
encrypted mempools

1) *"what"* basics
2) *"why"* motivation
3) *"how"* metadata

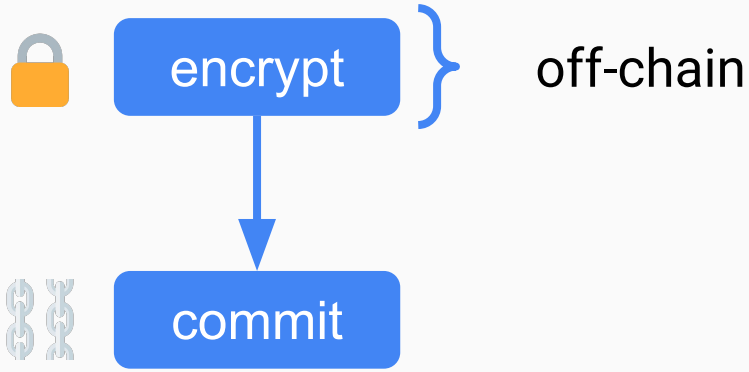
1) *"what"* basics

2) *"why"* motivation

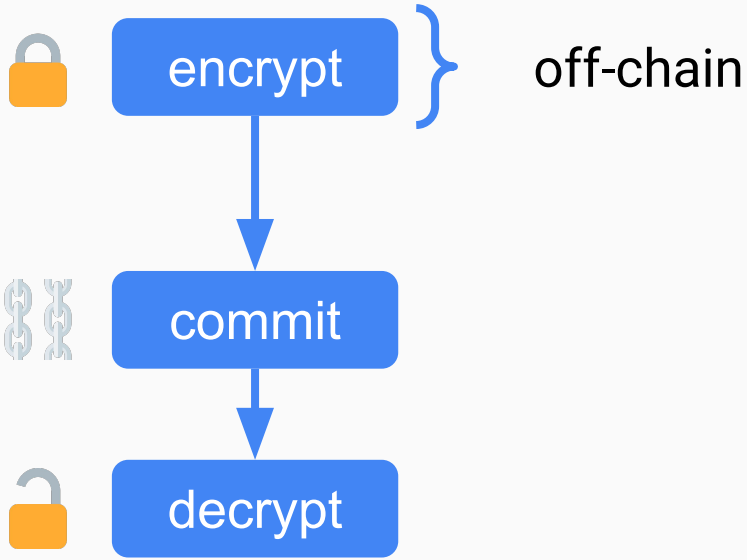
3) *"how"* metadata

simple framework

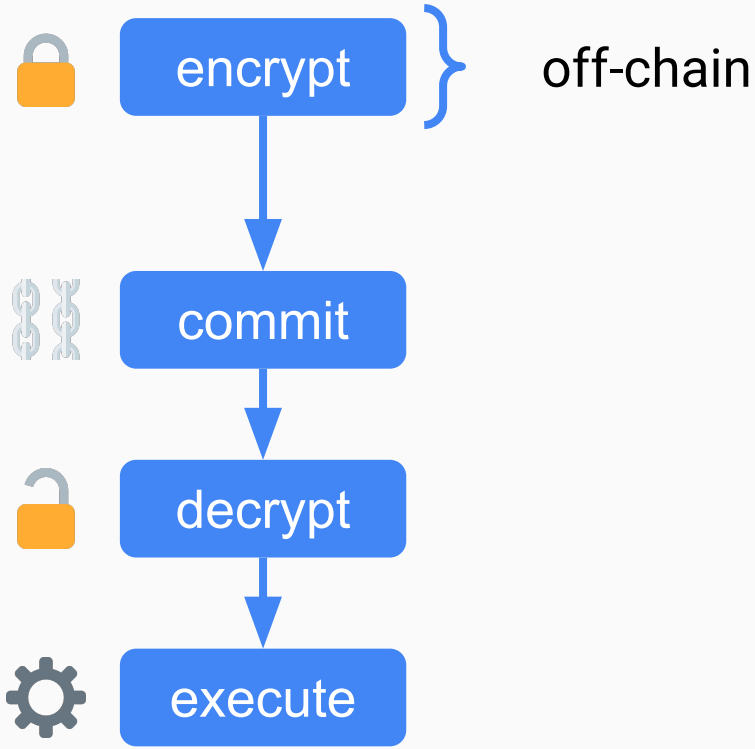
simple framework



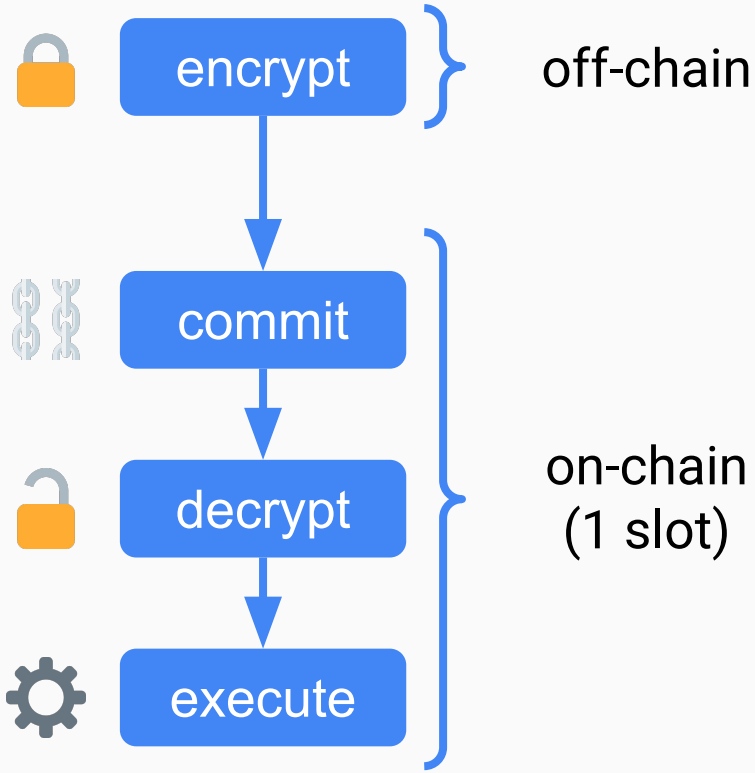
simple framework



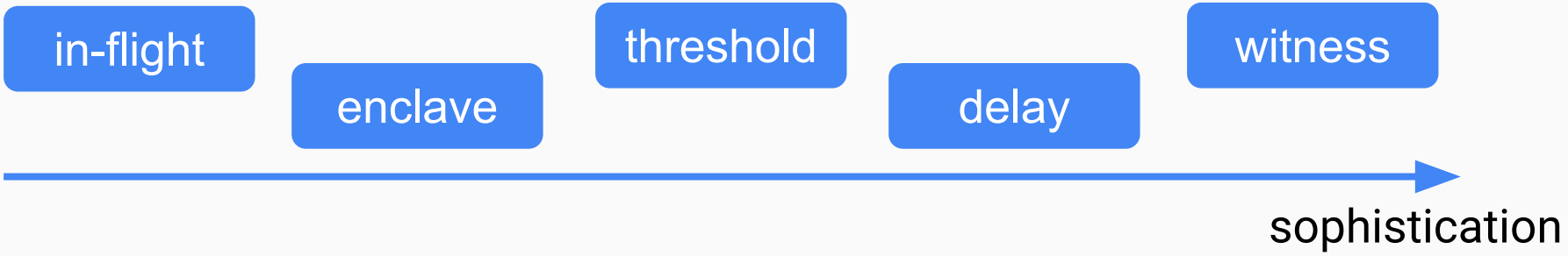
simple framework



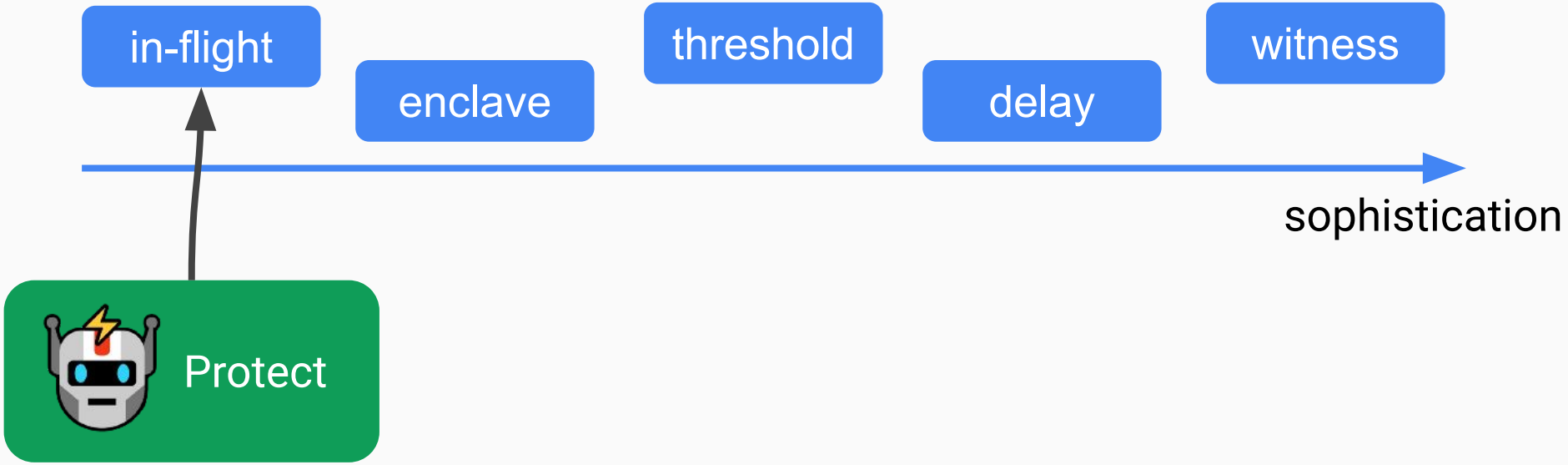
simple framework



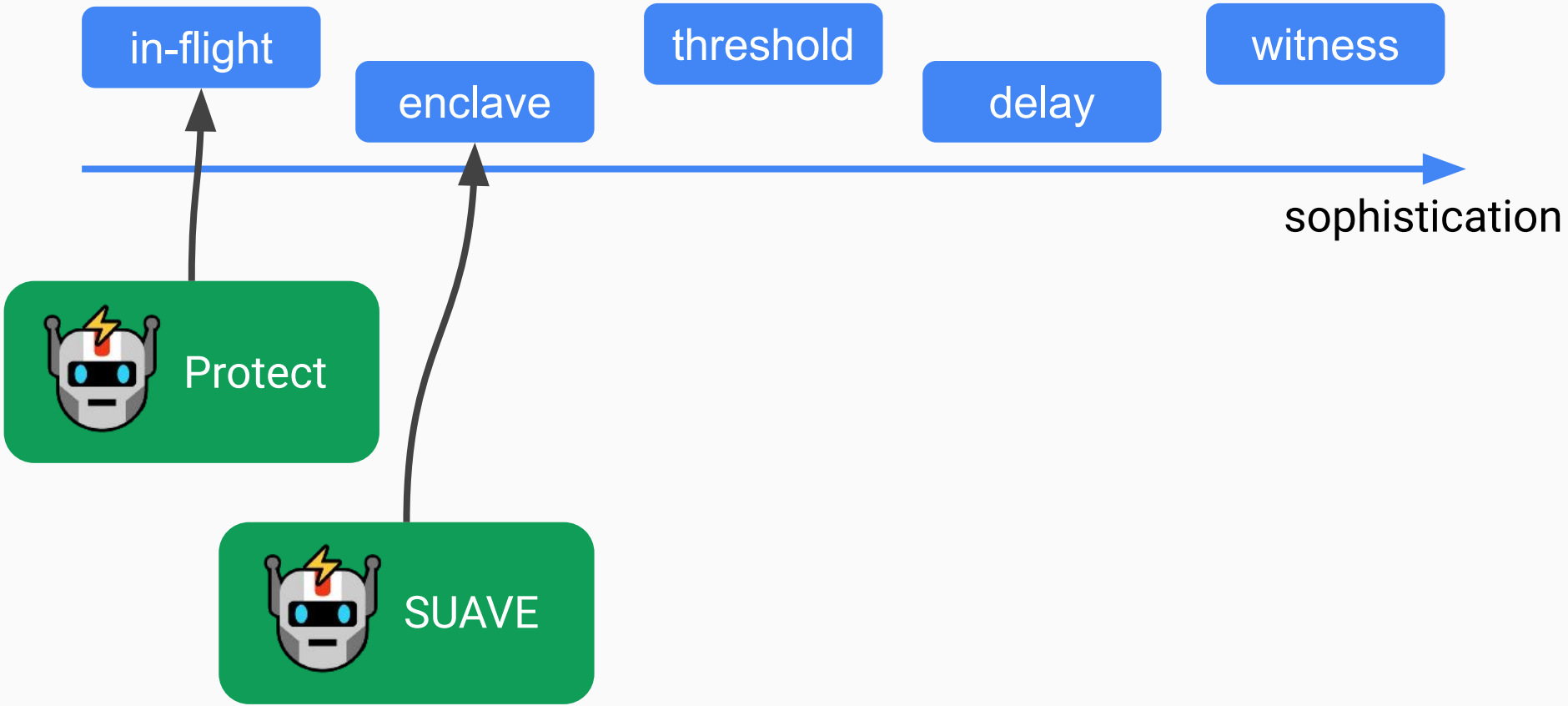
guaranteed decryption



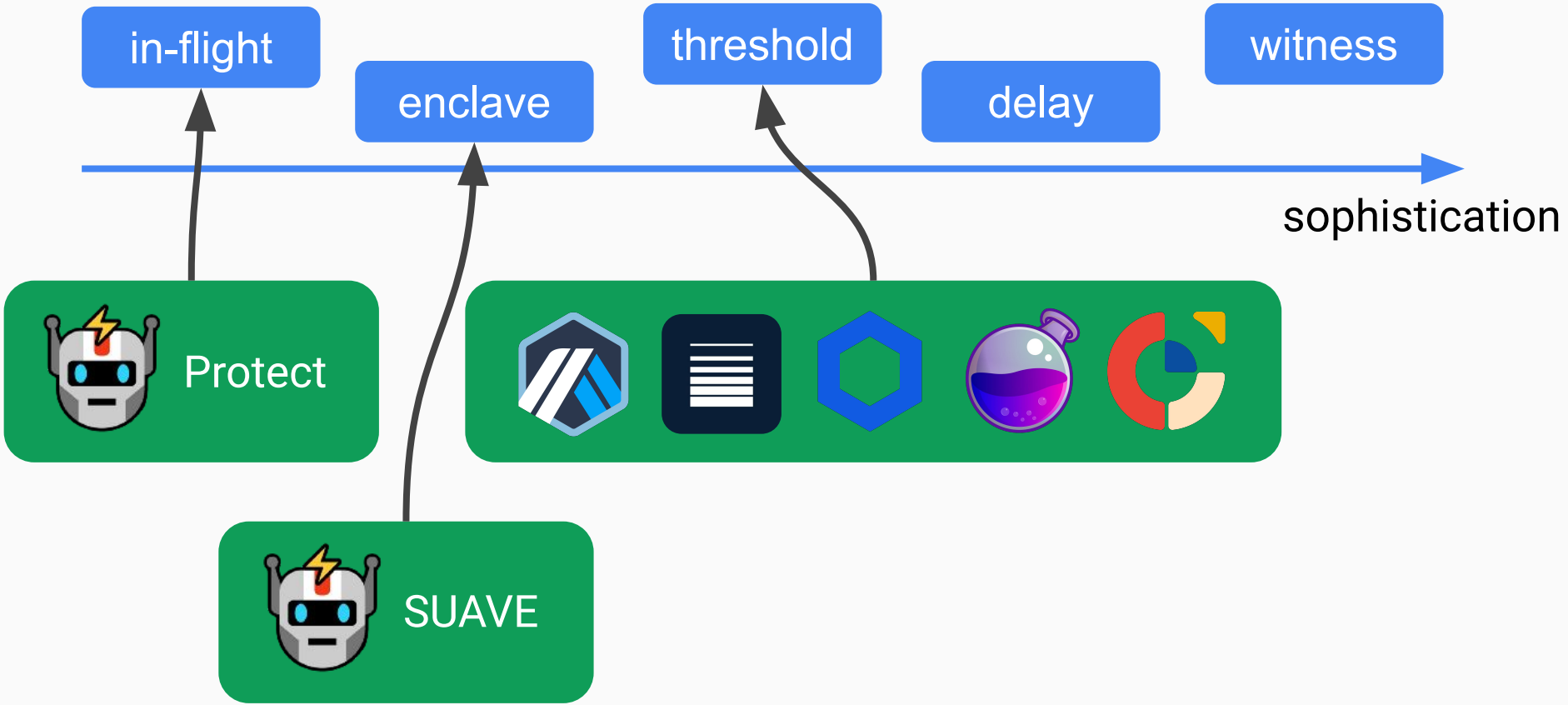
guaranteed decryption



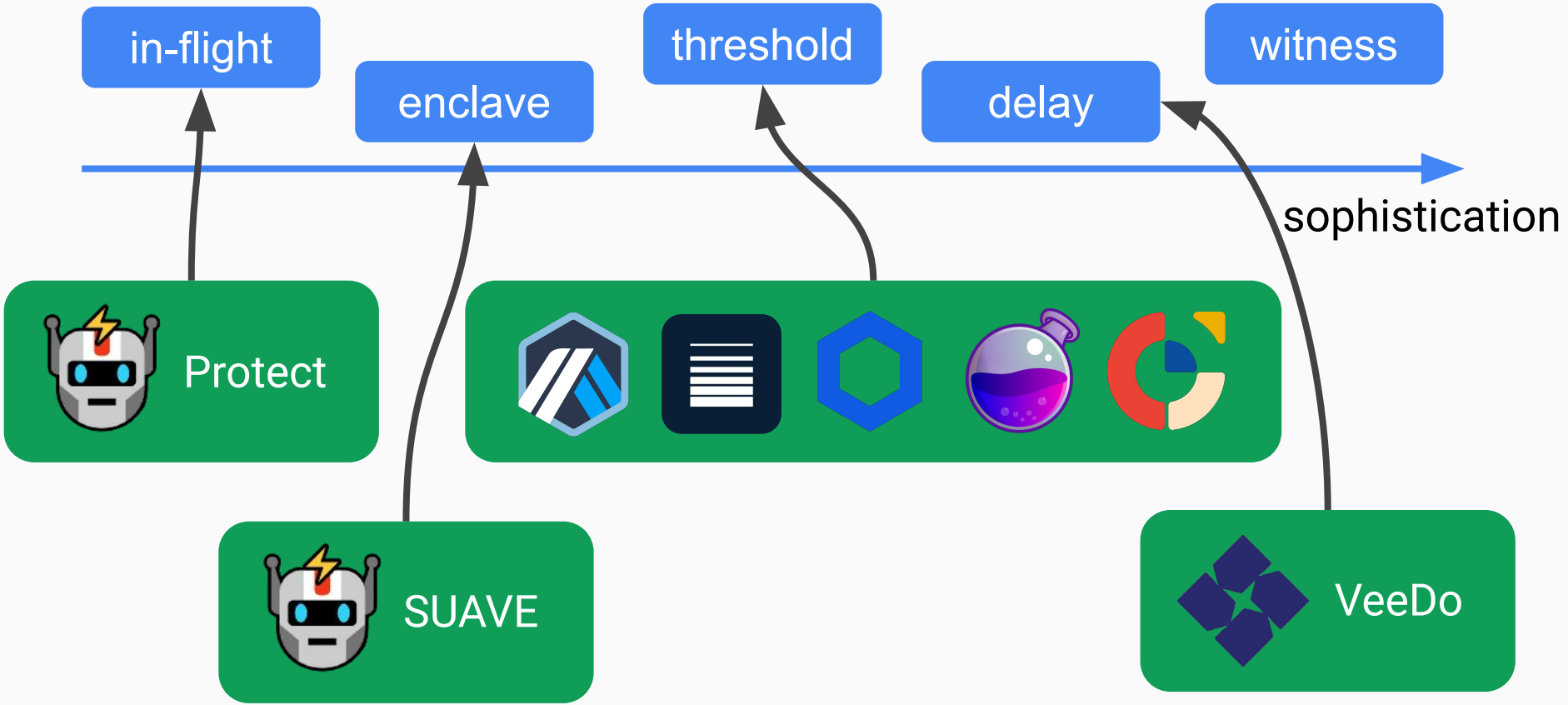
guaranteed decryption



guaranteed decryption



guaranteed decryption



readiness

	in-flight	enclave	threshold
ready?	✅	✅	✅

readiness

	in-flight	enclave	threshold	delay
ready?	✅	✅	✅	➡ SOON

readiness

	in-flight	enclave	threshold	delay	witness
ready?	✅	✅	✅	➡ SOON	

homomorphism

encryption($\mathbf{m_1}$), encryption($\mathbf{m_2}$)

homomorphism

encryption($\mathbf{m_1}$), encryption($\mathbf{m_2}$)

encryption($\mathbf{f}$($\mathbf{m_1}$, $\mathbf{m_2}$))

homomorphism

encryption($\mathbf{m_1}$), encryption($\mathbf{m_2}$)

encryption($\mathbf{f}(\mathbf{m_1}, \mathbf{m_2})$)

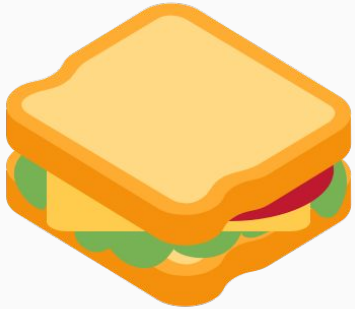
	in-flight	enclave	threshold	delay	witness
ready?	✅	✅	✅	➡ SOON	🙈
🪄	✅	✅	➡ SOON	➡ SOON ?	🙈

1) *"what"* basics

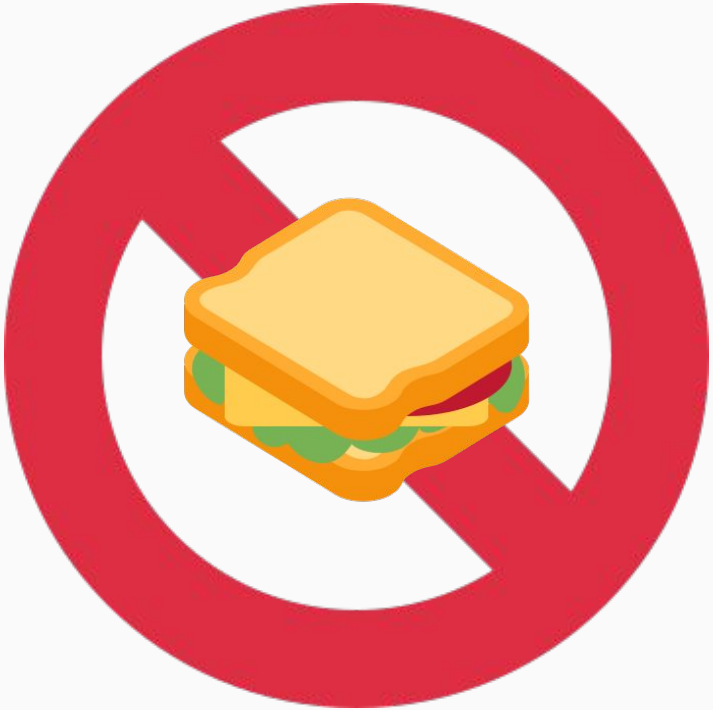
2) *"why"* motivation

3) *"how"* metadata

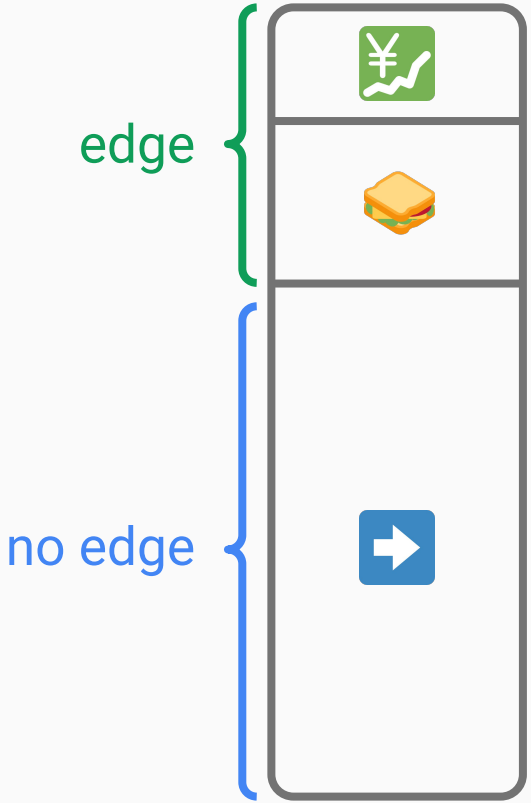
problem 1—frontrunning



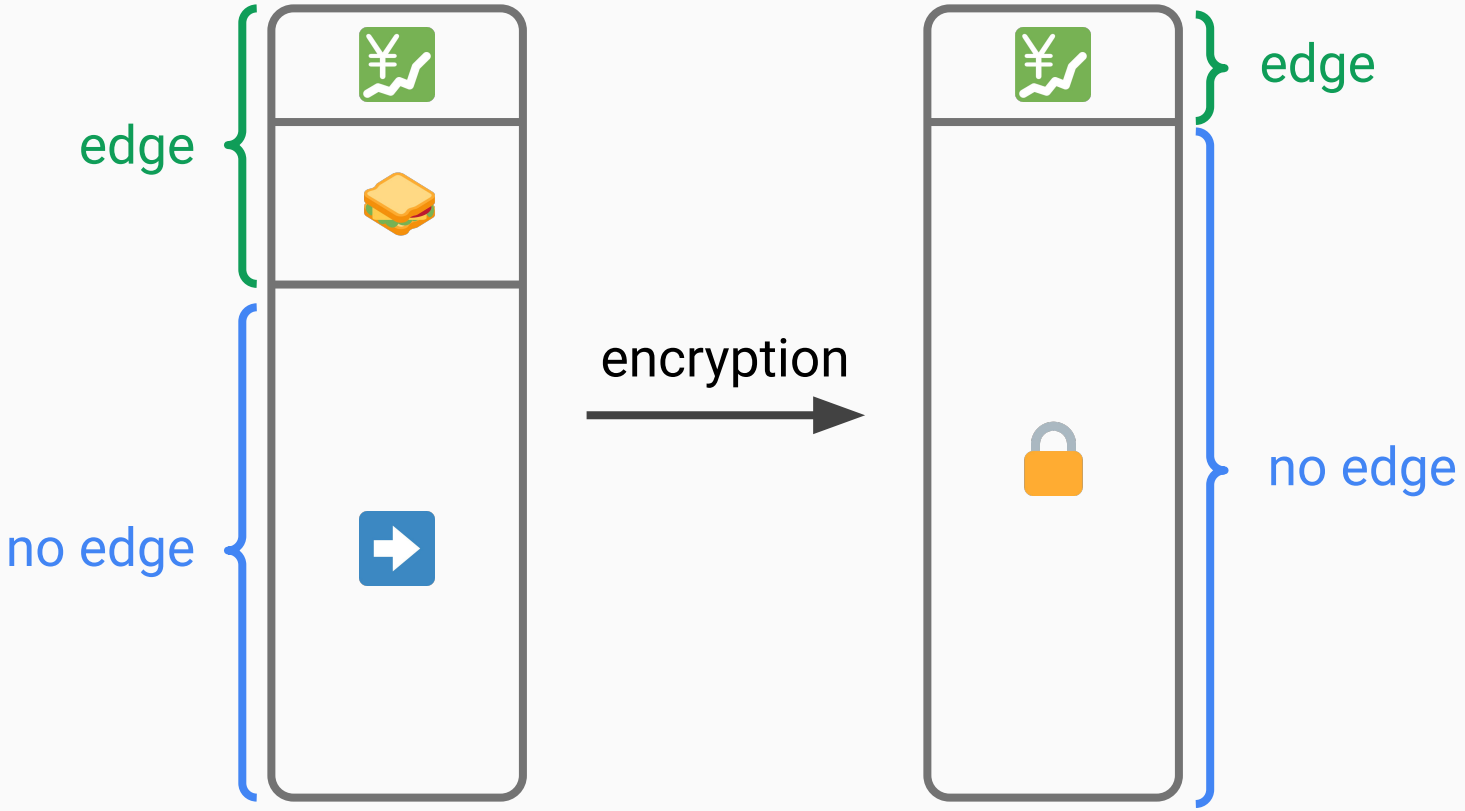
problem 1—frontrunning



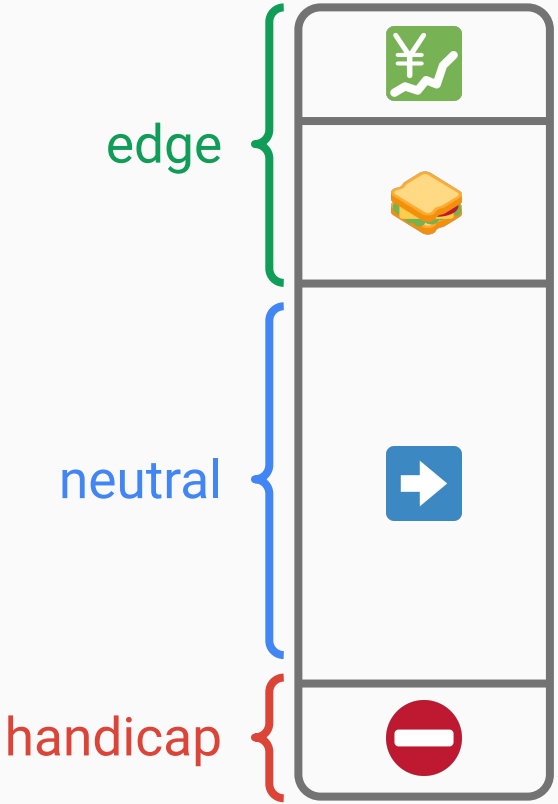
problem 1b—centralisation



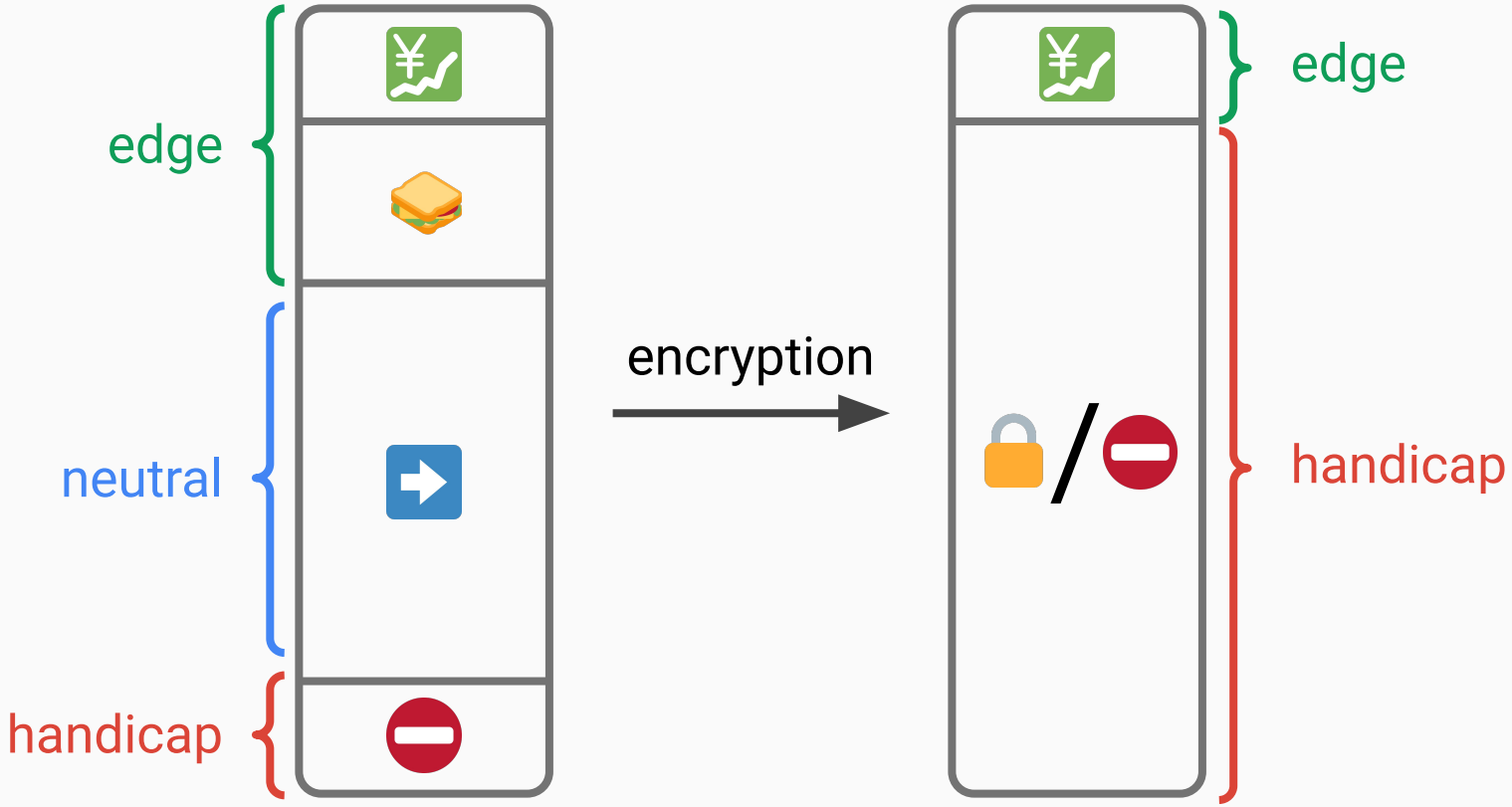
problem 1b—centralisation



problem 2—censorship



problem 2—censorship



1) *"what"* basics

2) *"why"* motivation

3) *"how"* metadata

transaction metadata

IP address

size

sender

tip

transaction metadata

 IP address

 gas price

 size

 gas limit

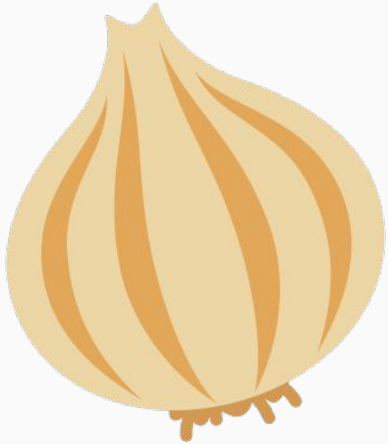
 sender

 nonce

 tip

 signature

IP address



private broadcast

(e.g. Tor)

signature 👌

public input: tx ciphertext

private witness: tx plaintext

zk statement: tx ciphertext valid

signature 👌

public input: tx ciphertext + state root

private witness: tx plaintext + sender pubkey Merkle proof

zk statement: tx ciphertext valid + Merkle proof valid

signature 👌

public input: tx ciphertext + state root

private witness: tx plaintext + sender pubkey Merkle proof

zk statement: tx ciphertext valid + Merkle proof valid

signature valid

gas payment 👆

public input: tx ciphertext + state root

private witness: tx plaintext + sender balance Merkle proof

zk statement: tx ciphertext valid + Merkle proof valid

gas payment 👆

public input: tx ciphertext + state root

private witness: tx plaintext + sender balance Merkle proof

zk statement: tx ciphertext valid + Merkle proof valid

sufficient sender balance

nonce ☝️

public input: tx ciphertext + state root

private witness: tx plaintext + nonce Merkle proof

zk statement: tx ciphertext valid + Merkle proof valid

new nonce valid

nonce ☝️

public input: tx ciphertext + state root + replay tag

private witness: tx plaintext + nonce Merkle proof

zk statement: tx ciphertext valid + Merkle proof valid

new nonce valid

replay tag = H(nonce, private key)

nonce ☝️

public input: tx ciphertext + state root + replay tag + slot

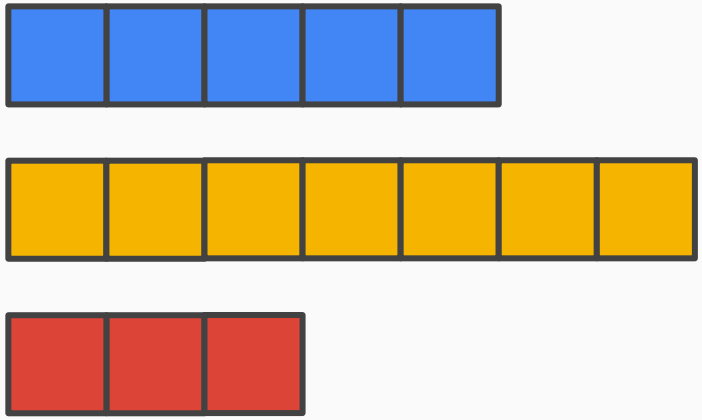
private witness: tx plaintext + nonce Merkle proof

zk statement: tx ciphertext valid + Merkle proof valid

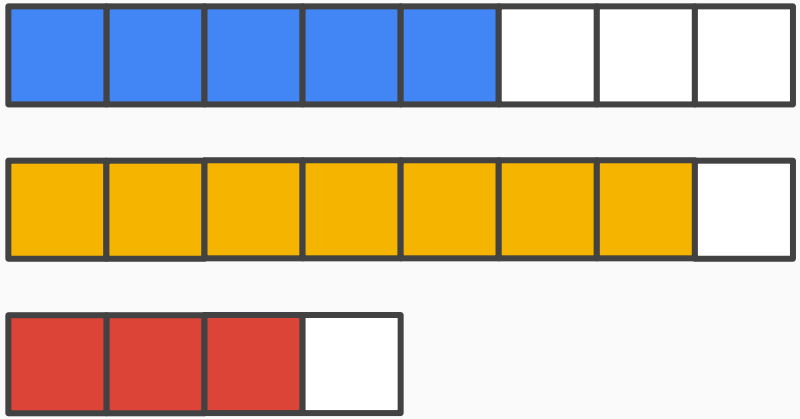
new nonce valid

replay tag = H(nonce, private key, slot)

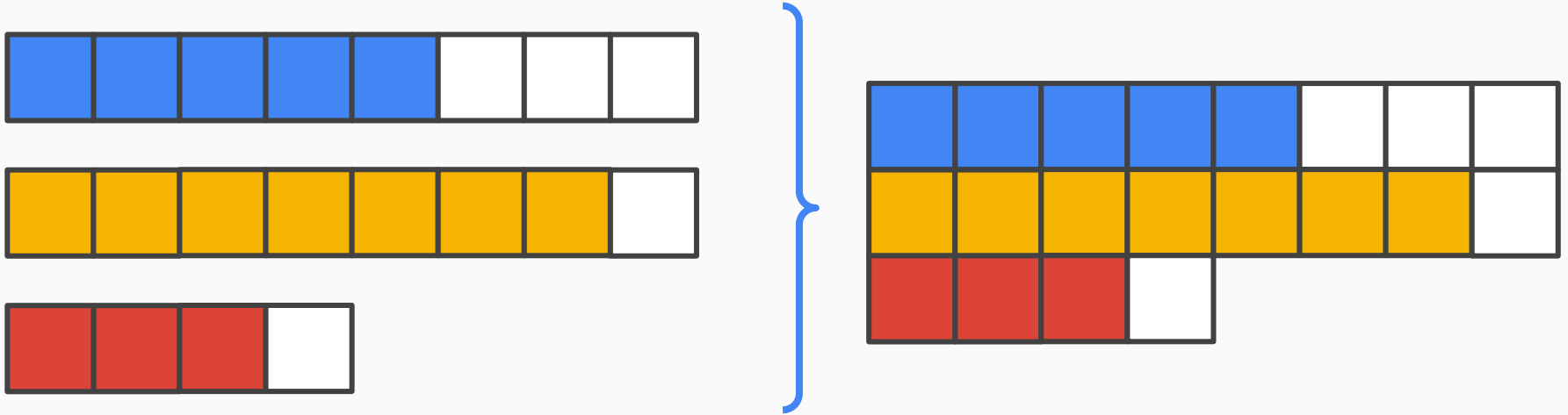
size 👉



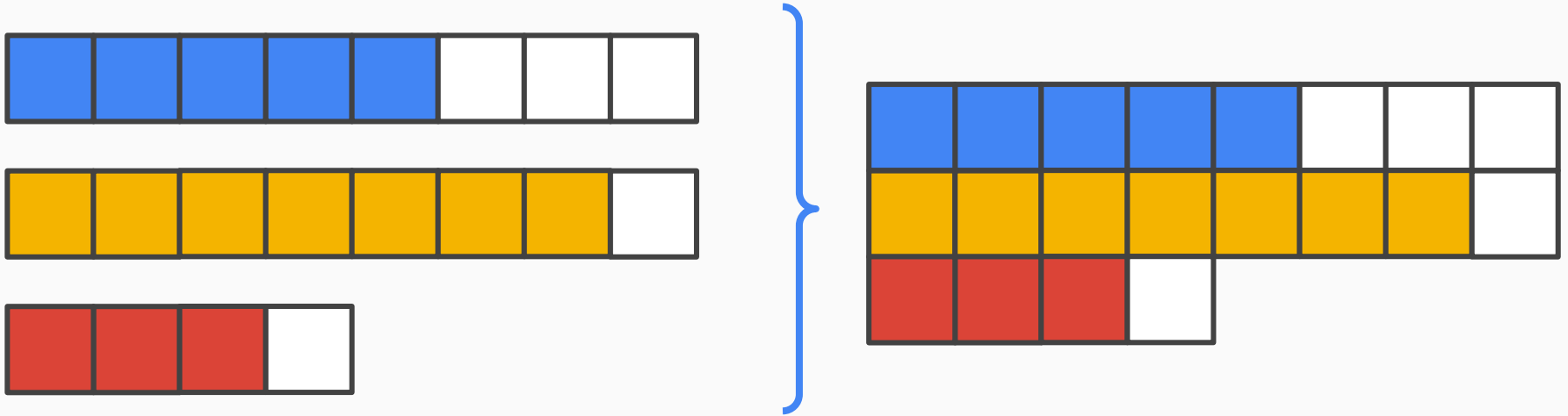
idea 1—pad to power of two



idea 1—pad to power of two



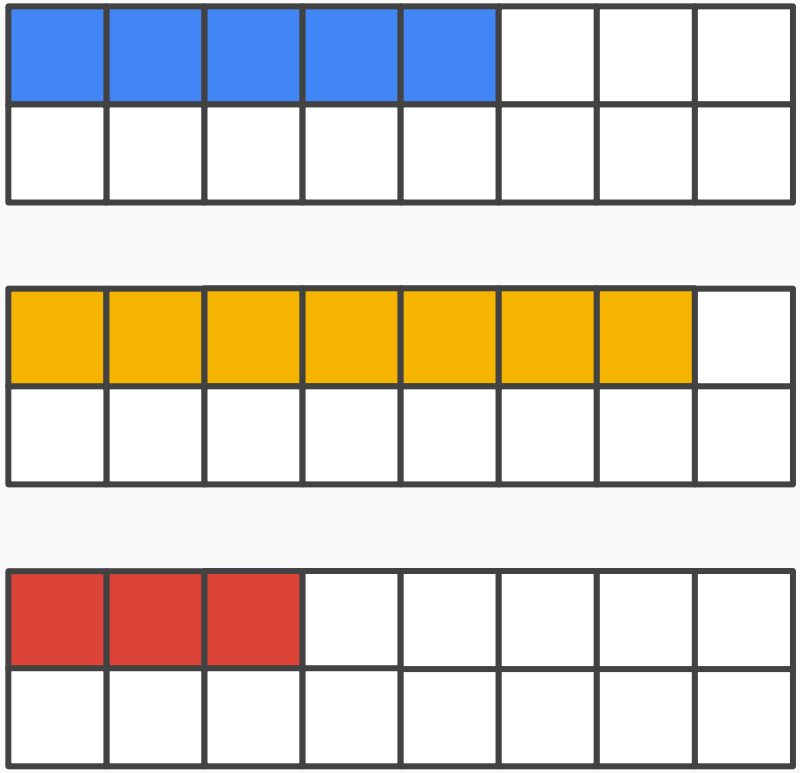
idea 1—pad to power of two



problem 1: imperfect packing

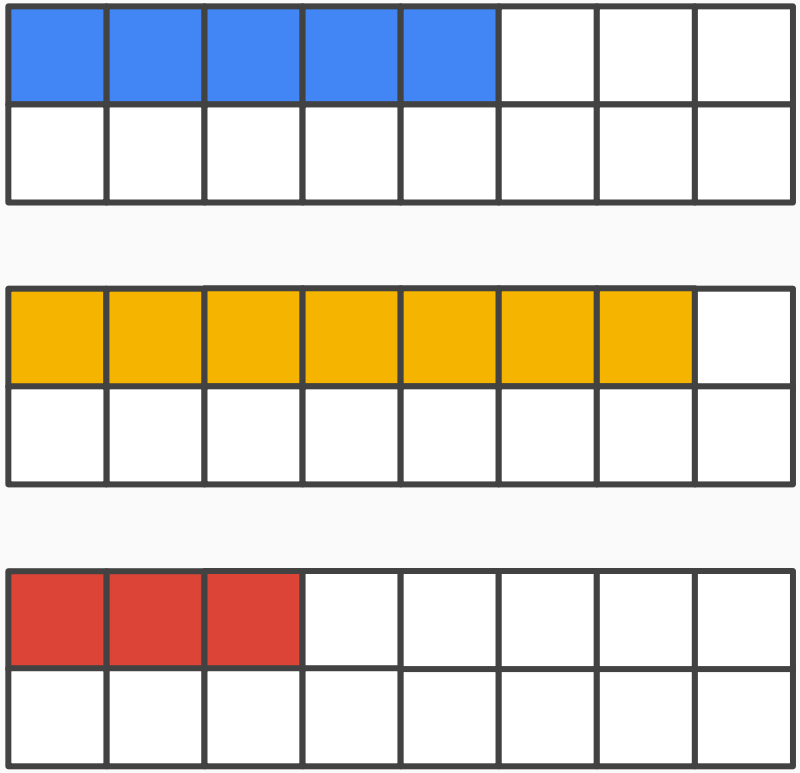
problem 2: imperfect privacy

size 👉

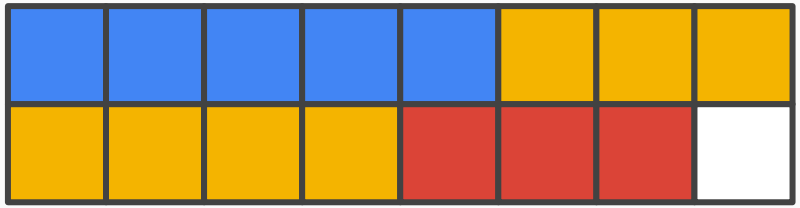


idea 2—use homomorphism

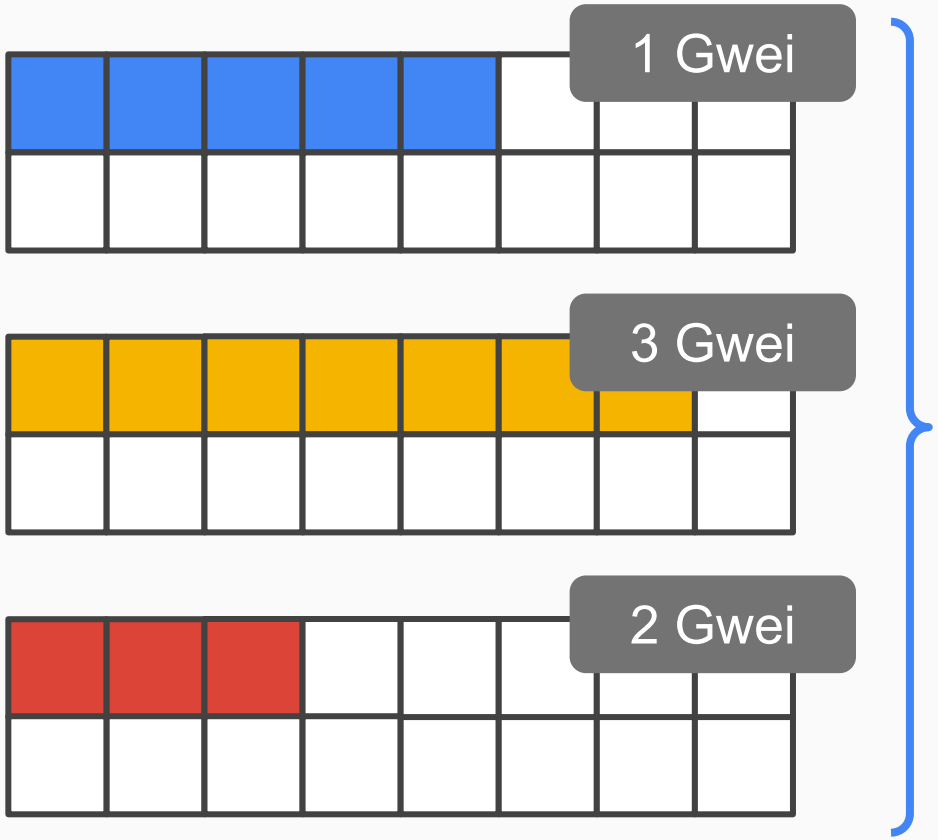
size 



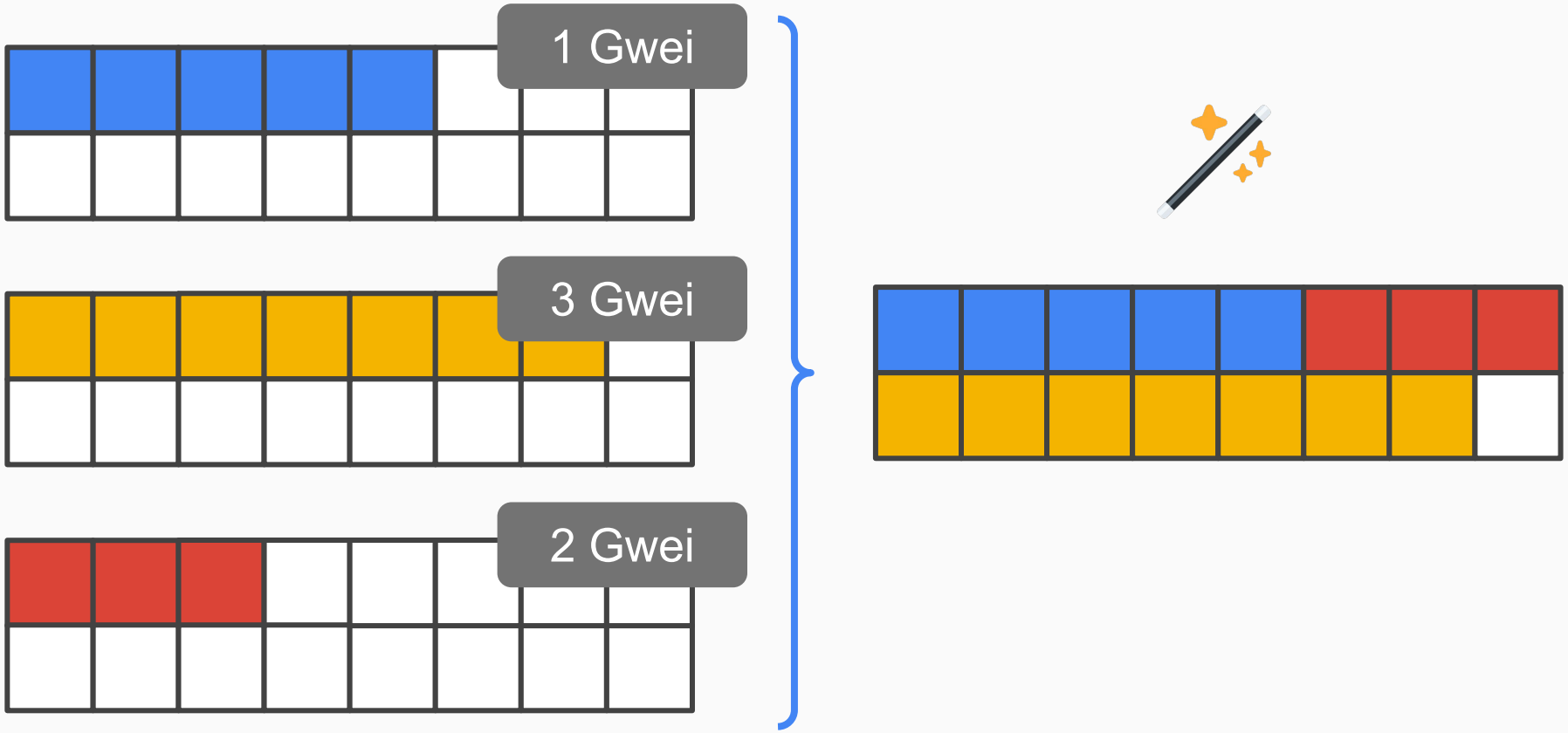
idea 2—use homomorphism



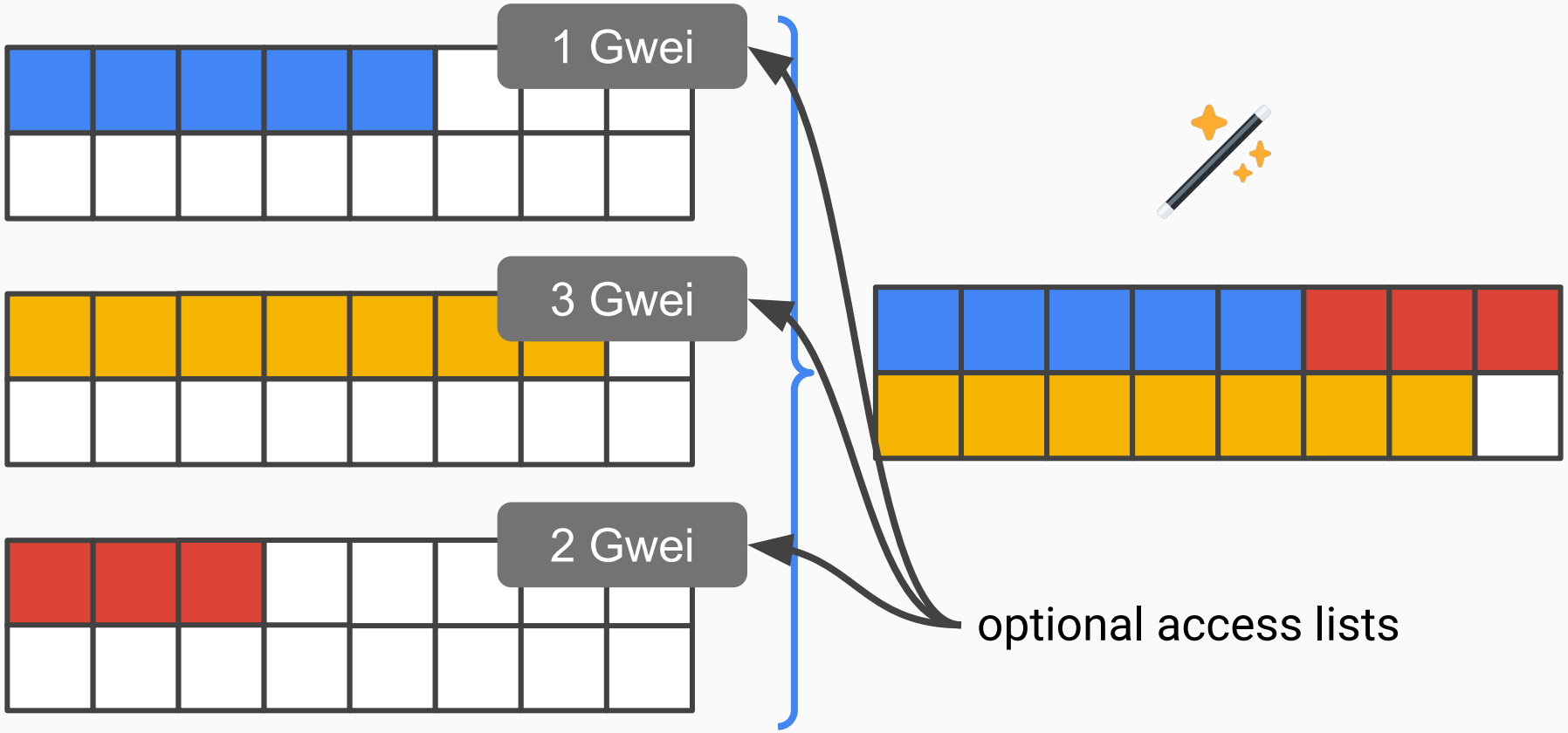
ordering by fee



ordering by fee

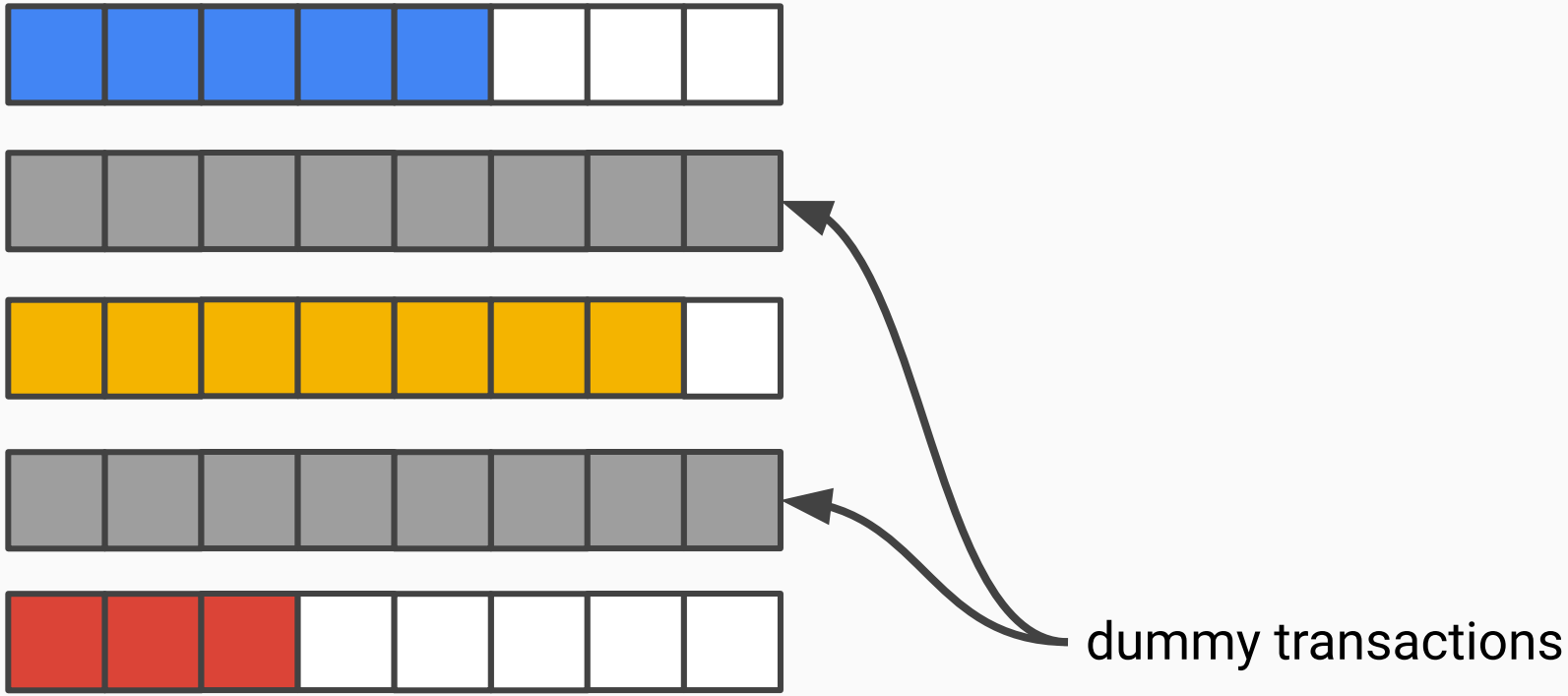


ordering by fee

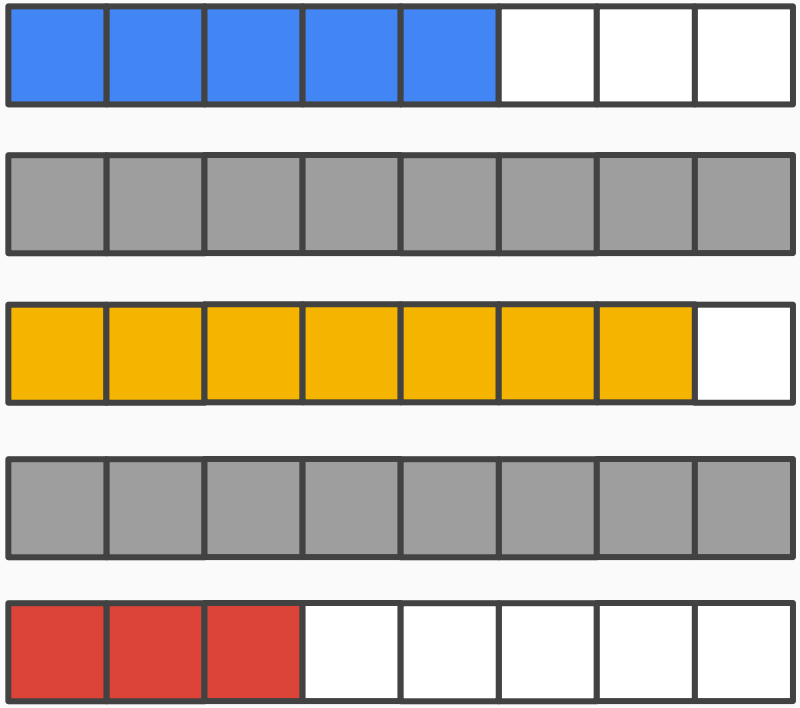


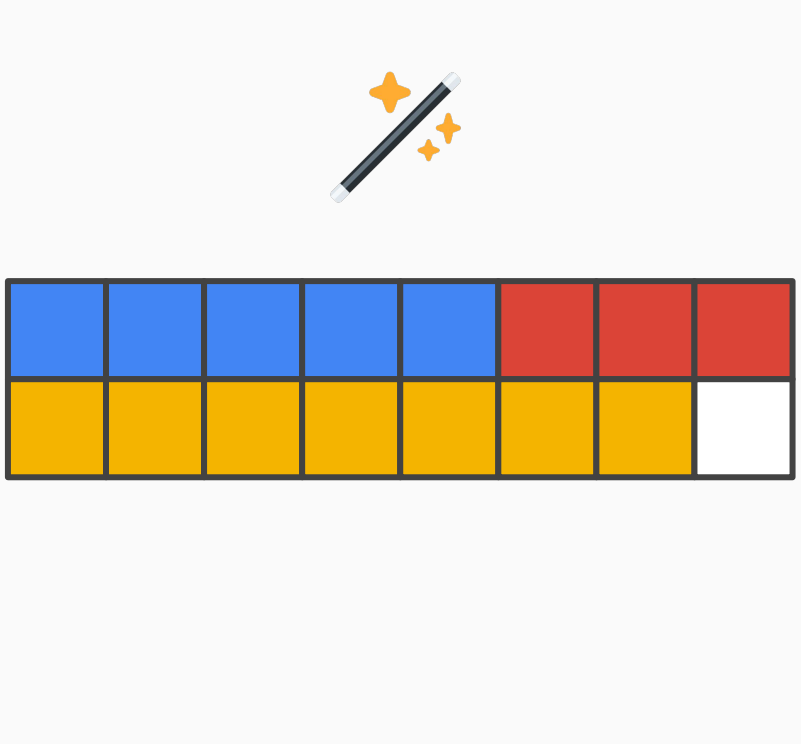
timestamp 👇

timestamp 👇



timestamp 👇





thank you :)

justin@ethereum.org

commitment strength