

Arke

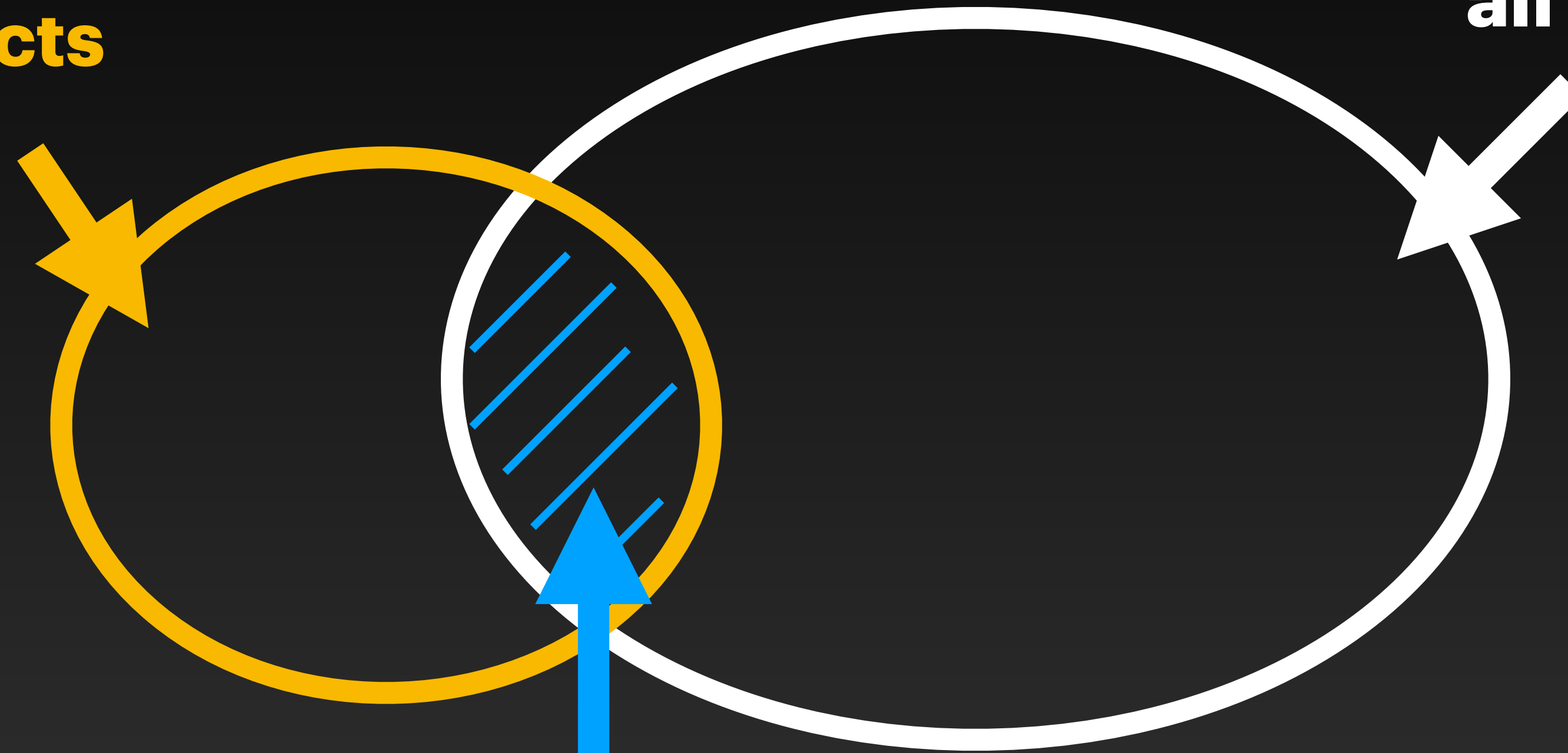
Scalable and Byzantine Fault Tolerant Privacy-
Preserving Contact Discovery

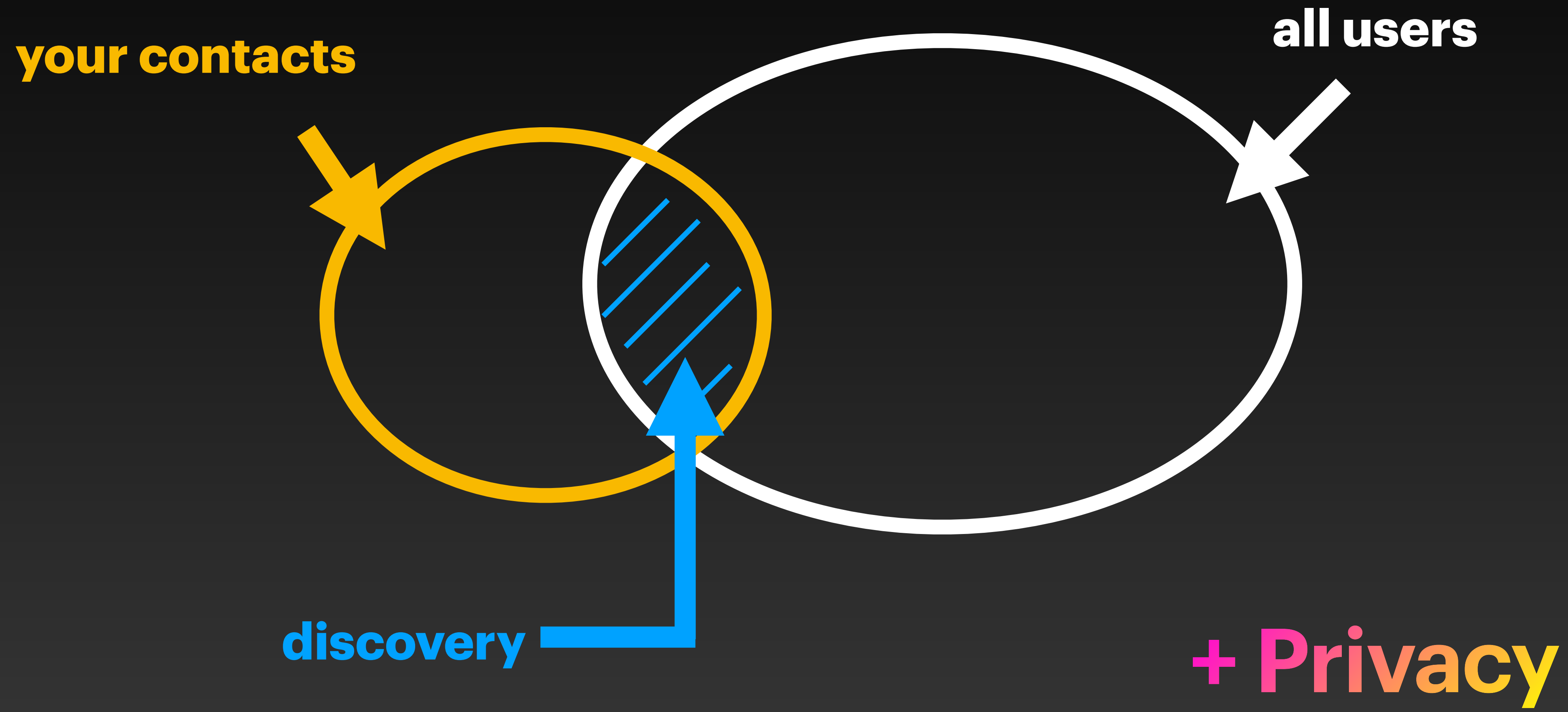
Alberto Sonnino

your contacts

all users

discovery





WhatsApp Needs

- 10 Million updates / day

Blockchains

- Decentralisation

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

req/s

120

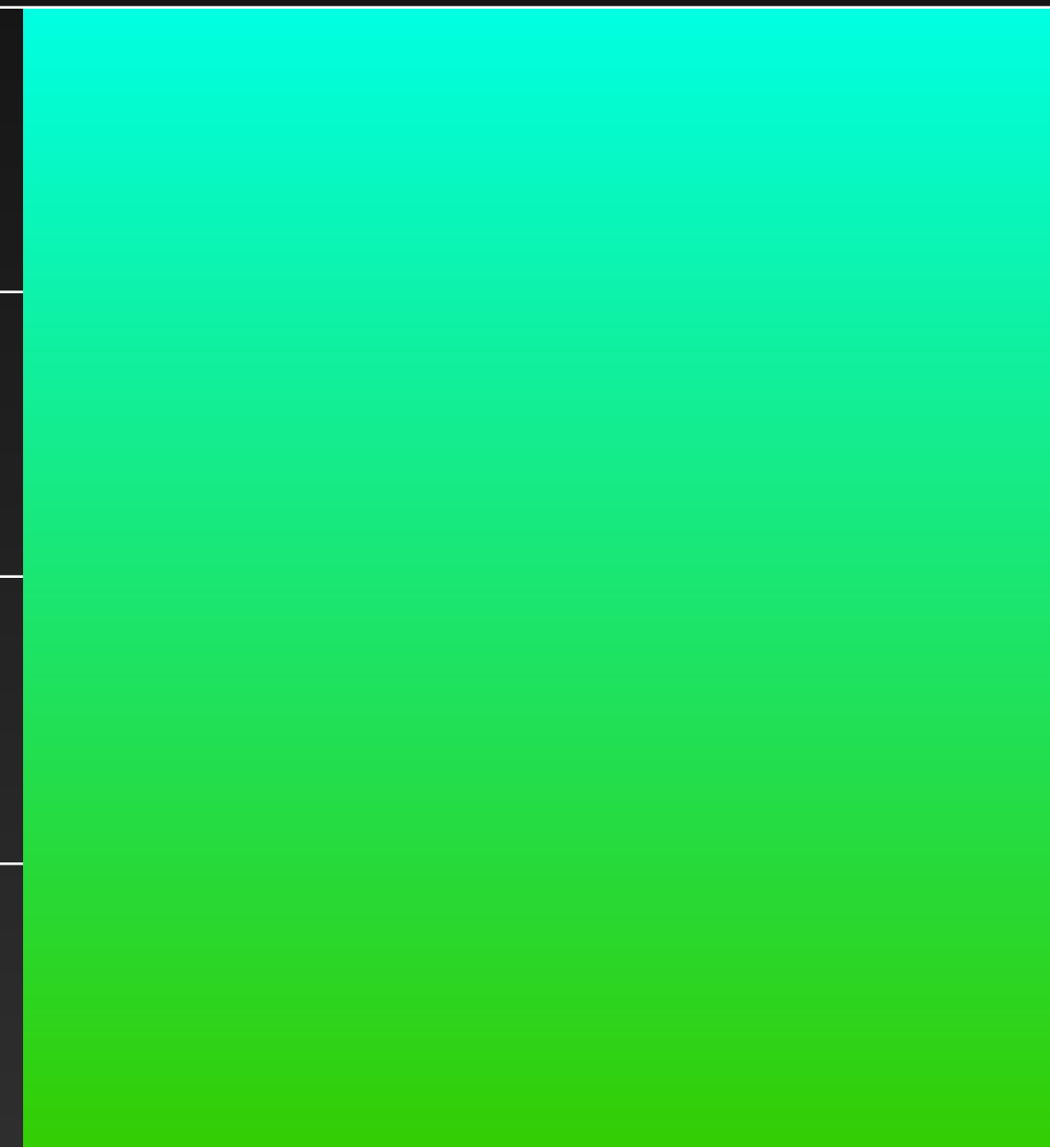
90

60

30

0

WhatsApp Needs



req/s

1,600

1,200

800

400

0

WhatsApp Needs

Arke



50 nodes

req/s

1,600

1,200

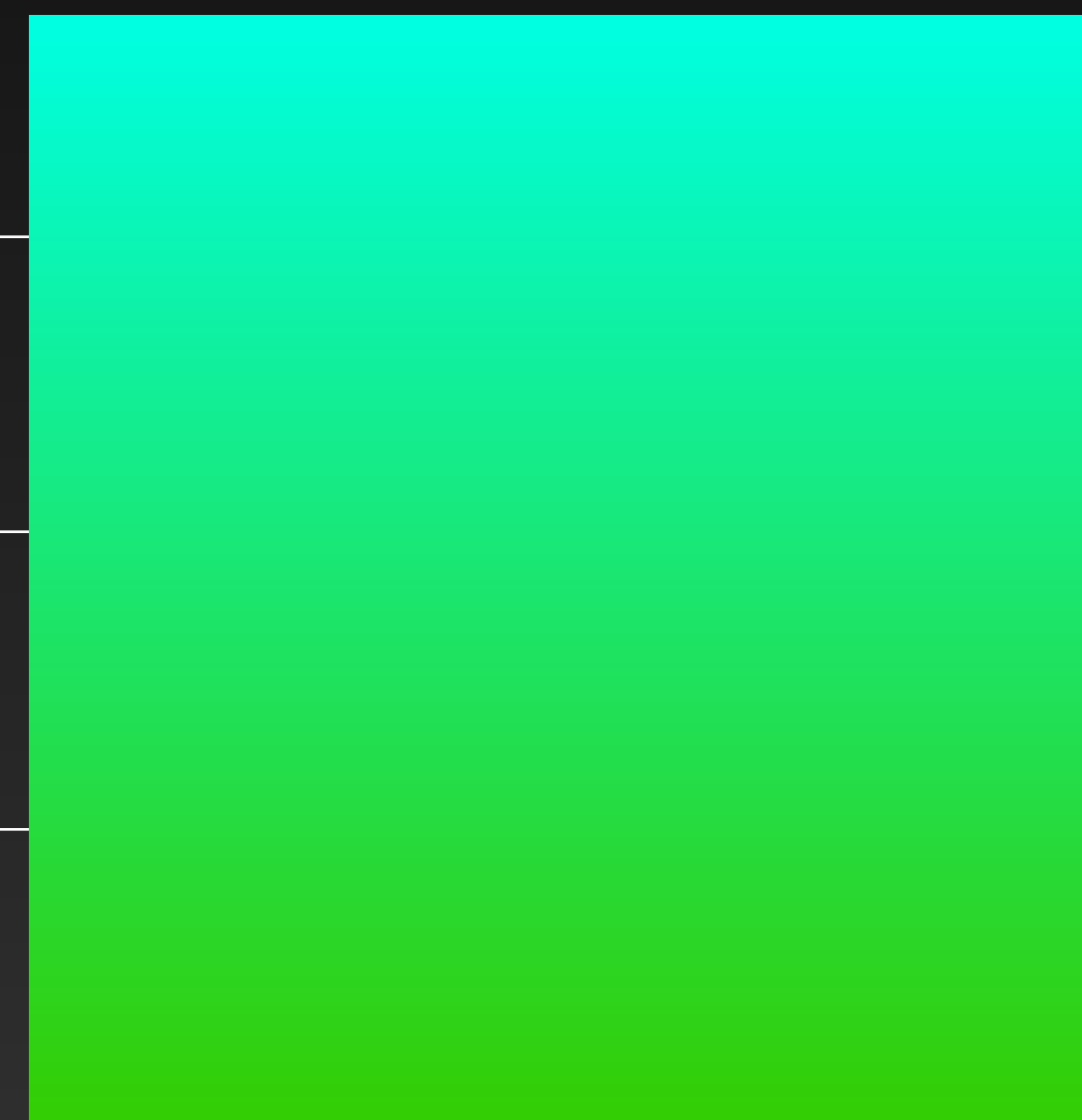
800

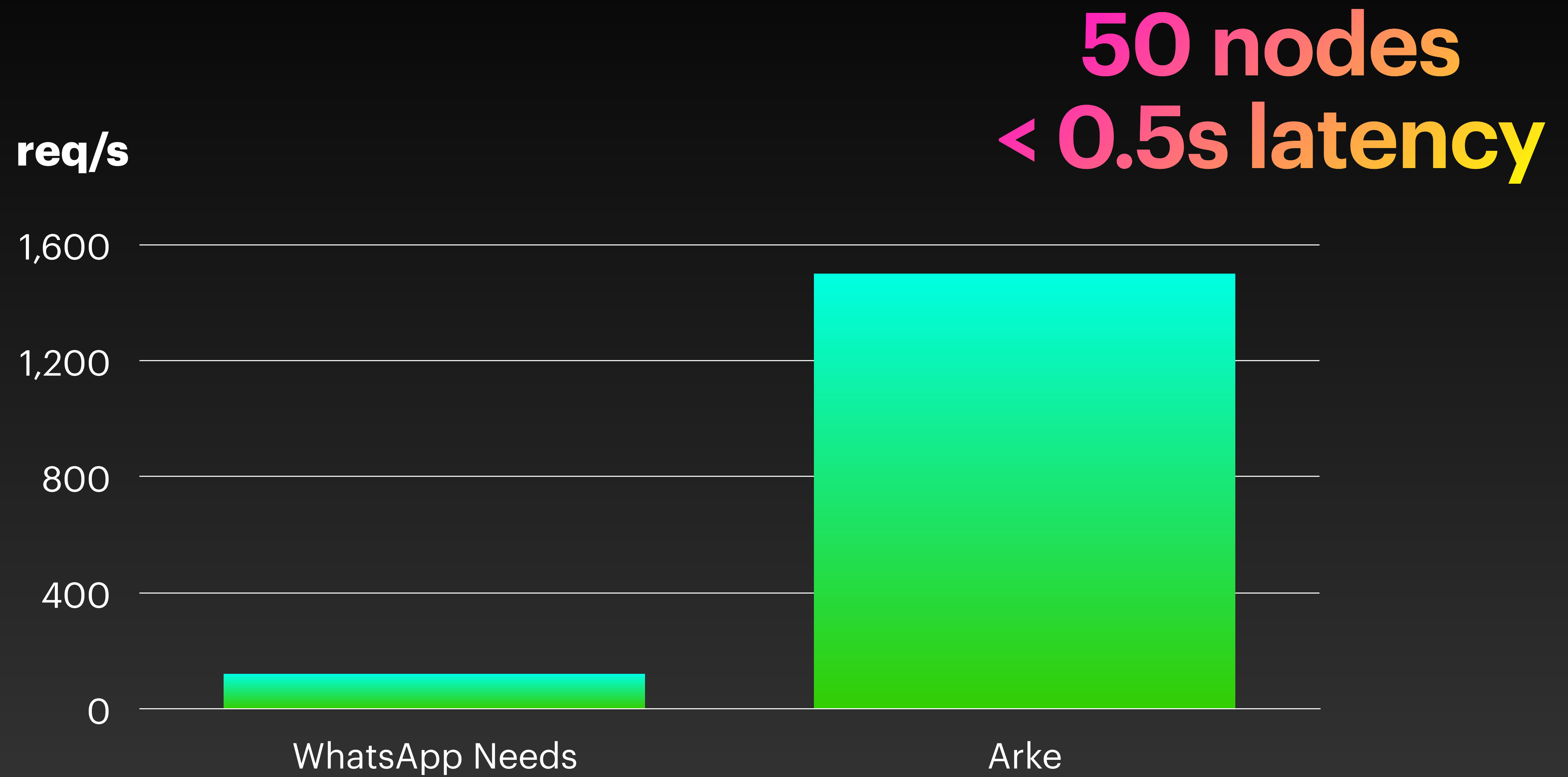
400

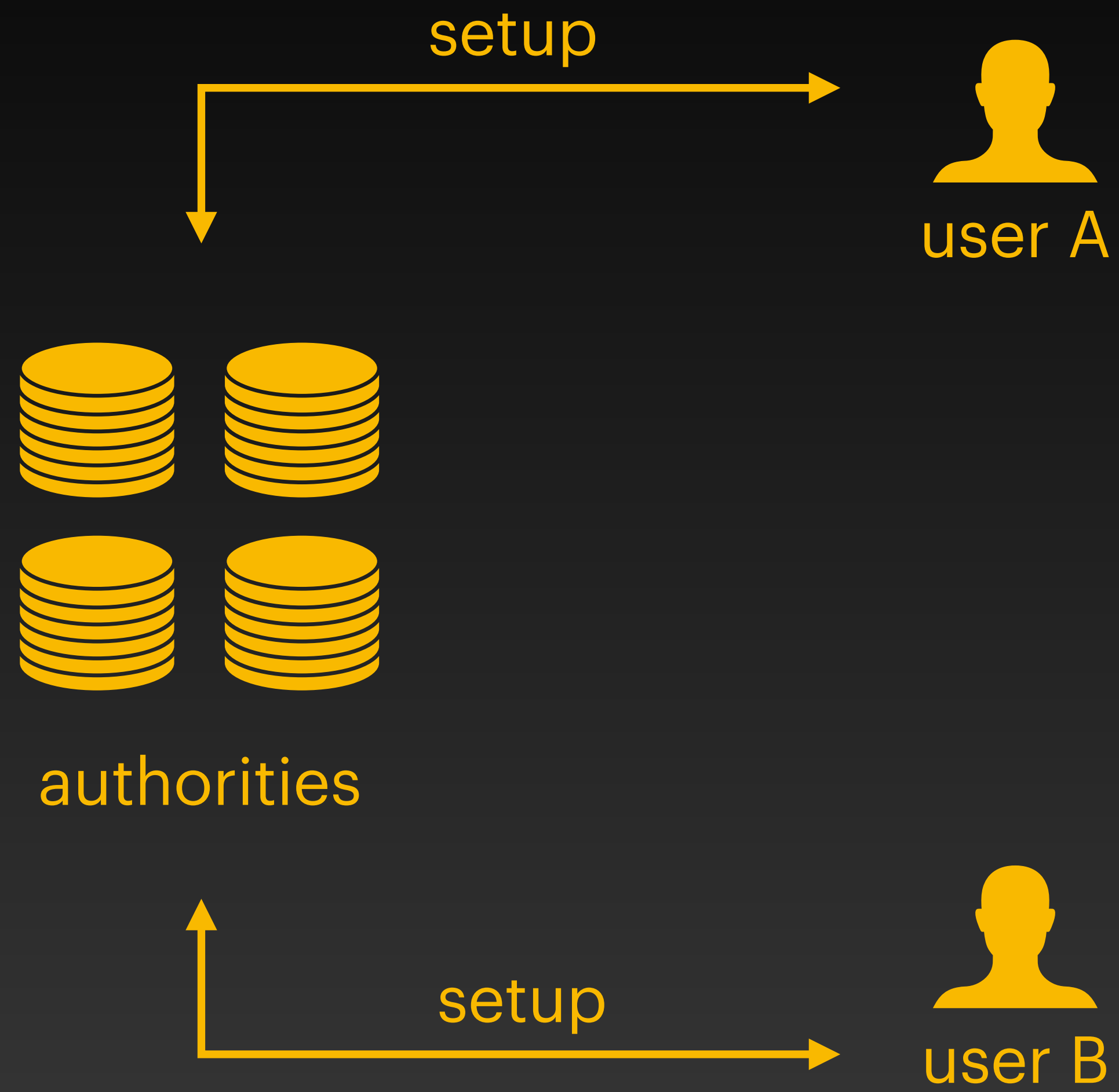
0

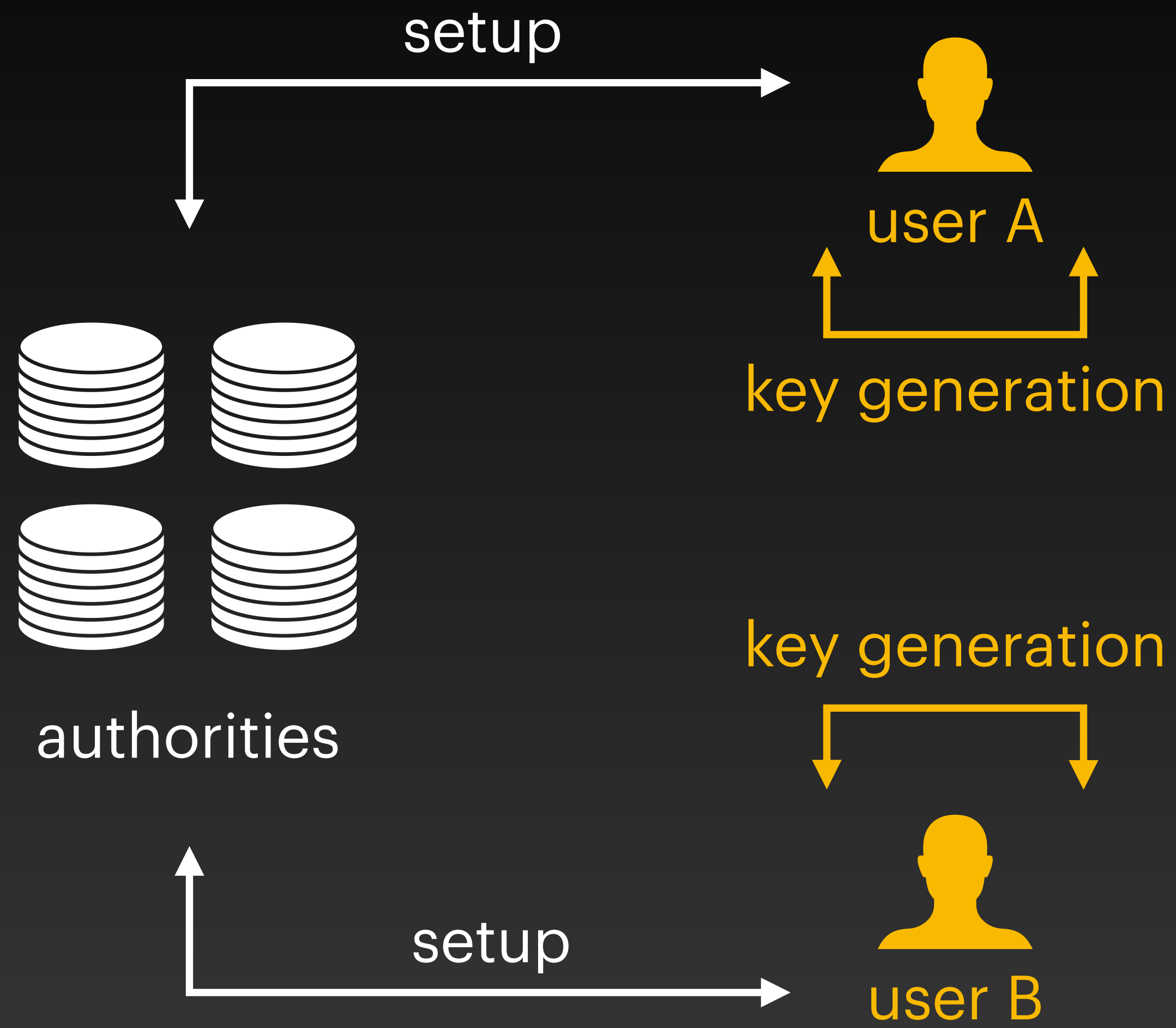
WhatsApp Needs

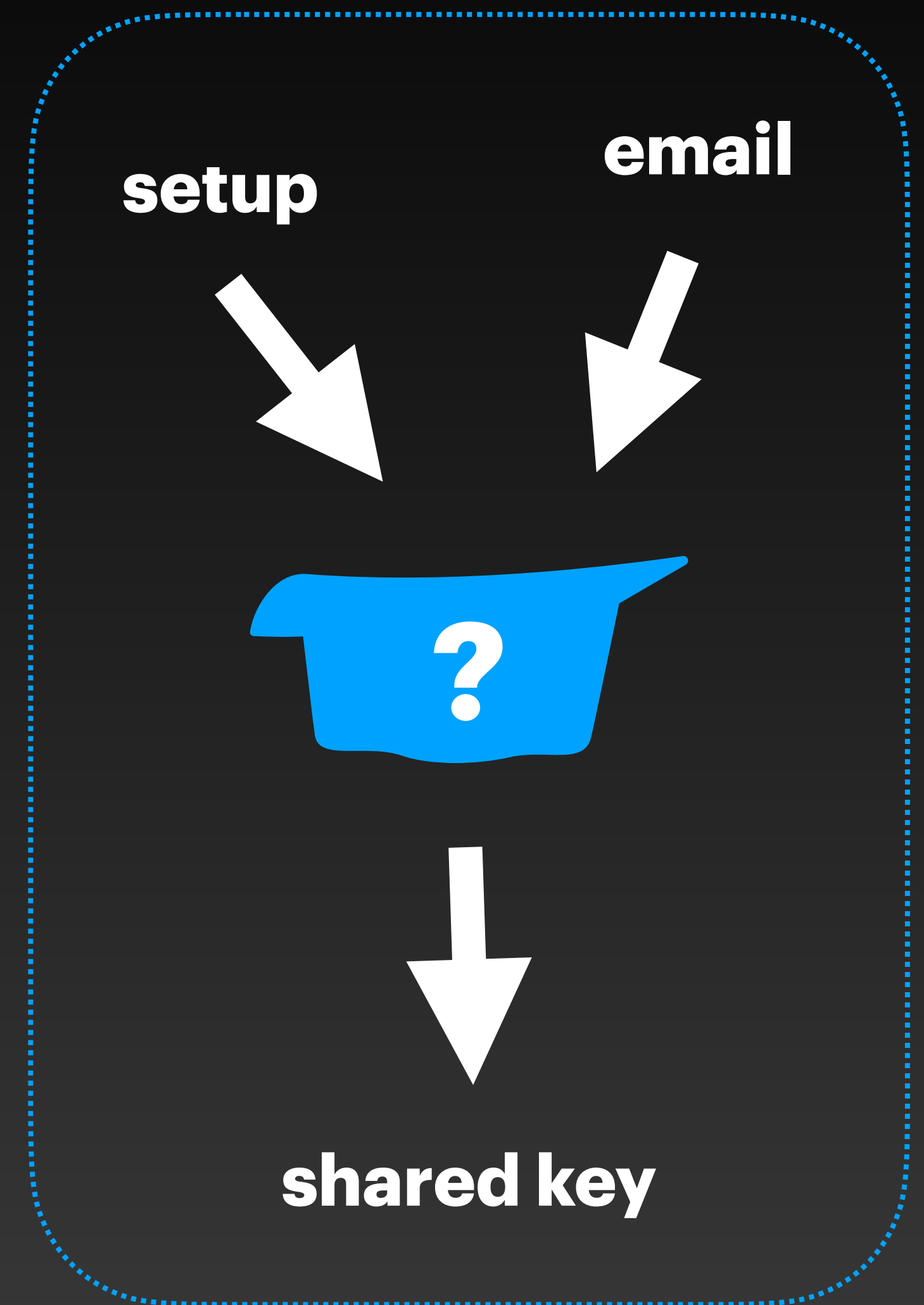
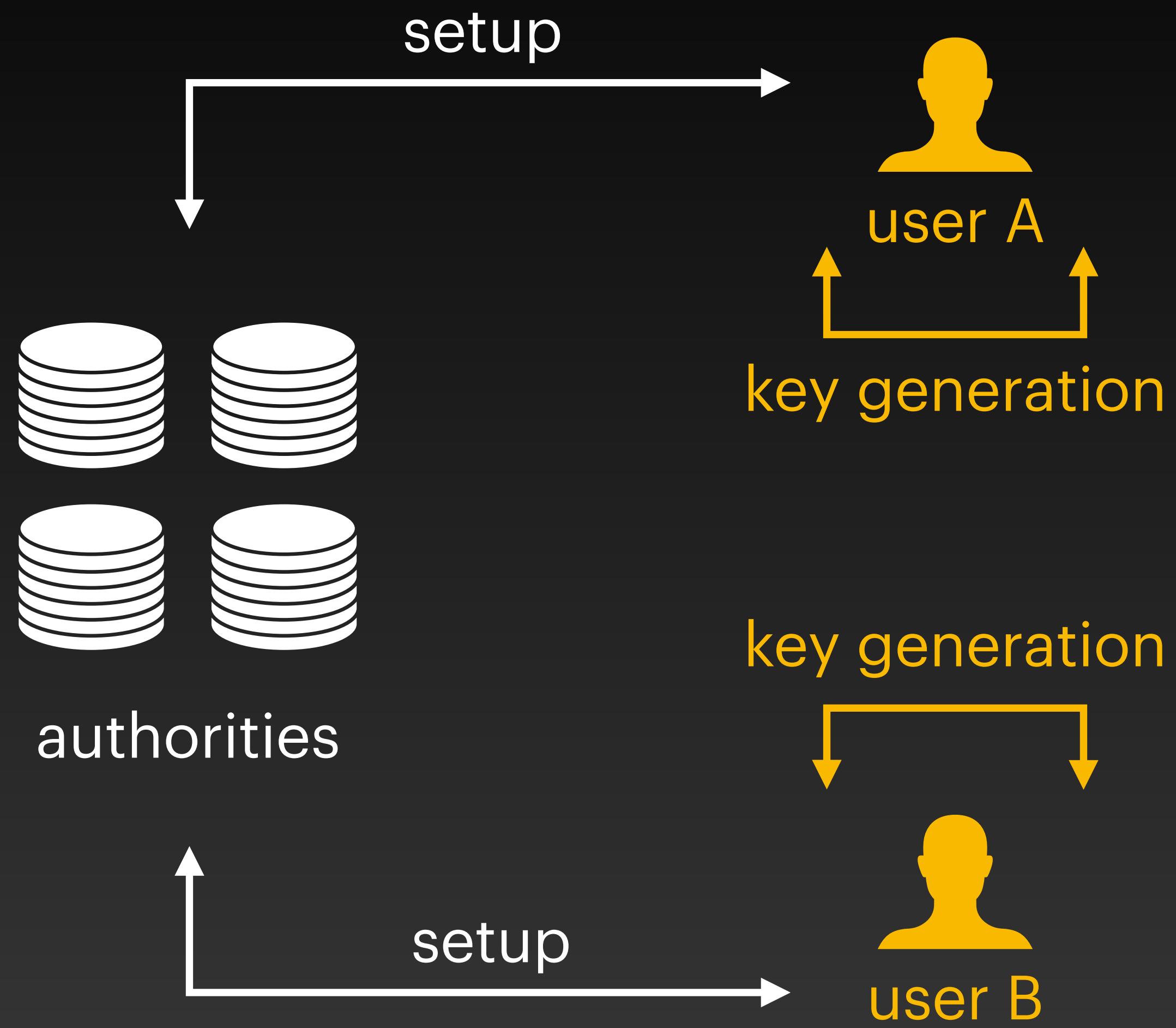
Arke

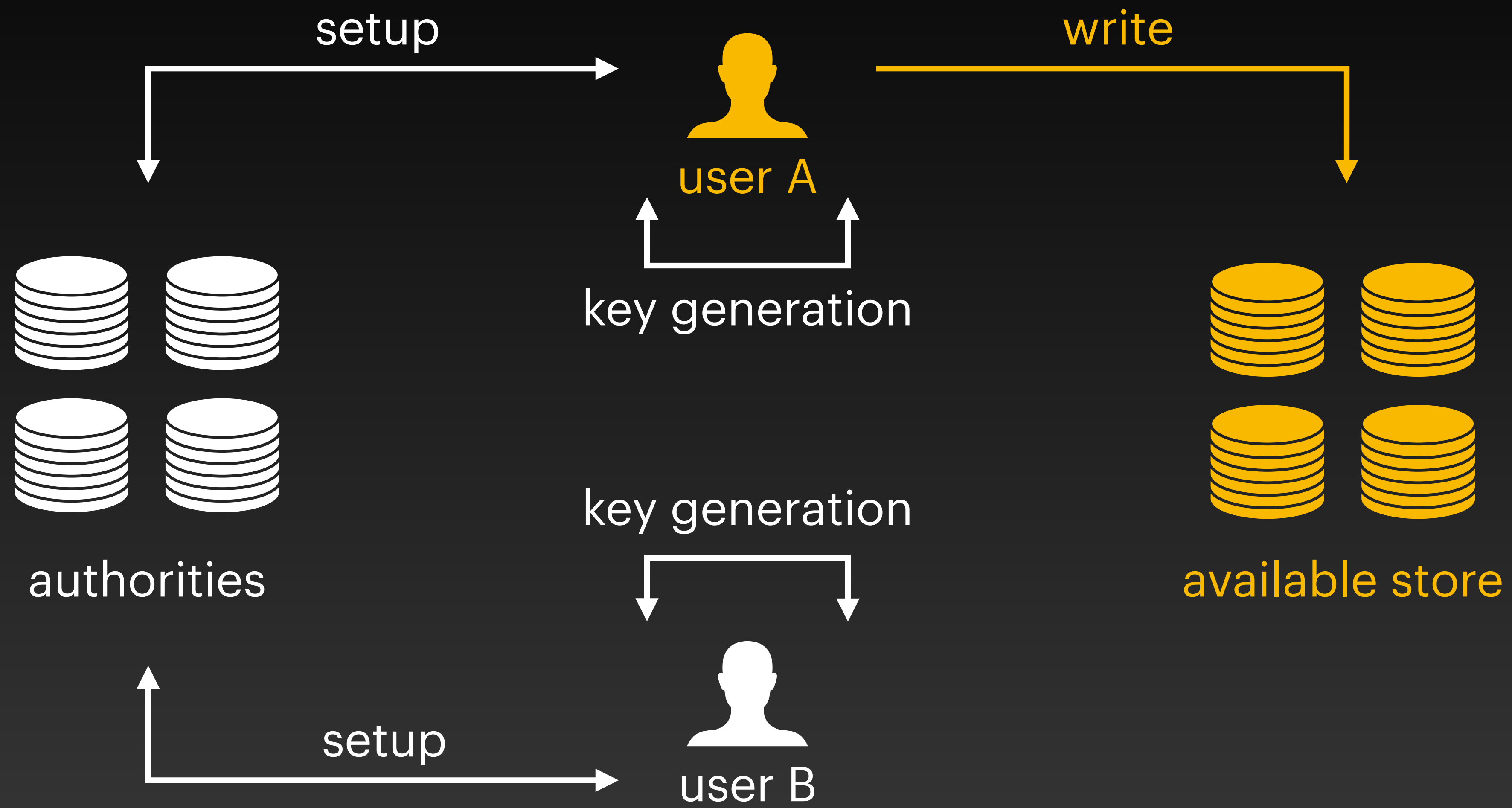


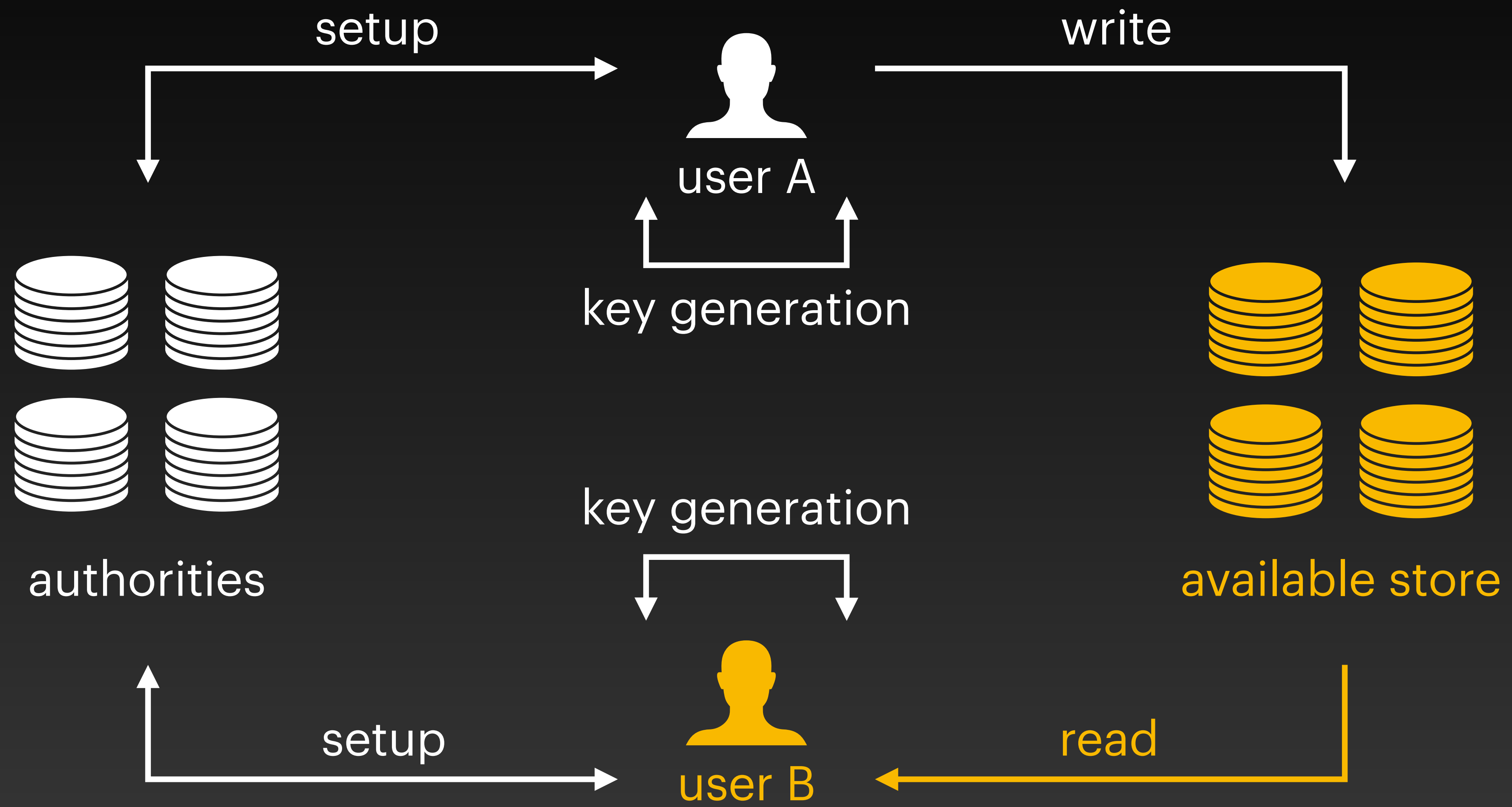












What about blockchains?



Signal



Twitter



Google



Apple



zkLogin



user



Signal



Twitter



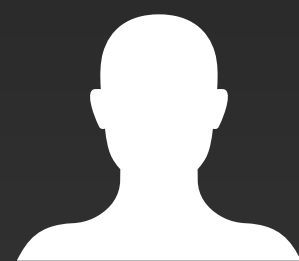
Google



Apple



zkLogin



user



blockchain



**Arke
setup**



user



Signal



Twitter



Google



Apple



zkLogin



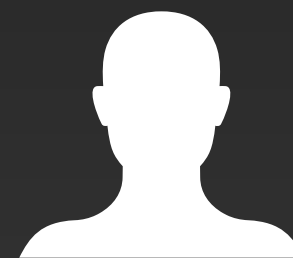
user



blockchain



**Arke
setup**



user

**derive
shared key**



user

More than private chats

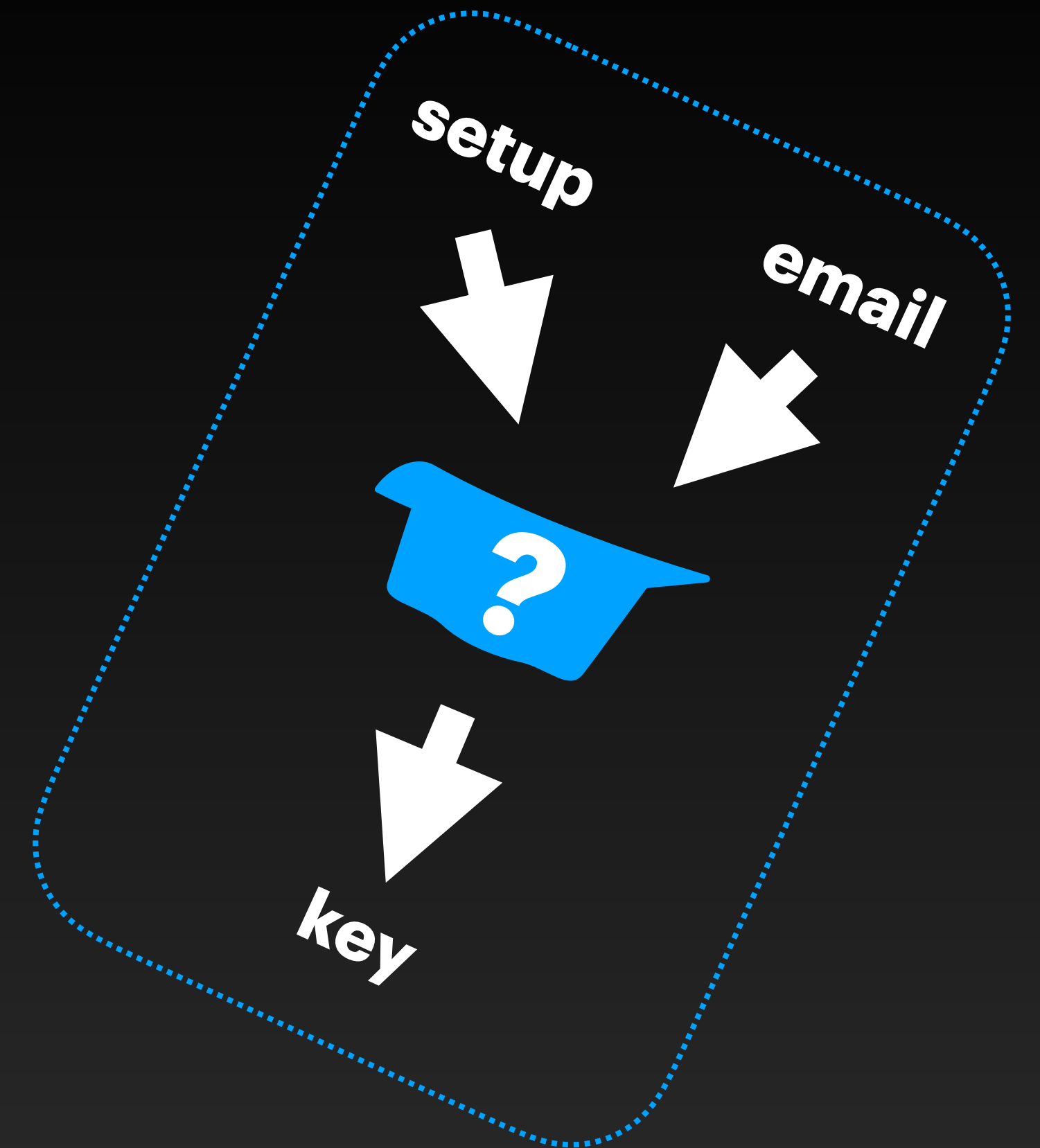
- Decentralised messaging
- Bootstrap multi-user gaming sessions
- Airdrops / payments even before recipient has an account

Paper



alberto@mystenlabs.com

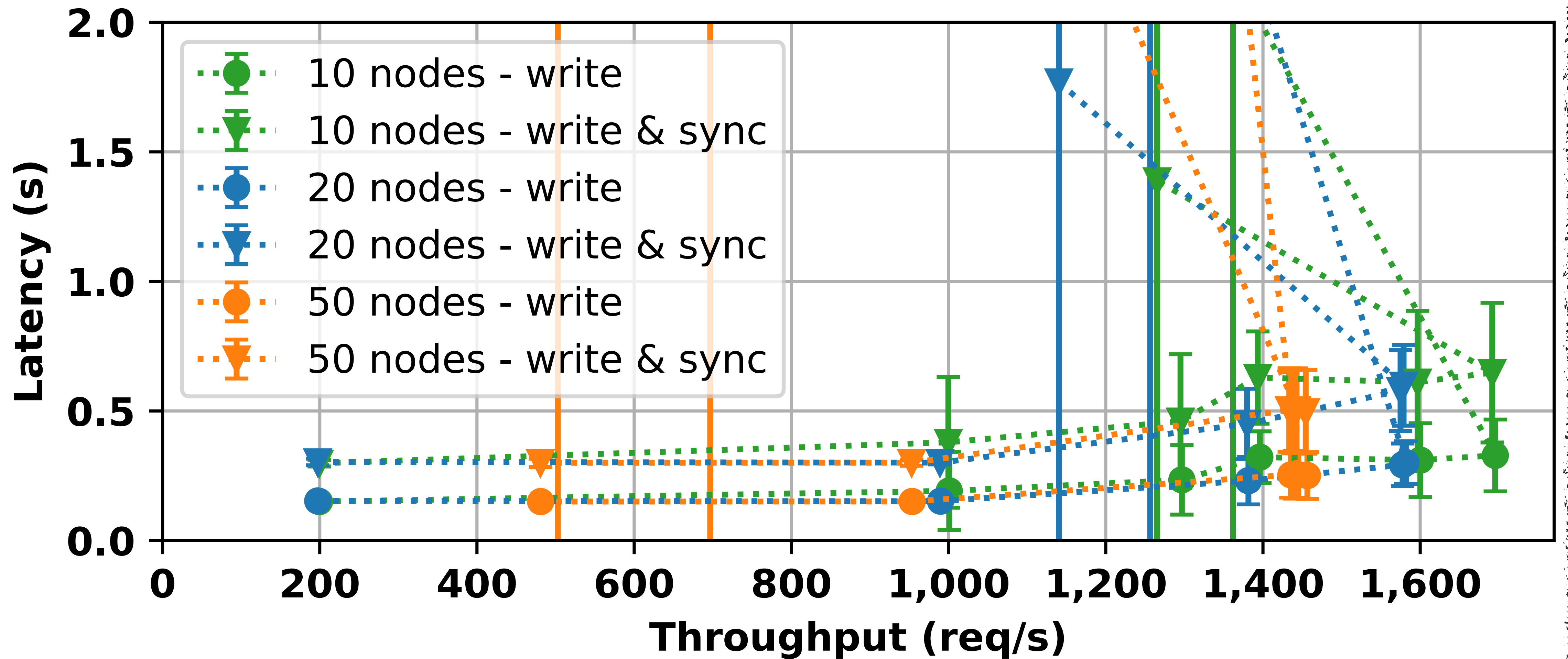
Paper



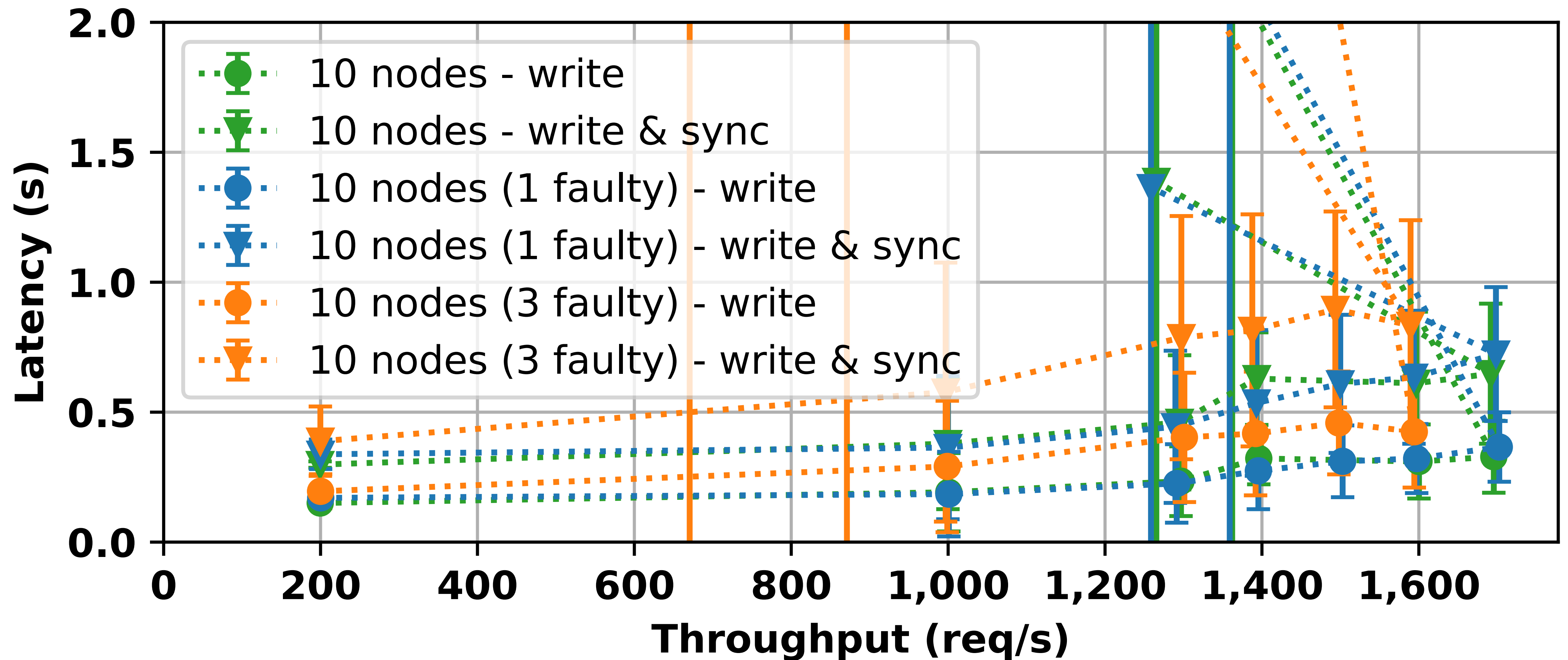
alberto@mystenlabs.com

L-Graphs

Performance

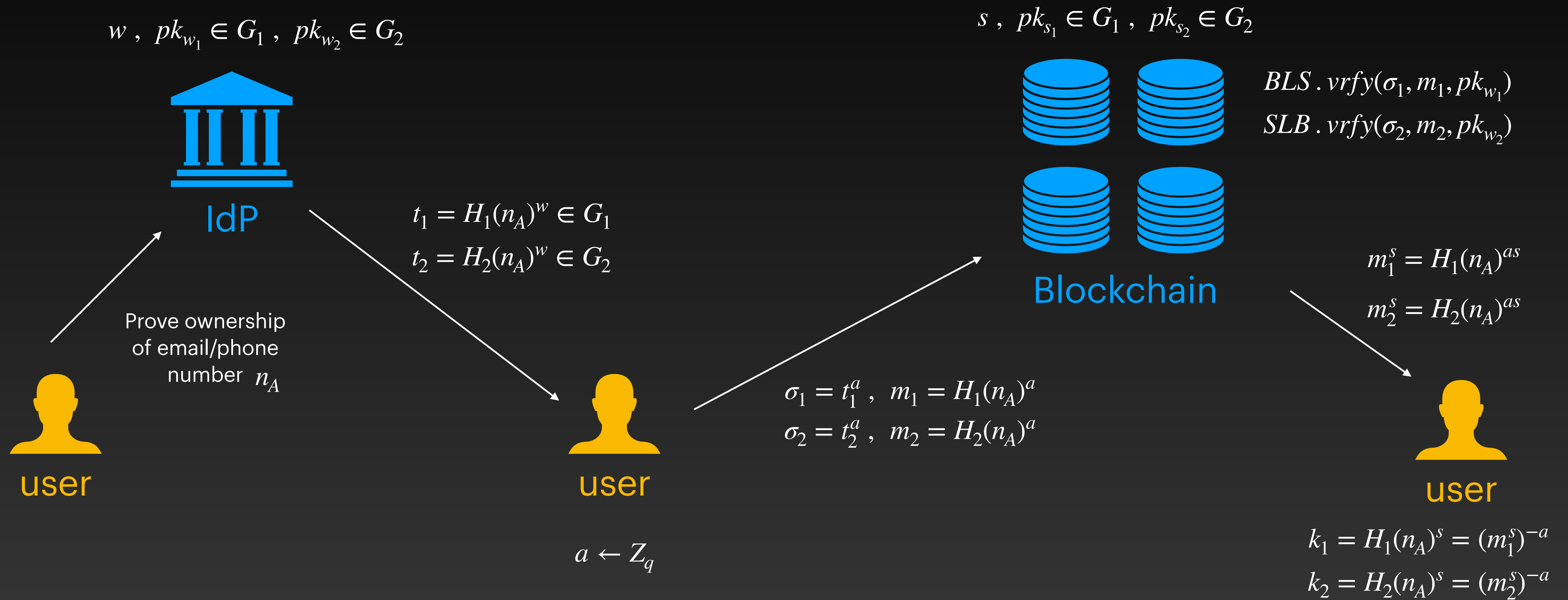


Performance



The Crypto

Setup



Key Derivation



user A

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

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

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

$$key = g_1^{t_{AB}}, \quad t_{AB} = H(s_{AB})$$

$$val = c_{AB} = AEAD_k(pk_A)$$



user B

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

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

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

$$key = g_1^{t_{BA}}, \quad t_{BA} = H(s_{BA})$$

$$val = c_{BA} = AEAD_k(pk_B)$$

Sui is special



user A

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

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

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

$$key = g_1^{t_{AB}}, \quad t_{AB} = H(s_{AB})$$

$$val = c_{AB} = AEAD_k(addr_A)$$

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