n_B), H_2(n_A)^s)$$

$$k_{AB} = KDF(S_{AB}\ XOR\ S_{BA})$$

$$key = g_1^{t_{AB}}\ ,\ \ t_{AB} = H(s_{AB})$$
$$val = c_{AB} = AEAD_k(pk_A)$$

user B

$$S_{AB} = e(H_1(n_A), k_2) = e(H_1(n_A), H_2(n_B)^s)$$
$$S_{BA} = e(k_1, H_2(n_A)) = e(H_1(n_B)^s, H_2(n_A))$$

$$k_{AB} = KDF(S_{AB}\ XOR\ S_{BA})$$

$$key = g_1^{t_{BA}}\ ,\ \ t_{BA} = H(s_{BA})$$
$$val = c_{BA} = AEAD_k(pk_B)$$

Sui is special

user A

$$S_{AB} = e(k_1, H_2(n_B)) = e(H_1(n_A)^s, H_2(n_B))$$
$$S_{BA} = e(H_1(n_B), k_2) = e(H_1(n_B), H_2(n_A)^s)$$

$$k_{AB} = KDF(S_{AB}\ XOR\ S_{BA})$$

$$key = g_1^{t_{AB}}\ ,\ \ t_{AB} = H(s_{AB})$$
$$val = c_{AB} = AEAD_k(addr_A)$$

1. Create a new owned object with owner *hash(key)*
2. The object/event contains a single field: *val*
3. Readers gather all objects owned by a public key
4. Single-owner object structure remains because there is a single writer for every key