Brick

Asynchronous Payment Channels

Zeta Avarikioti

Eleftherios Kokoris-Kogias & Roger Wattenhofer

Fundamentals of Channels

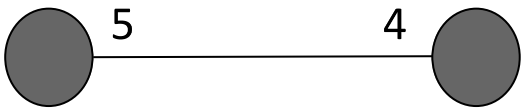
Fundamentals of Channels

Fundamentals of Channels

Funding transaction

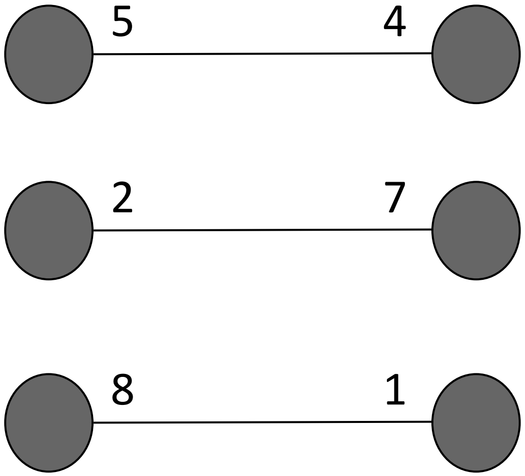
Fundamentals of Channels

Commitment transaction

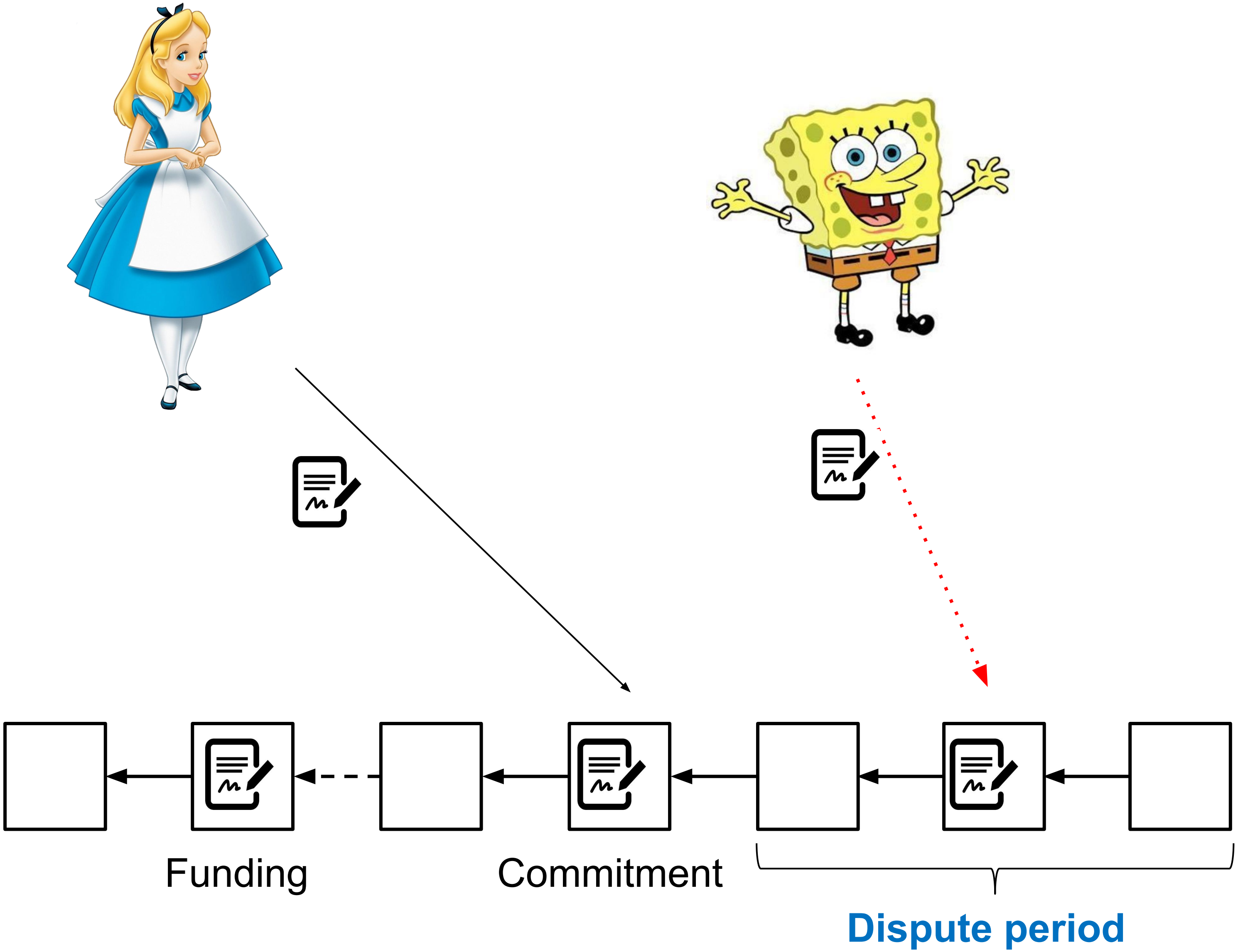


Fundamentals of Channels

Commitment transaction



Fundamentals of Channels

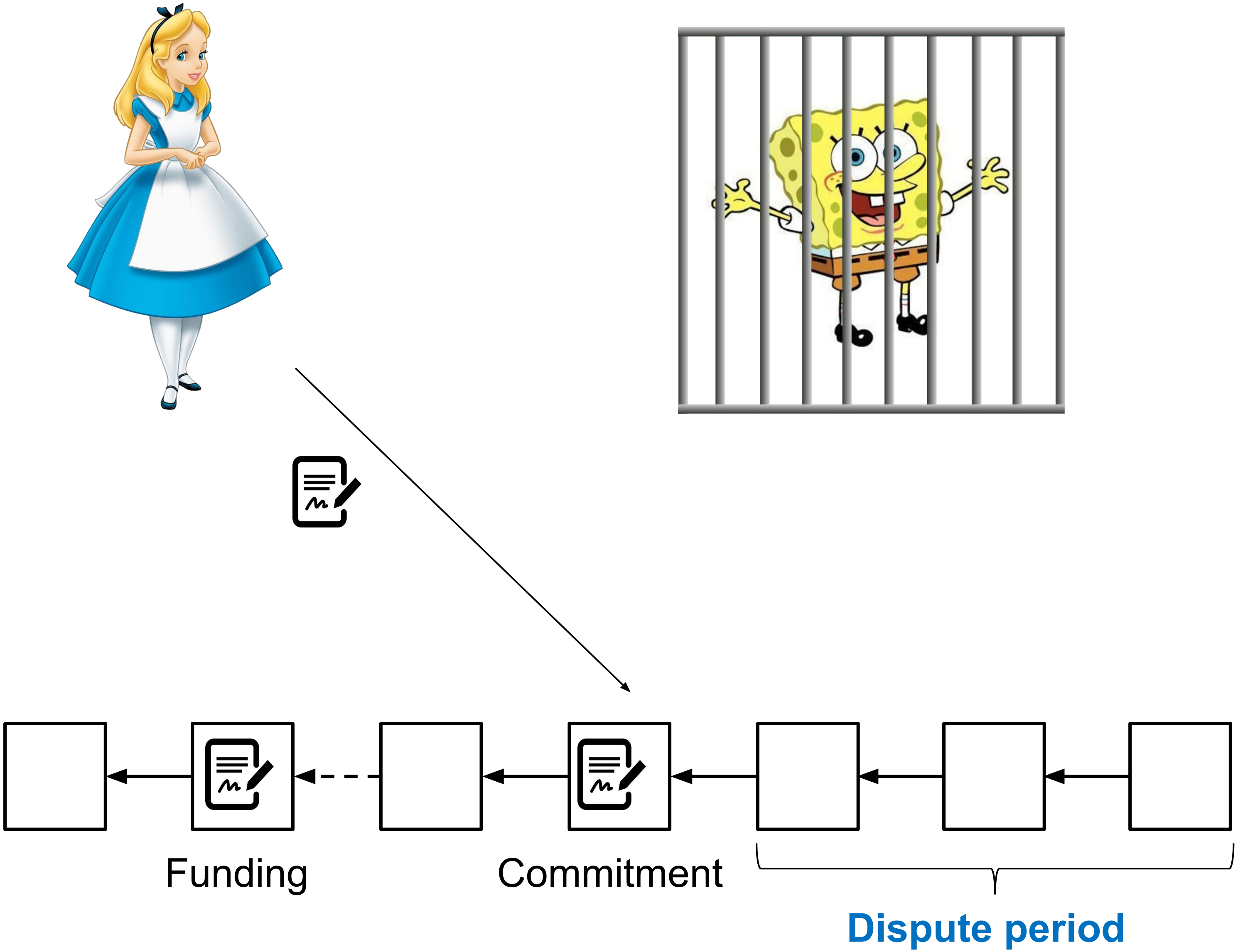


Funding

Commitment

Dispute period

Inactive Counter Party

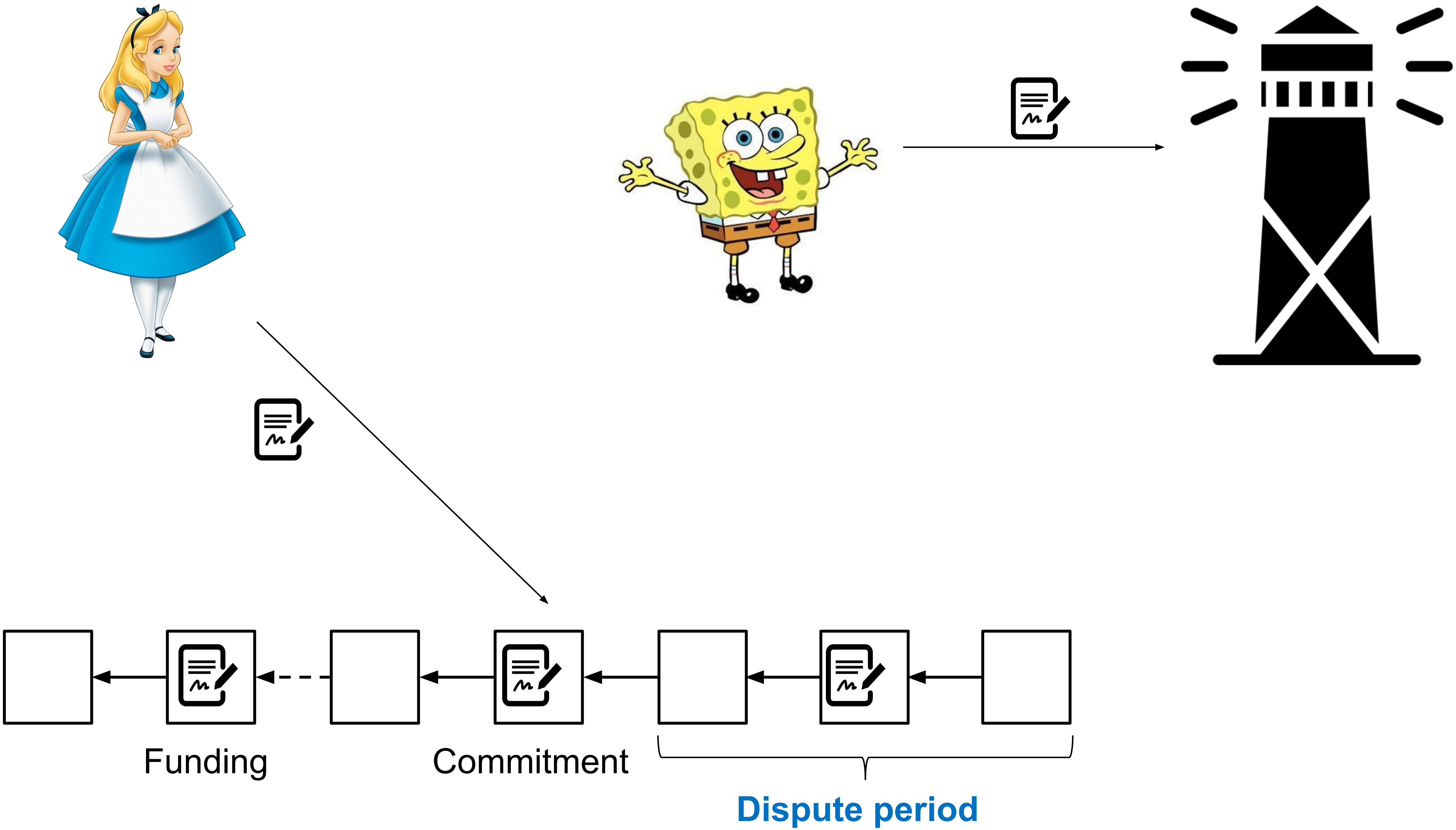


Funding

Commitment

Dispute period

Watchtowers

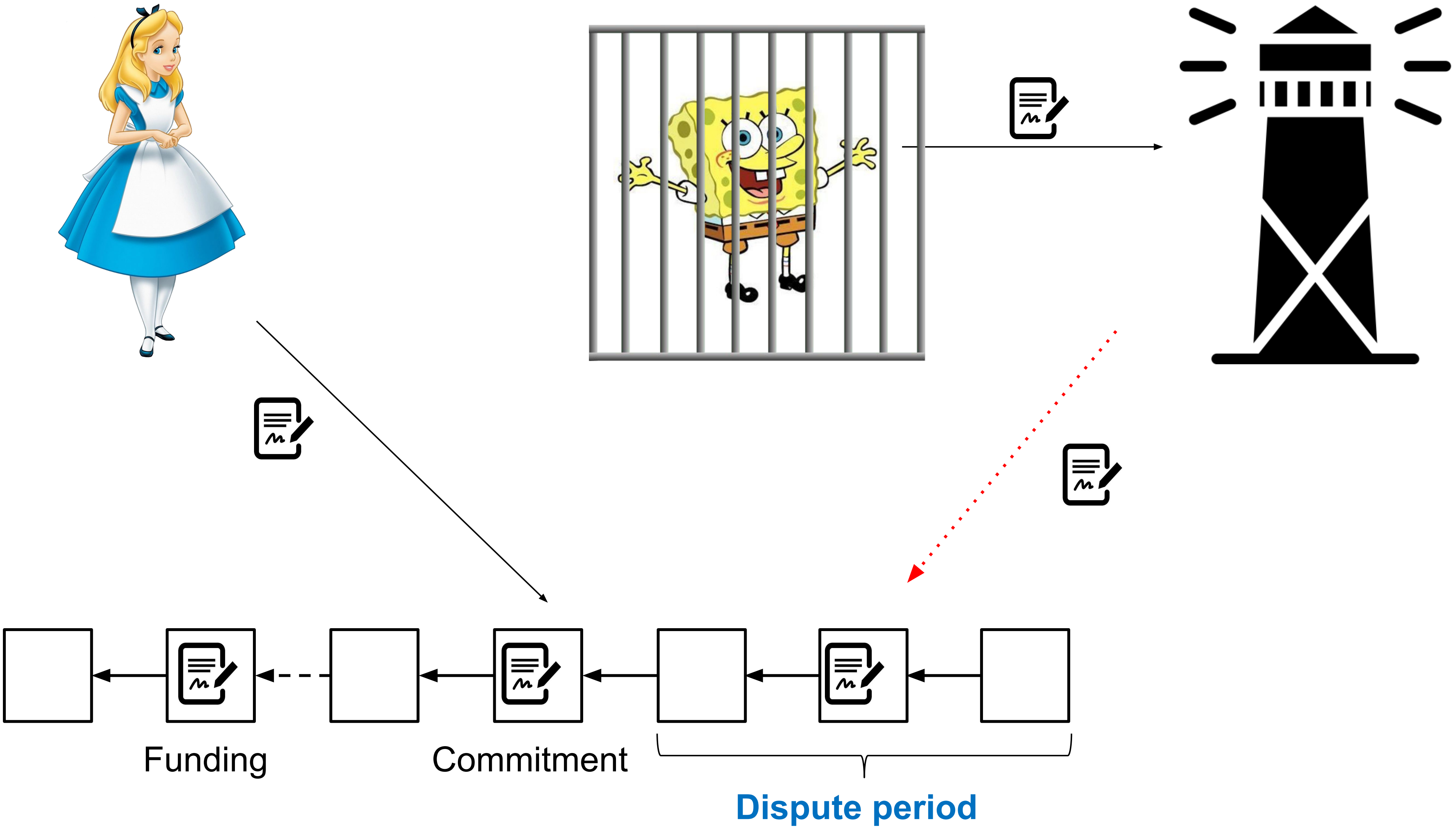


Funding

Commitment

Dispute period

Watchtowers

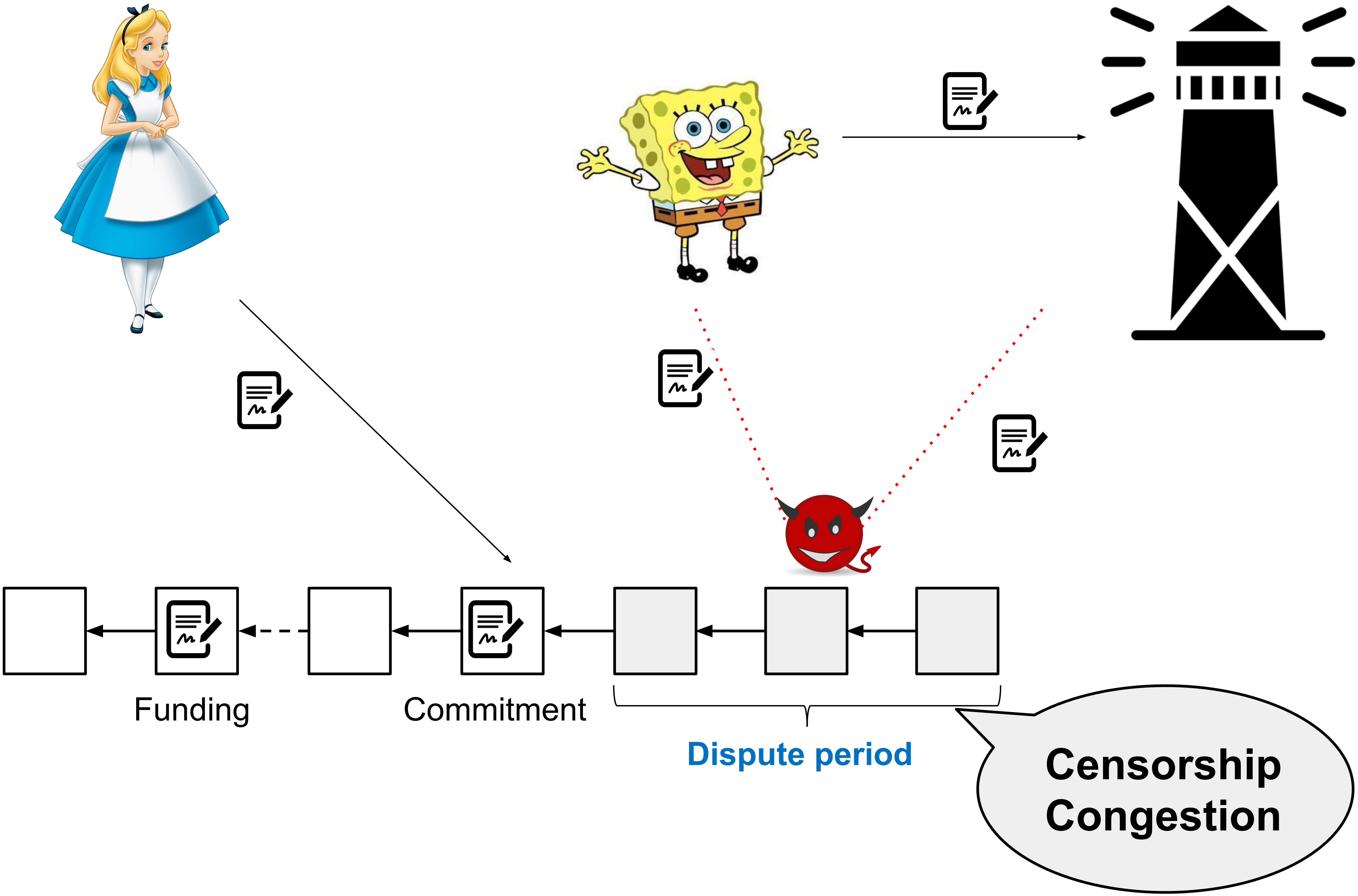


Funding

Commitment

Dispute period

Attack the Liveness of the Blockchain

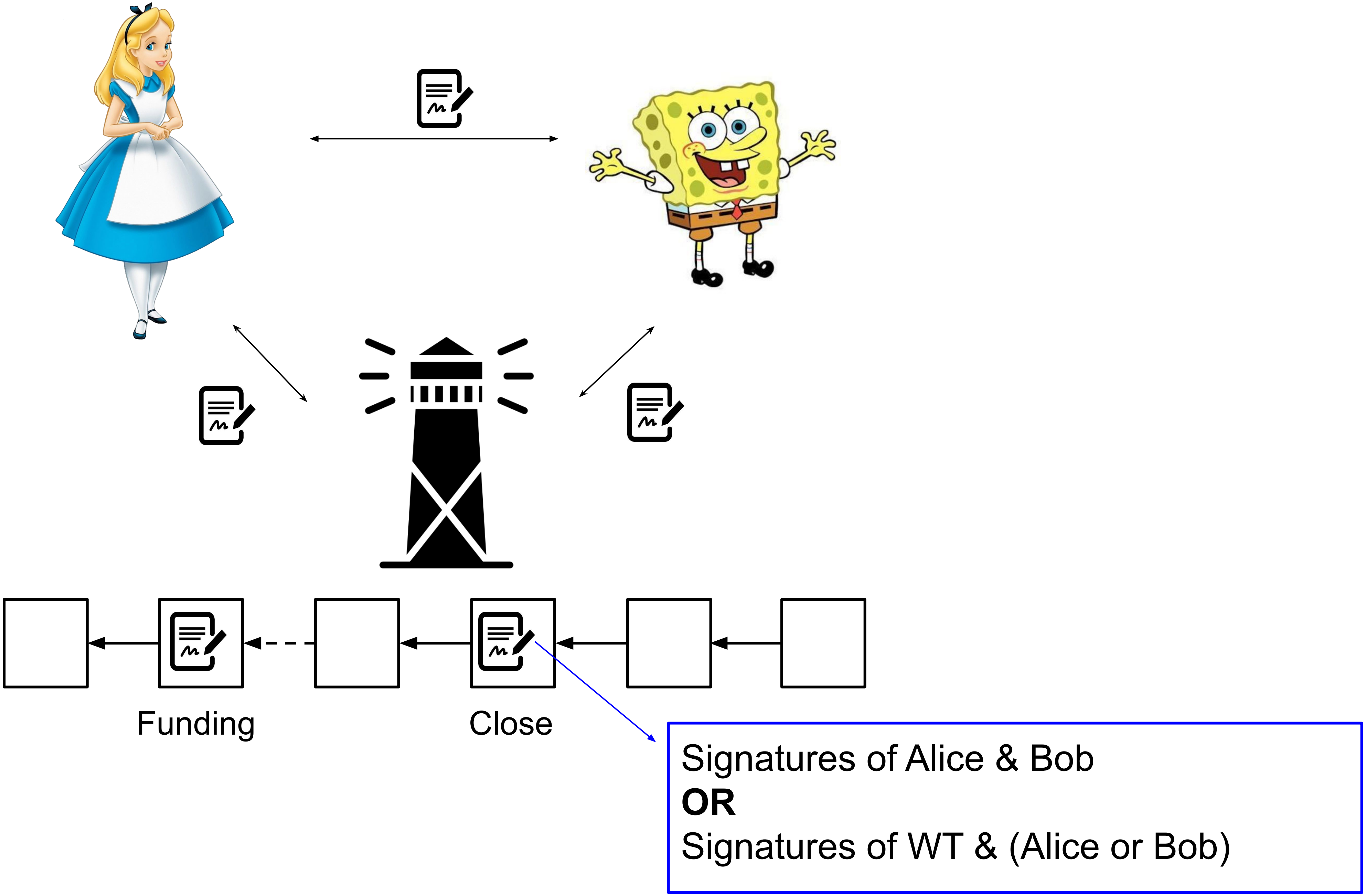


Time = CryptoMoney!

Time = CryptoMoney!

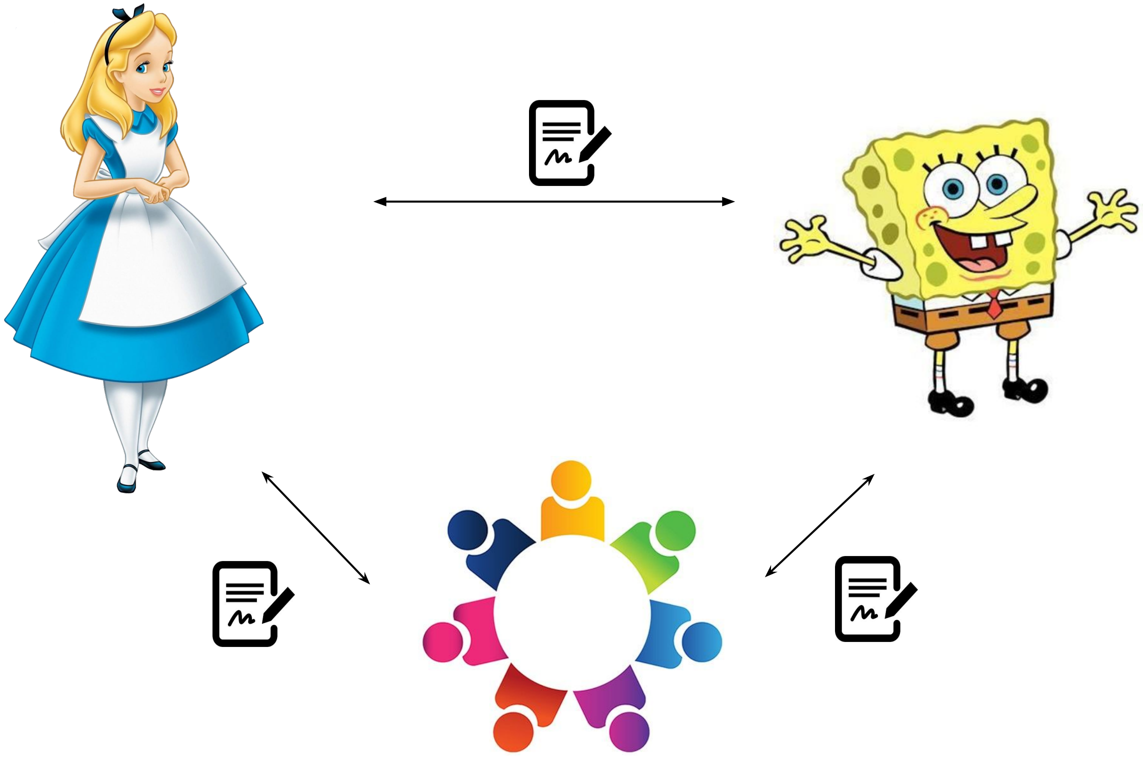
Be proactive, not reactive

Be proactive, not reactive

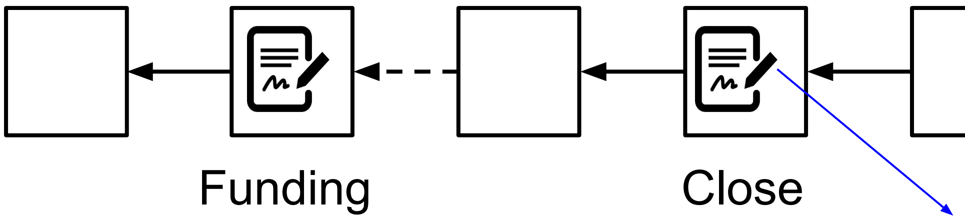


Signatures of Alice & Bob
OR
Signatures of WT & (Alice or Bob)

Watchtower Committee

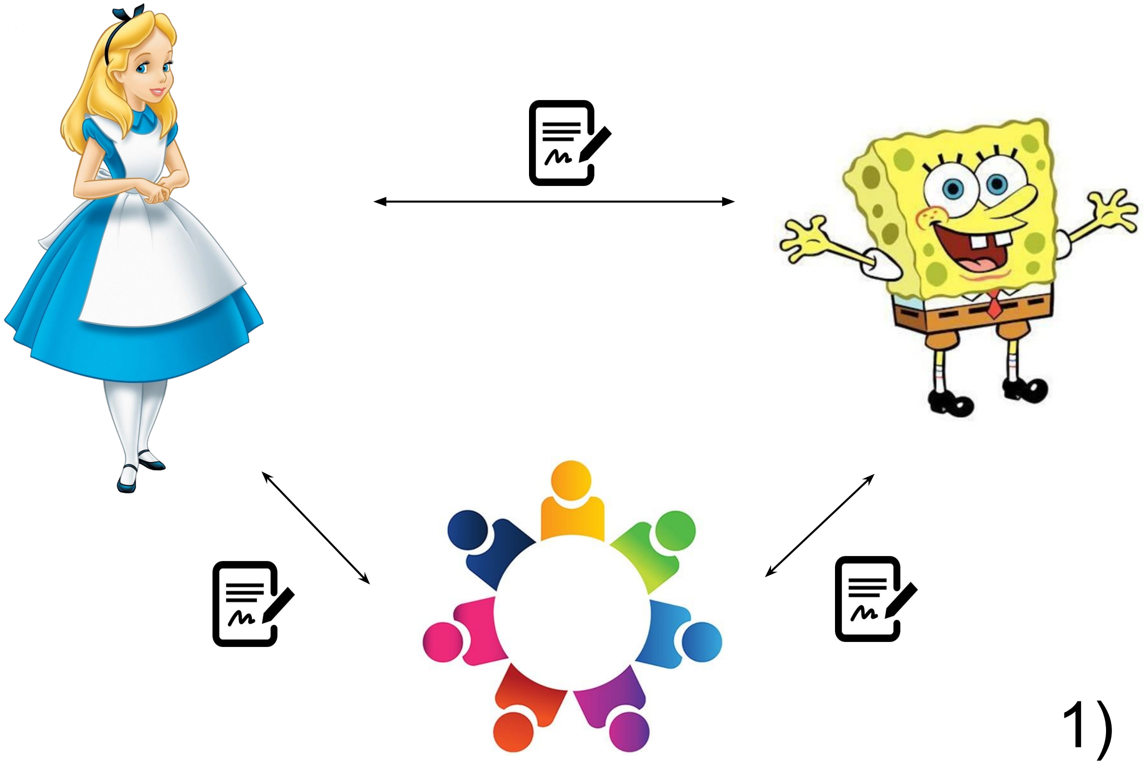


Committee
n = 3f+1
f Byzantine



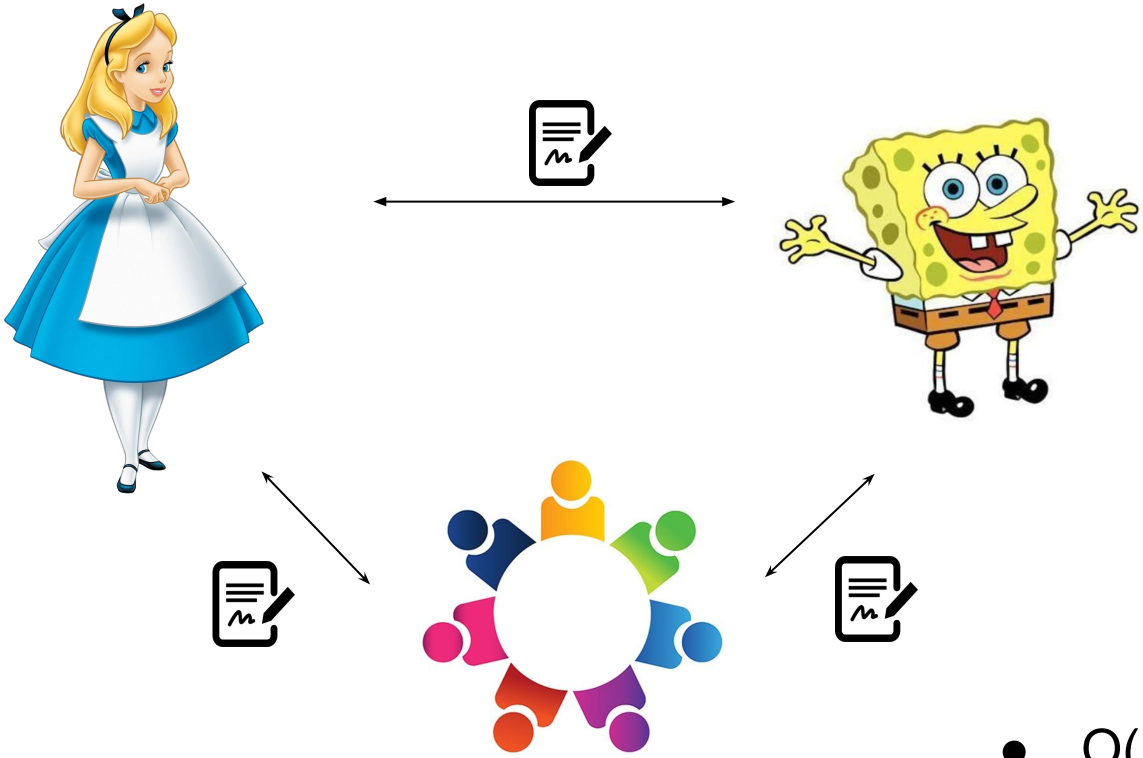
Signatures of Alice & Bob
OR
Signatures of 2f+1 WTs & (Alice or Bob)

Challenges



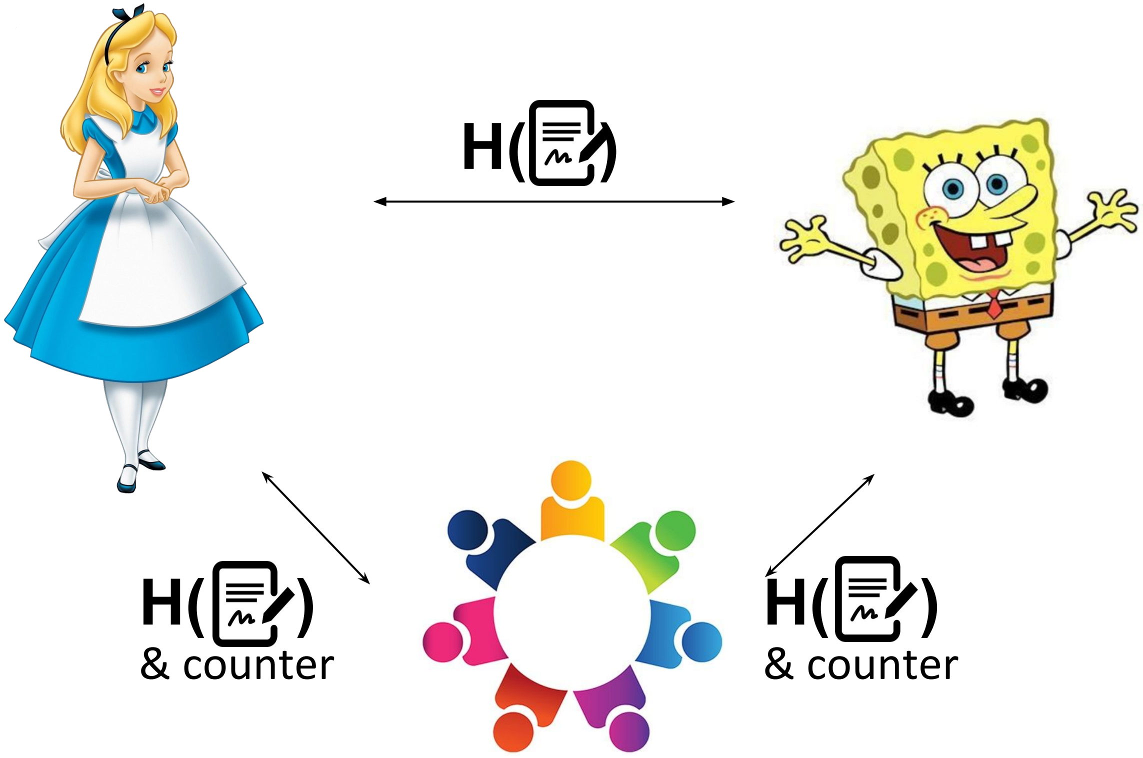
1) Consensus is costly
2) Privacy is important
3) Incentives are critical

Consistent Broadcast



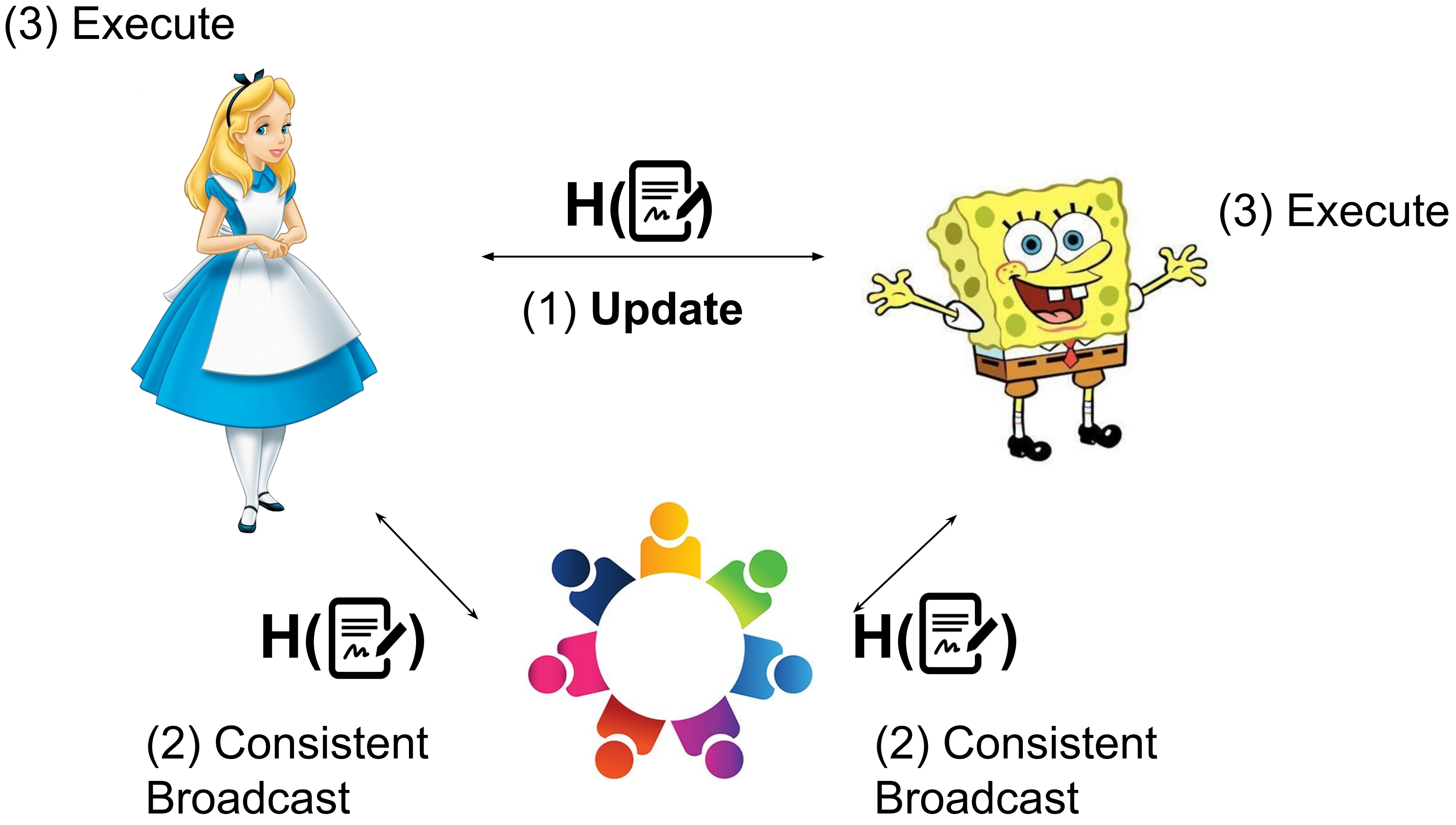
- O(n) communication complexity for state updates
- Verification of consensus between Alice & Bob
- No guarantees, if Alice & Bob both misbehave

Encrypted State



- Privacy preserving
- Alice/Bob cannot publish a previous transaction

Brick Architecture



(3) Execute

H(📝)

(1) **Update**

(3) Execute

H(📝)

(2) Consistent Broadcast

H(📝)

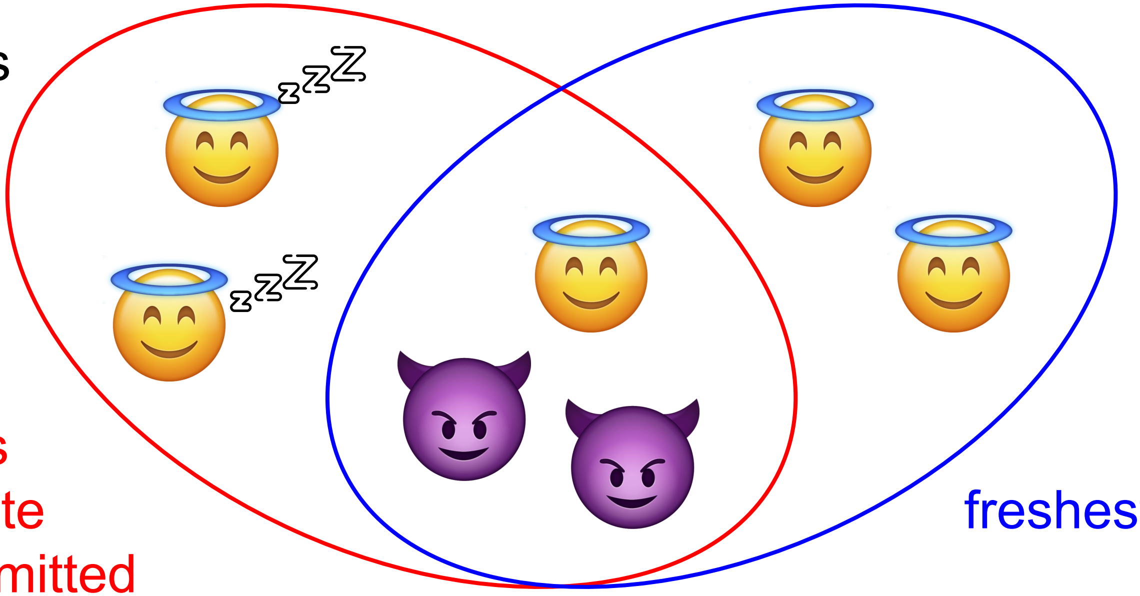
(2) Consistent Broadcast

Close: max state of 2f+1 submitted states.

Brick Security Analysis

Safety

A channel will only close in the freshest committed state



f slow honest WTs

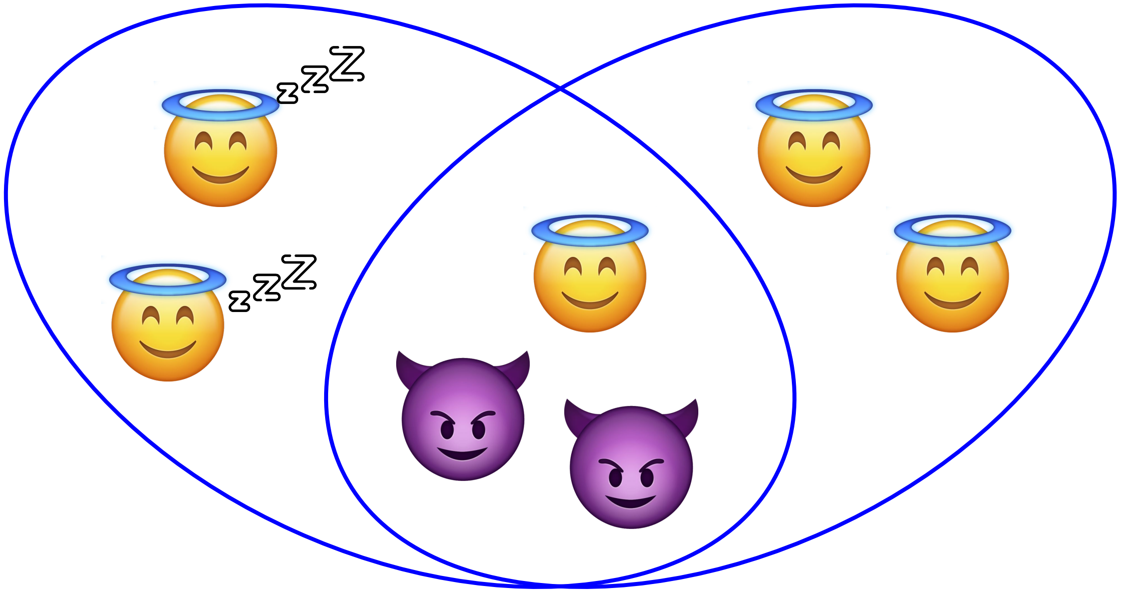
2f+1 WTs
closing state
(previous committed
state)

2f+1 WTs
freshest committed state

Brick Security Analysis

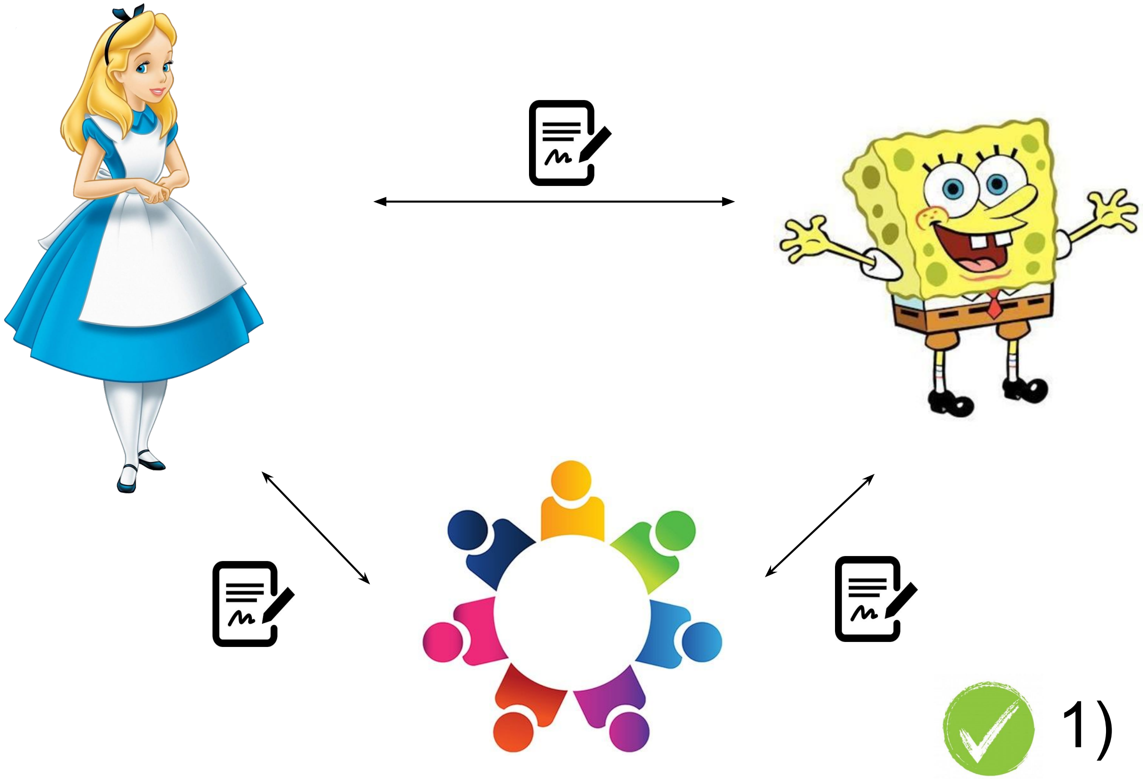
Liveness

Any valid operation (close, update)
will eventually be committed



Not committed = Invalid operation (failed verification)

Challenges



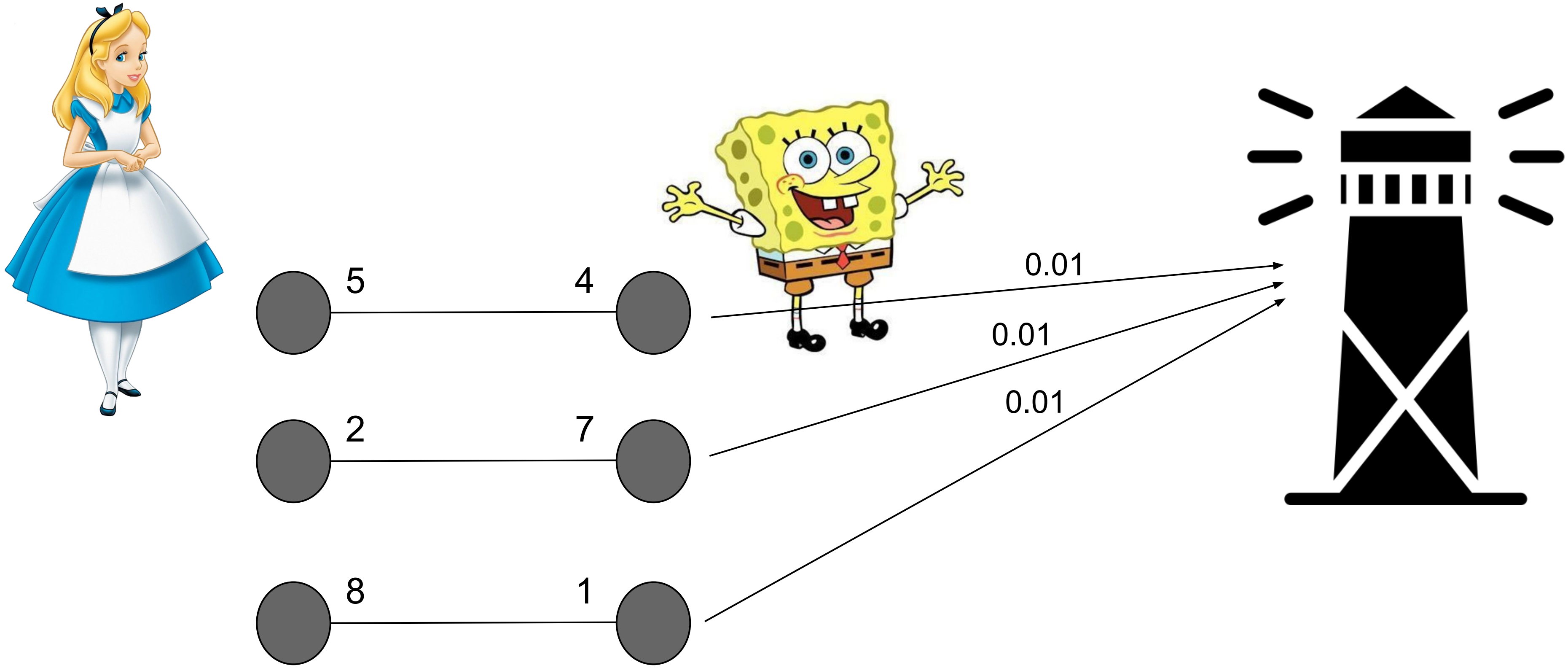
✓ 1) Consensus is costly

✓ 2) Privacy is important

3) Incentives are critical

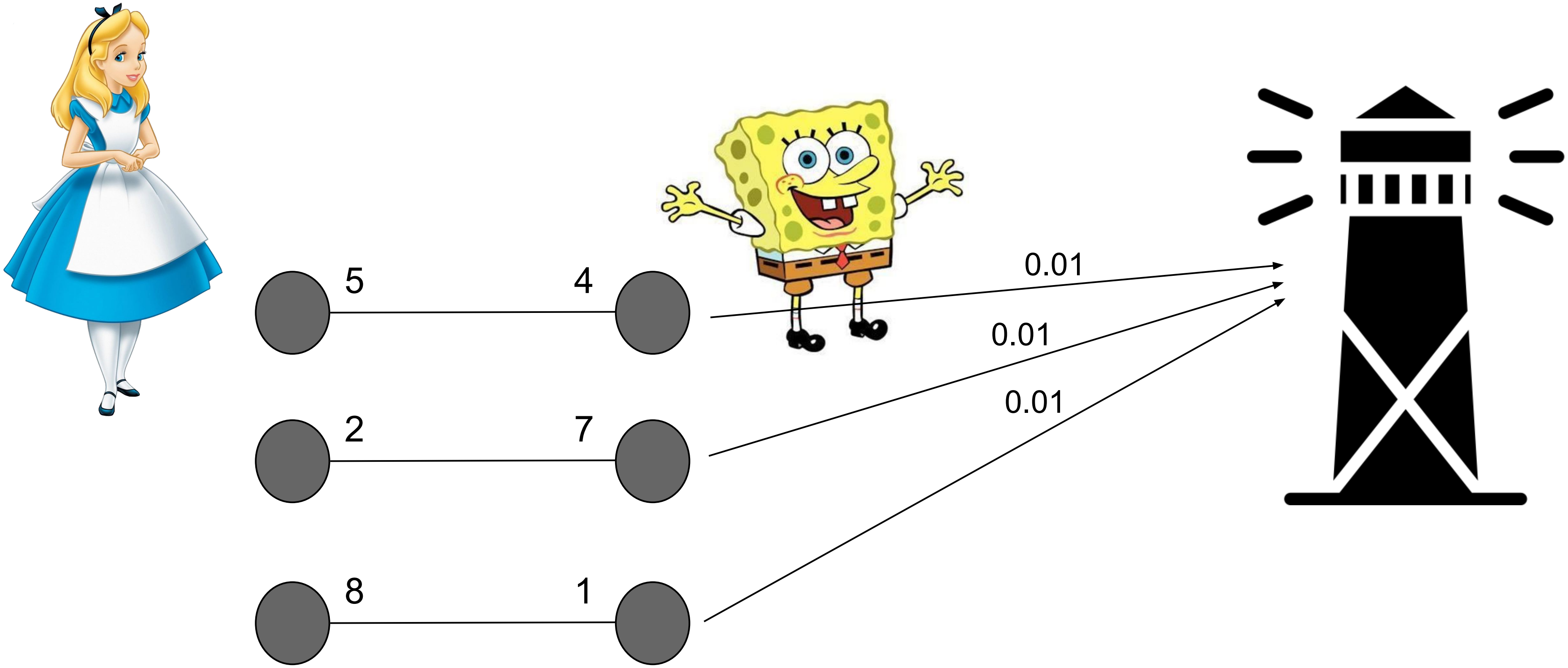
Why be a Watchtower?

Per-update fees



Repeated game lifts the fair-exchange impossibility

Per-update fees



Watchtower paid while channel is alive!
Incentives to close?

Why assist to close honestly?

Collateral

Why assist to close honestly?

Collateral

Asynchronous channels?

Collateral

Fraud proofs
two signed conflicting states

Party claims the collateral

Collateral

Fraud proofs
two signed conflicting states

Party claims the collateral

channel value
v

claimed collateral
$v/f * (f+1)$

Collateral

Where do we close?
when **>f fraud proofs** are submitted

all channel value→ counterparty

Collateral

Where do we close?
when **≤f fraud proofs** are submitted

run close again without the malicious → max state of 2f+1

Collateral

Profit =
channel balance (c) + fraud proofs (v/f) - bribes (v/f + ε)

1. **0 FPs**: profit = $c \leq v$

2. **> f FPs**: profit $\leq v + y*v/f - y*(v/f-\varepsilon) = v - \varepsilon$

3. **f FPs and "correct" close:** profit = $c + v$

4. **f FPs and "incorrect" close:** profit = $v - v/f - \varepsilon$

v = channel value
f = Byzantine watchtowers
y = bribed watchtowers

Why assist to close?

WTs collude → Hostage situations

Closing fees

prisoner’s dilemma

Why request close?

Parties collude → Hostage situations

Committee size > 7

richest party loses more

Committee size

The more (WTs) the merrier!

↑ robustness

↓ collateral per WT

≃ cost for parties

Brick Advantages

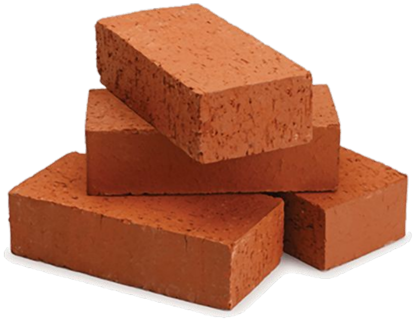
- Privacy
- Incentive-compatible
- Good performance
- **Asynchronous**
  - censorship
  - congestion
  - liveness attacks

Limitations, Extensions & Future Work

- Minimum collateral
- Update fees via one-way channel

Limitations, Extensions & Future Work

- Minimum collateral
- Update fees via one-way channel
- Watchtower replacement
- Auditability
- Consensus → fork resilient



Limitations, Extensions & Future Work

- Minimum collateral
- Update fees via one-way channel
- Watchtower replacement
- Auditability
- Consensus → fork resilient
- Multiple parties

Thank you!

Questions?

Z. Avarikioti, E. Kokoris-Kogias, R. Wattenhofer. *Brick: Asynchronous State Channels.* arXiv:1905.11360