Lightning watchtowers and why you should use them

Sergi Delgado

sr_gi

Talaia Labs

LIGHTNING 101

sr_gi

Talaia Labs

CHANNEL LIFE CYCLE: OPENING A CHANNEL

- When two parties want to open a channel they lock-up funds in a 2-2 multisig contract (P2WSH)
- This lockup translates into a transaction on chain, known as **funding transaction**
- This is currently, mostly, single funded

funding transaction

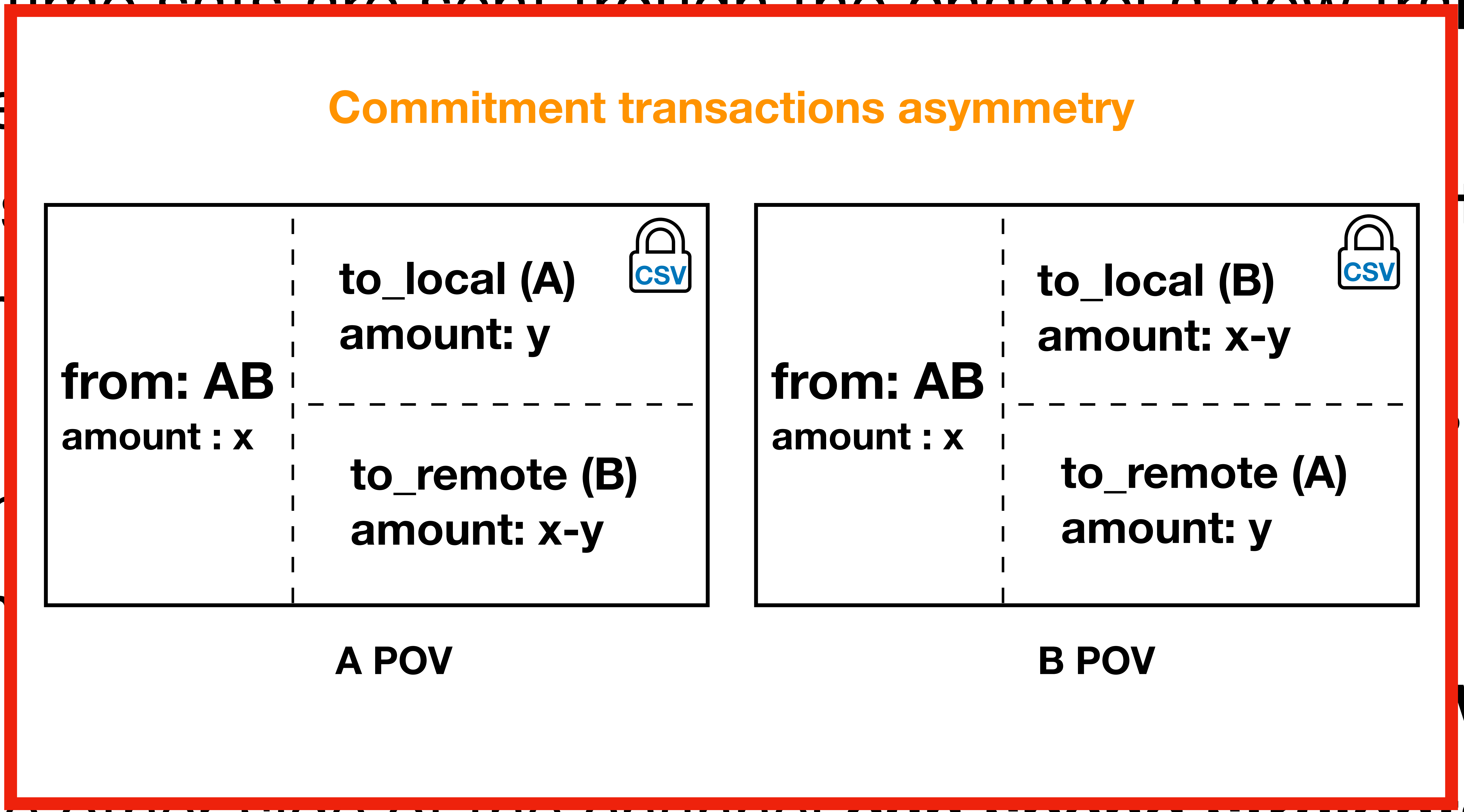
from: A	**to: AB**

CHANNEL LIFE CYCLE: USING A CHANNEL I

- Each time sats are sent trough the channel a new transaction is created (actually two)
- All this transactions spend from the **funding transaction** and are known as **commitment transactions**
- Each side of the channel has a non-symmetrical version of each commitment
- Commitment transaction are time-locked as follows:
  - The spender **has to wait to spend his funds** (**CSV**)
  - The other side of the channel **can spend straightaway**

CHANNEL LIFE CYCLE: USING A CHANNEL I

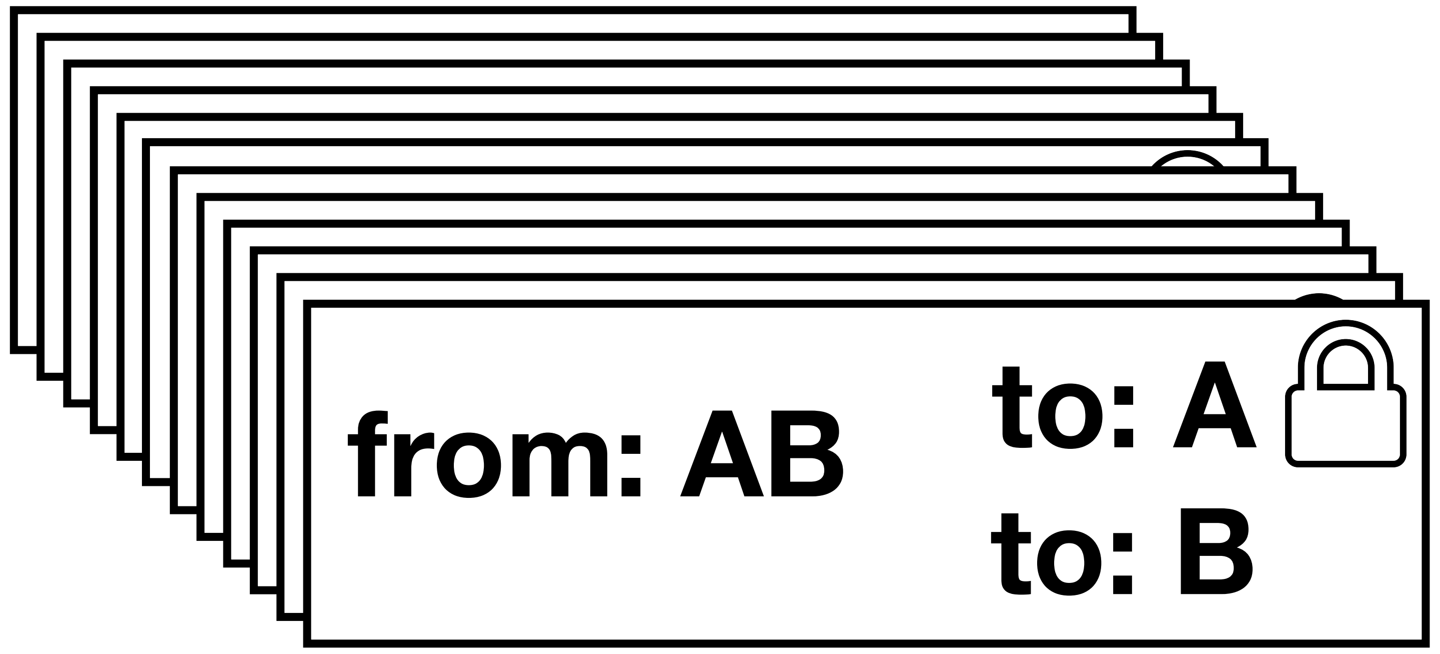
- Each time sats are sent trough the channel a new transaction is create
- All thi **tion** and are kn
- Each ion of each comm
- Comm
  - Th **V**)
  - The other side of the channel **can spend straightaway**



CHANNEL LIFE CYCLE: USING A CHANNEL II

- A new commitment transaction revokes the previous one
- All revoked commitments are valid Bitcoin transactions
- Only a single commitment transaction is valid on chain
- Only the owners of the channel know which is the last commitment

commitment transactions



CHANNEL LIFE CYCLE: CLOSING A CHANNEL

- The last commitment transaction is used as the base to create a transaction **without time locks**
- This transaction is known as **closing transaction**
- Both parties **can spend straightaway** from the closing transaction
- The closing transaction will be stored on chain

closing transaction

from: AB	**to: A** **to: B**

CHANNEL LIFE CYCLE: BREACHING A CHANNEL

- A side of the channel can try to spend an **already revoked commitment** (either “by mistake” or maliciously)
- This attempt is known as **channel breaching** and can result in the other side getting less funds than they deserve
- Only the channel owners can identify a revoked commitment, therefore **both sides may remain online**
- This is known as the **always online assumption / requirement**

CHANNEL LIFE CYCLE: PUNISHING MISBEHAVIOUR I

- If a channel is breached by one side, the other side can (and will) penalise the attempt
- A transaction will be created using the revocation data and spending from the channel breach
- This transaction is known as **penalty transaction** and it claims all funds of the channel instantly

penalty transaction

from: AB	to: B

CHANNEL LIFE CYCLE: PUNISHING MISBEHAVIOUR II

But what if the owners of the channel **were not the only ones** who could react to channel breaches?

CHANNEL LIFE CYCLE: PUNISHING MISBEHAVIOUR II

But what if the owners of the channel **were not the only ones** who could react to channel breaches?

Watchtowers

sr_gi

Talaia Labs

GENERAL CONCEPT

What is the general paradigm behind third party watching systems (**AKA Watchtowers**)?

User:

- Sends **data** to the server alongside a **trigger** condition

Server:

- **Looks for triggers** on a communication channel
- If the a trigger is seen, perform **an action** with the provided data

BASIC WATCHTOWER PROTOCOL

BASIC WATCHTOWER PROTOCOL

BASIC WATCHTOWER PROTOCOL

BASIC WATCHTOWER PROTOCOL

[…]
commitment_txid,
penalty_tx,
[…]

BASIC WATCHTOWER PROTOCOL

[…]
commitment_txid,
penalty_tx,
[…]

BASIC WATCHTOWER PROTOCOL

appointment

[...]
commitment_txid,
penalty_tx,
[...]

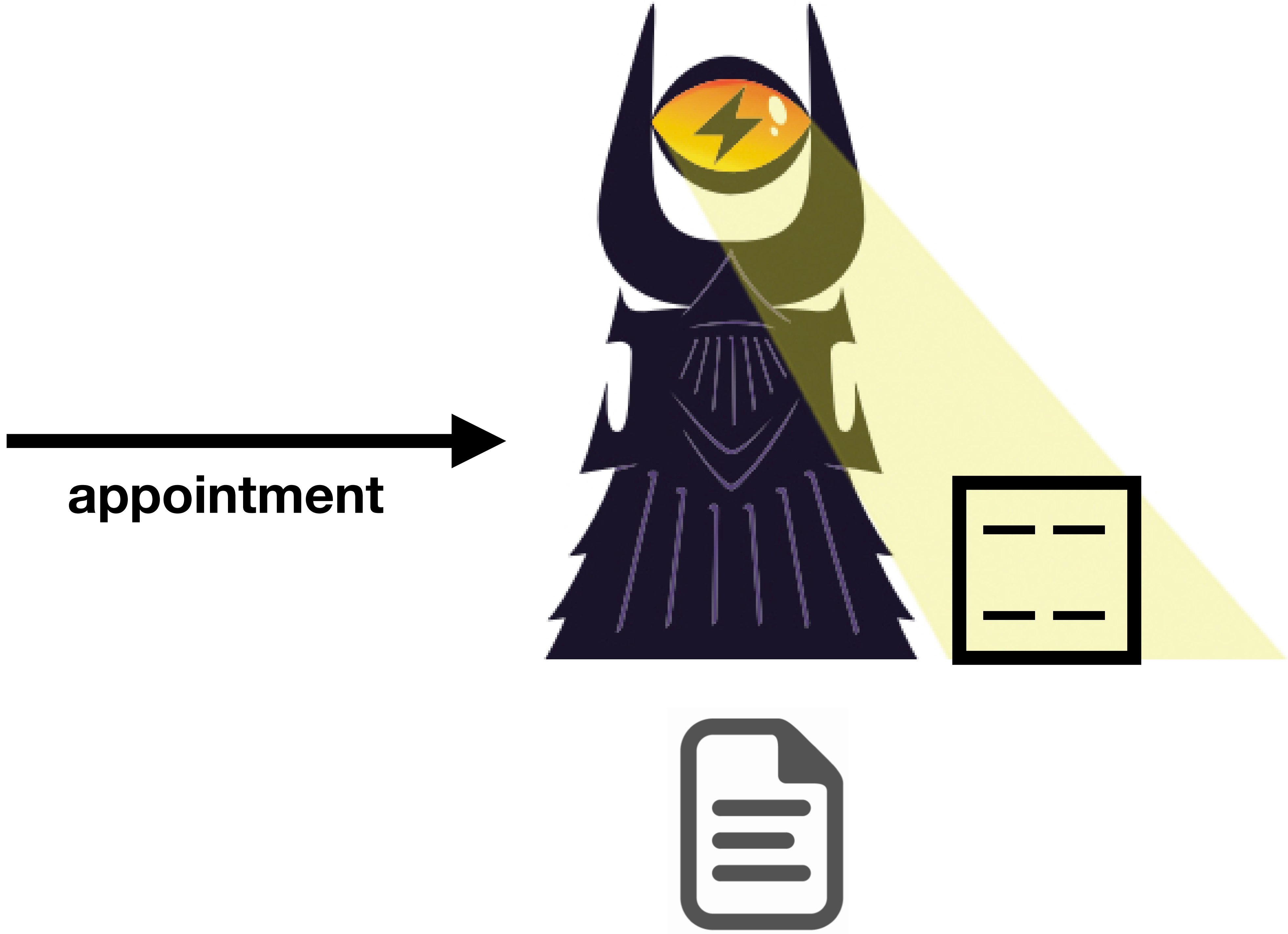
BASIC WATCHTOWER PROTOCOL

appointment

[…]
commitment_txid,
penalty_tx,
[…]

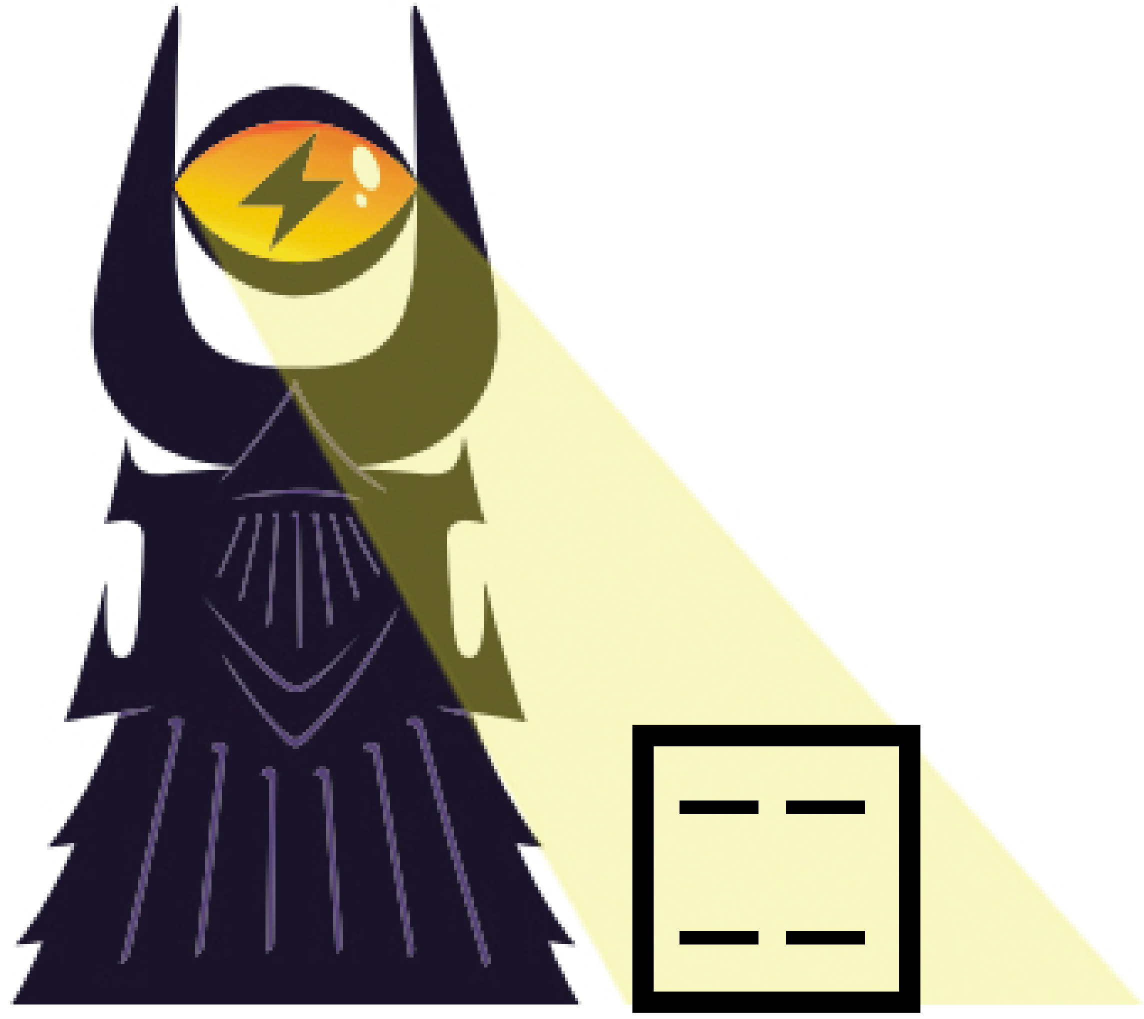
BASIC WATCHTOWER PROTOCOL

[…]
commitment_txid,
penalty_tx,
[…]



BASIC WATCHTOWER PROTOCOL

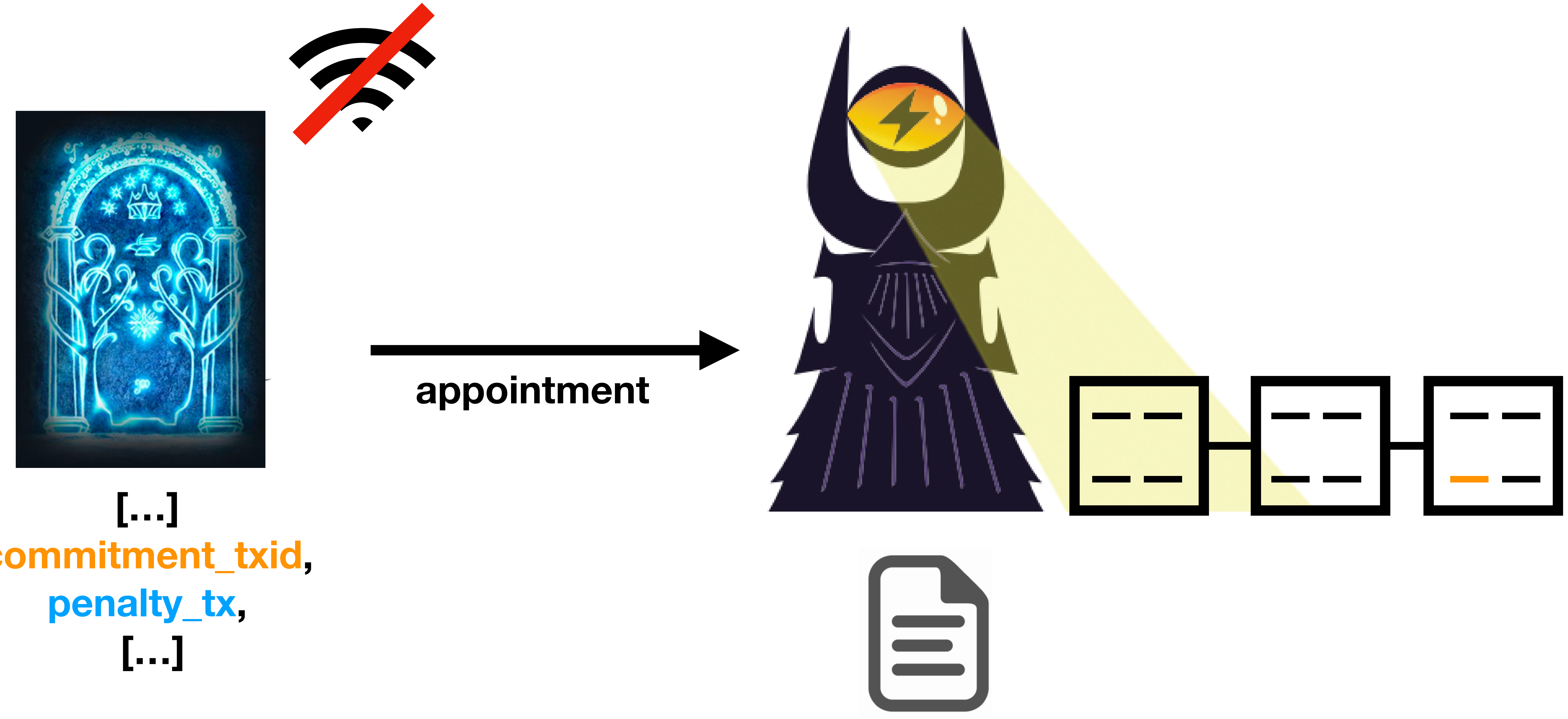
appointment



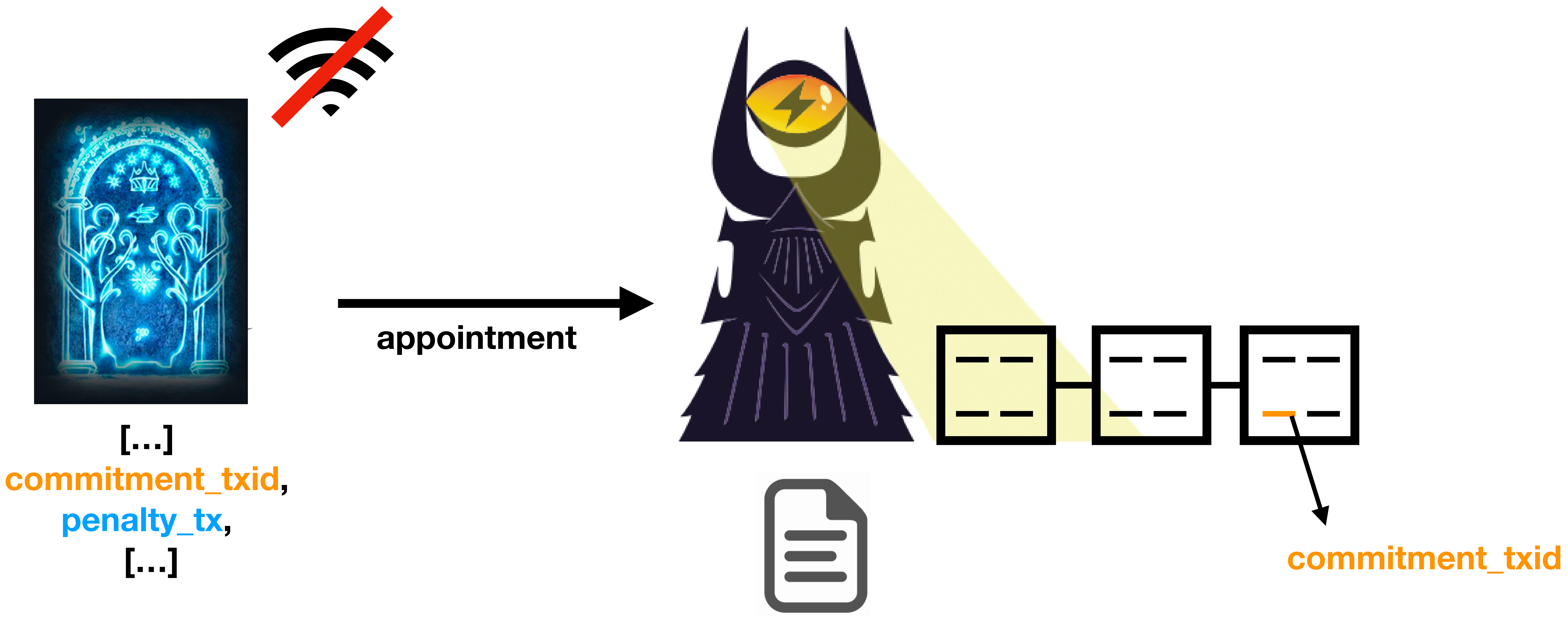
[...]
commitment_txid,
penalty_tx,
[...]

BASIC WATCHTOWER PROTOCOL

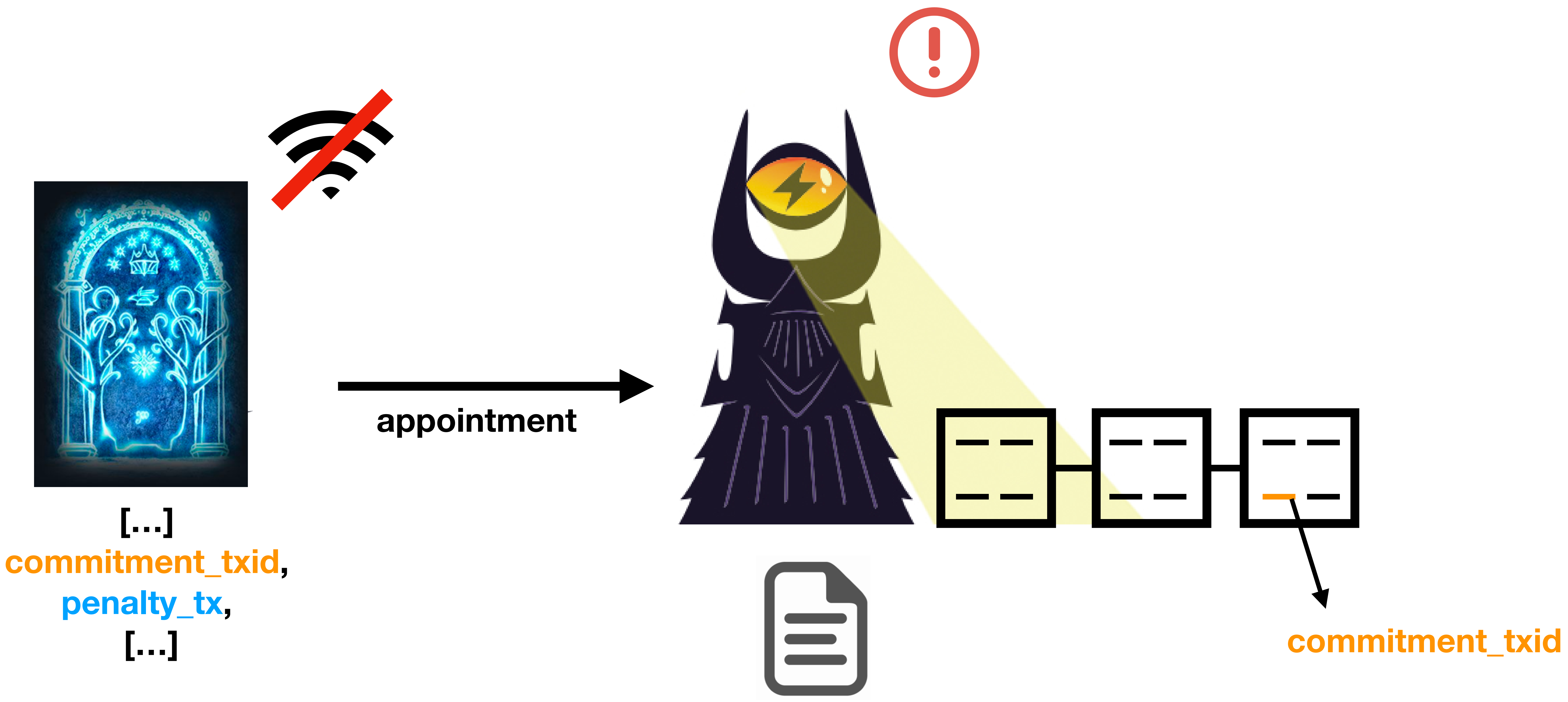
BASIC WATCHTOWER PROTOCOL



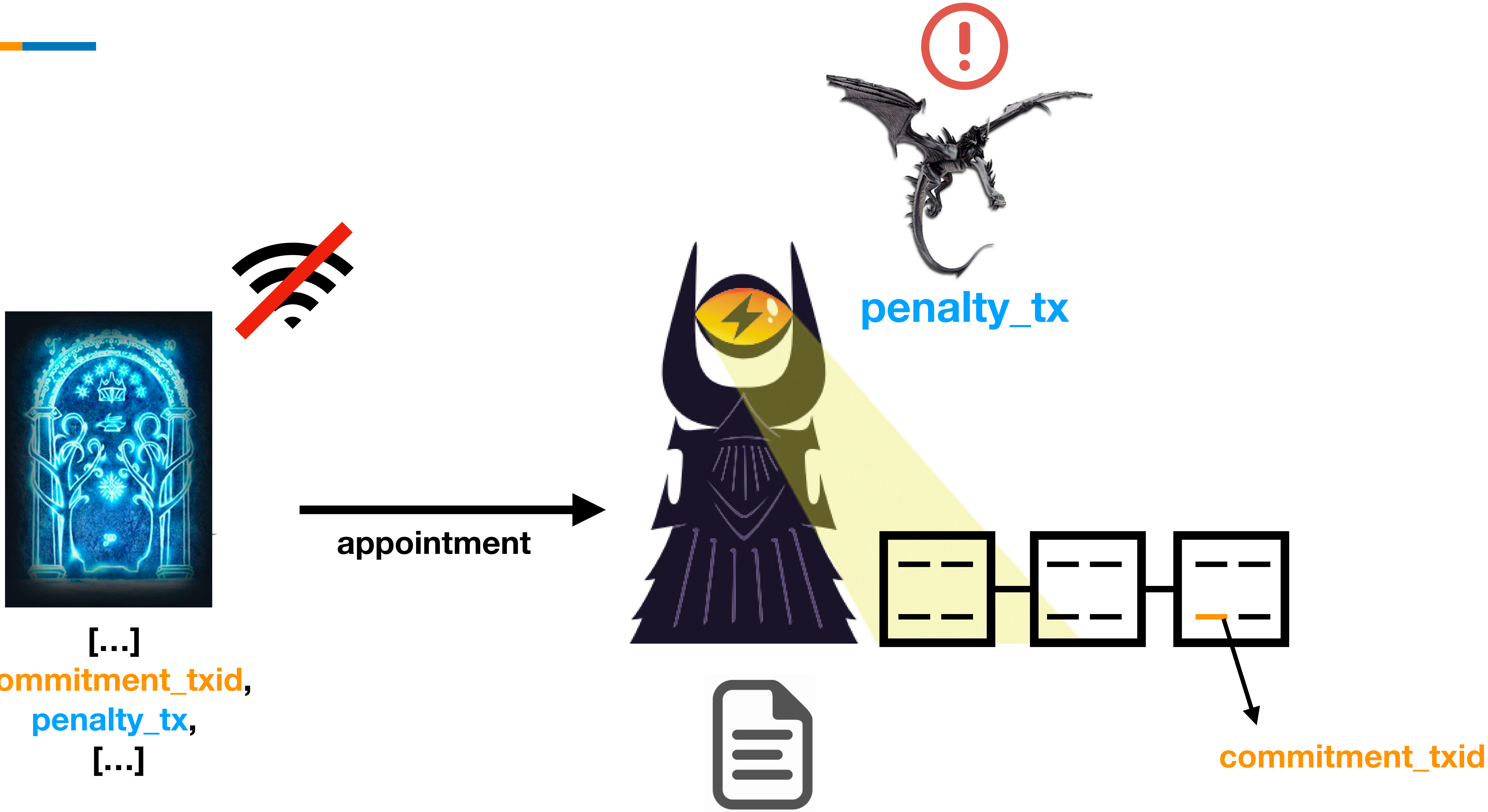
BASIC WATCHTOWER PROTOCOL



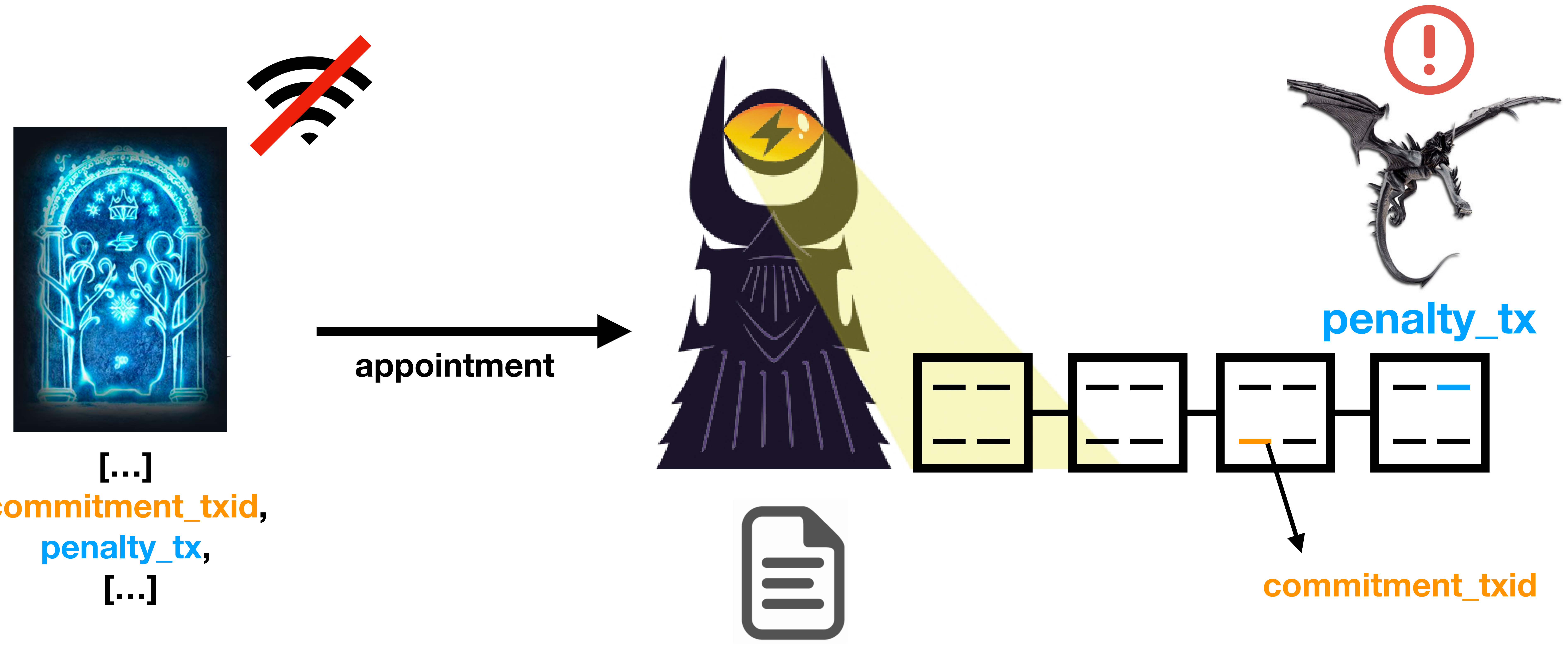
BASIC WATCHTOWER PROTOCOL



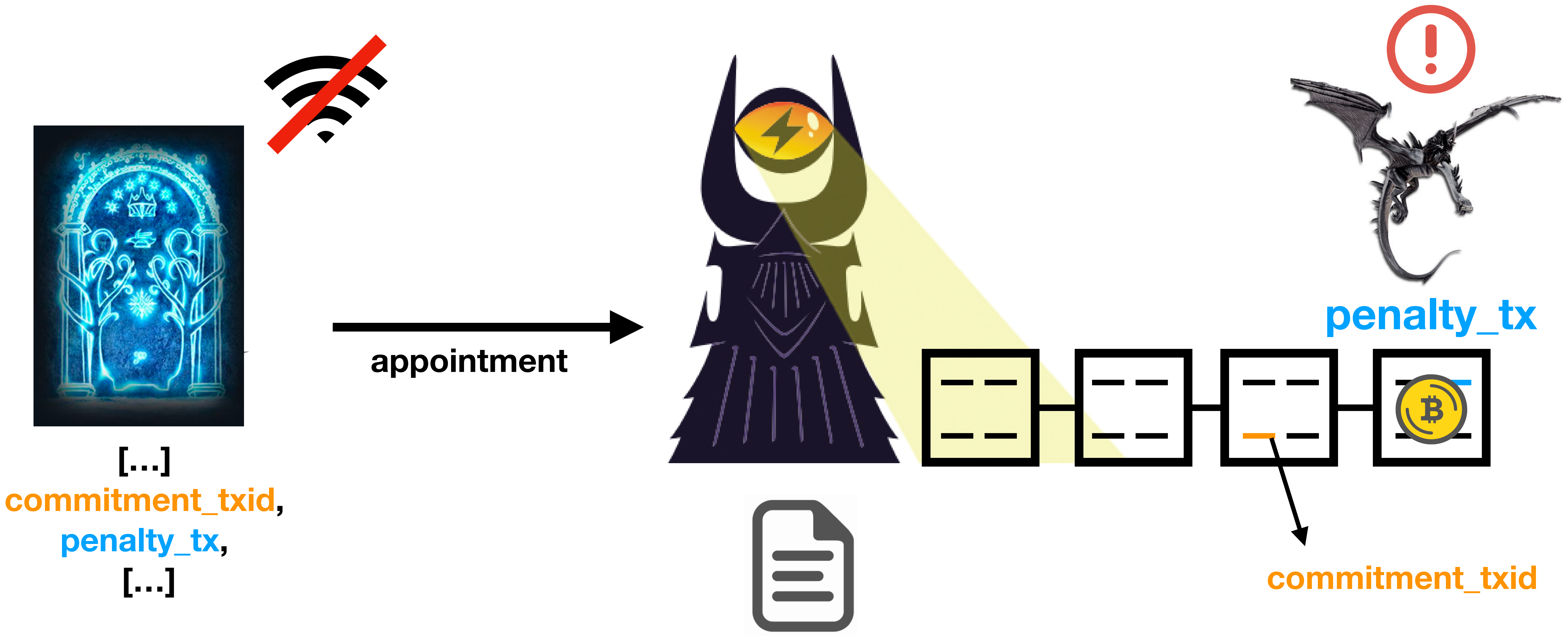
BASIC WATCHTOWER PROTOCOL



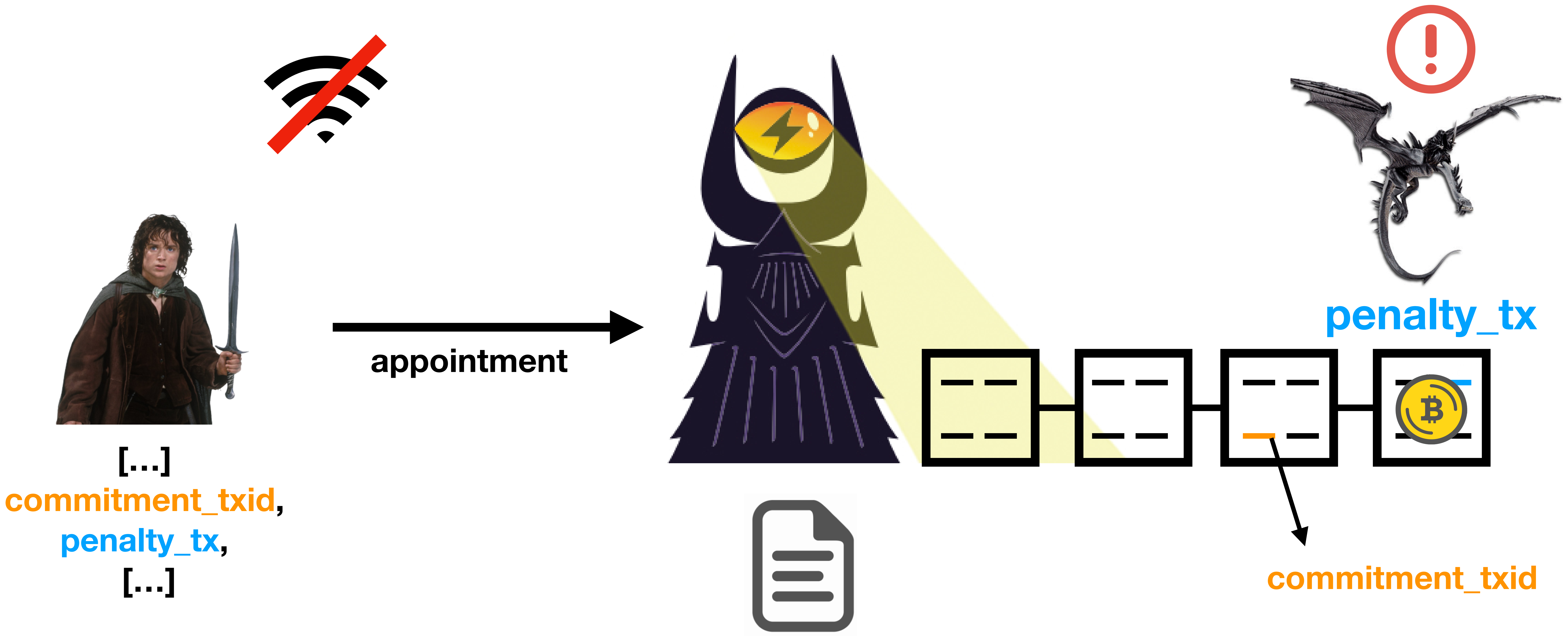
BASIC WATCHTOWER PROTOCOL



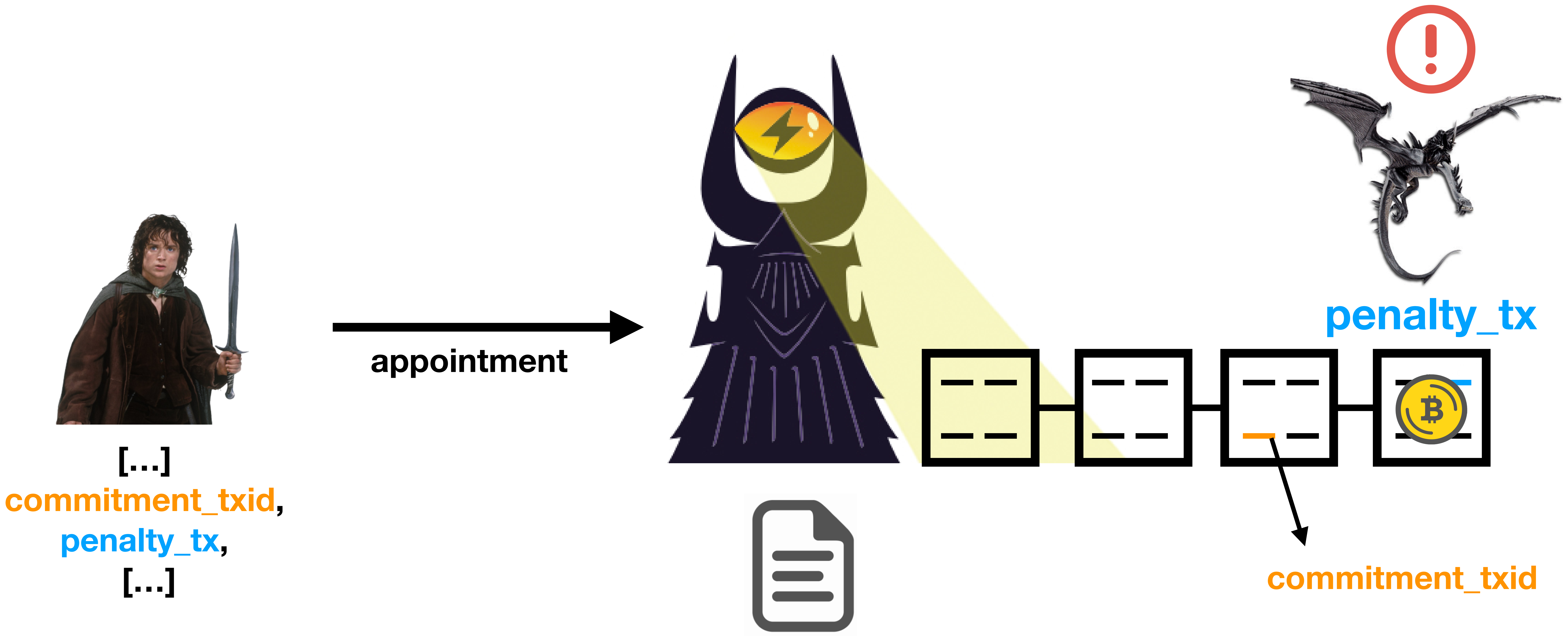
BASIC WATCHTOWER PROTOCOL



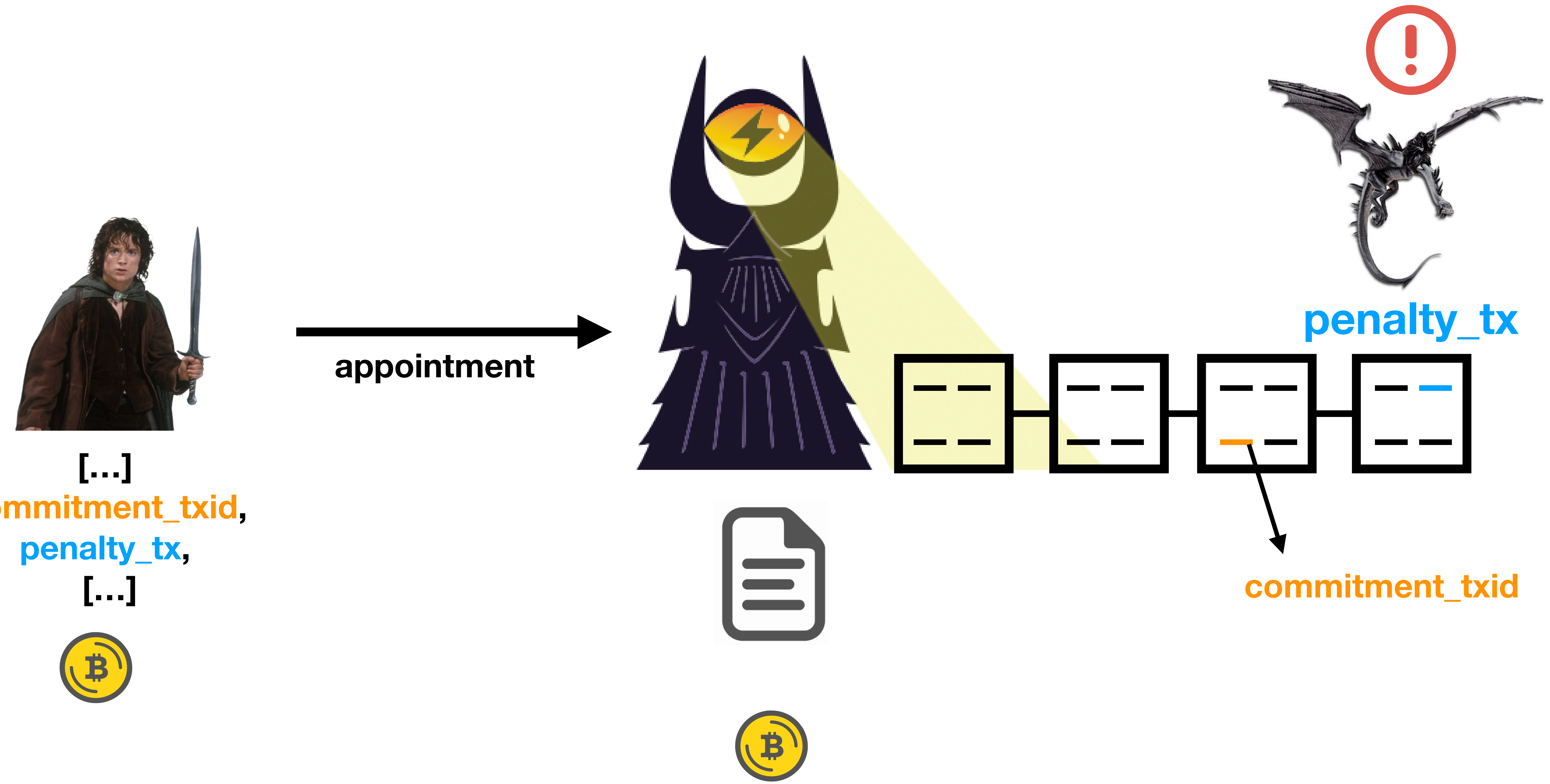
BASIC WATCHTOWER PROTOCOL



BASIC WATCHTOWER PROTOCOL



BASIC WATCHTOWER PROTOCOL



WHY USING A TOWER (OR WHY NOT)? I

A watchtower is a **failsafe** mechanism with, mainly, two properties:

- High availability
- Data redundancy for your node

It is useful for all kind of nodes, but specially for those that lack some of the properties it provides (e.g. mobile nodes)

So, when may we need one and what are the alternatives?

WHY USING A TOWER (OR WHY NOT)? II

We may **not need** a tower if we have a highly available node. That means:

- Power supply redundancy
- Internet service redundancy
- Data redundancy

This may be the case of a top tier routing node, **not the average node.** Even here, some types of towers may be worth considering.

WHY USING A TOWER (OR WHY NOT)? III

We may **need** a tower if we have:

- A non-highly available node (see previous slide)
- A mobile phone node

Currently, the network is mostly run by techies and enthusiast, but the average user is **unlikely to run a highly available node.**

WHY USING A TOWER (OR WHY NOT)? IV

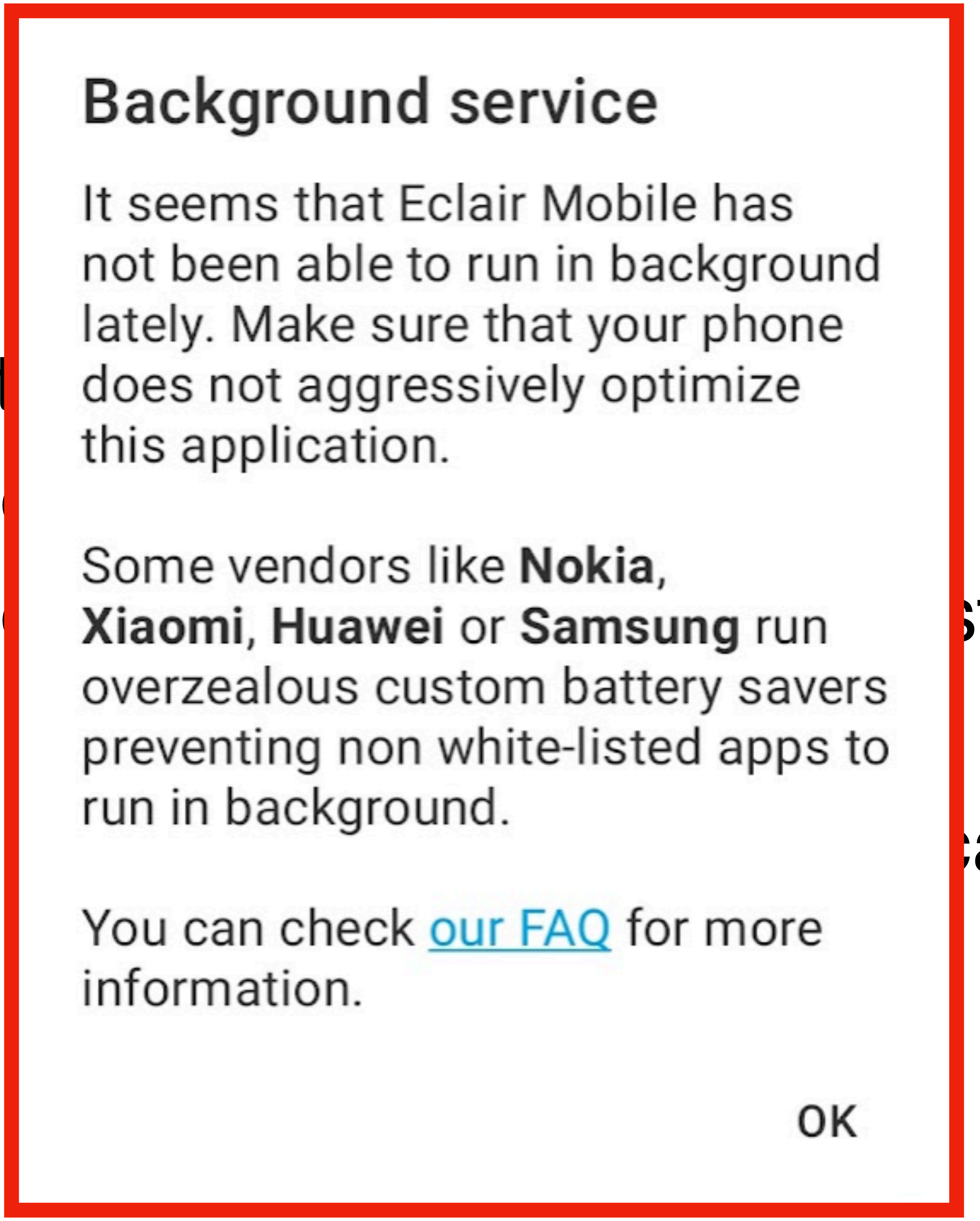
Mobile nodes:

- Intermittent Internet access
- Lower bandwidth use w.r.t routing nodes
- Easier to lose data (phone breaks / gets stolen / …)
- Node may not be always online
- Sporadically used (mainly when paying, can be offline for days)

WHY USING A TOWER (OR WHY NOT)? IV

Mobile nodes:

- Intermittent Int
- Lower bandwi
- Easier to lose   stolen / …)
- Node may not
- Sporadically u   can be offline for days)



Types of towers

@sr_gi

Talaia Labs

GENERAL PURPOSE TOWER I

For every channel update:

- The penalty transaction is **encrypted** under a key derived from the commitment transaction id
- A **locator** is also derived from the commitment transaction id
- The tower receives the **encrypted blob and the locator**

Tadge Dryja - Unlinkable Outsourced Channel Monitoring - Scaling Bitcoin 2016

GENERAL PURPOSE TOWER II

User side

GENERAL PURPOSE TOWER II

User side

penalty_tx = 0200000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

GENERAL PURPOSE TOWER II

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

GENERAL PURPOSE TOWER II

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB

GENERAL PURPOSE TOWER II

User side

penalty_tx = 0200000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB ⟶ locator

GENERAL PURPOSE TOWER II

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → locator

cipher = CHACHA20POLY1305

sk = SHA256(commitment_txid)

IV = 0

GENERAL PURPOSE TOWER II

User side

penalty_tx = 0200000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → locator

cipher = CHACHA20POLY1305

sk = SHA256(commitment_txid)

IV = 0

encrypt (penalty_tx, sk, IV) →

GENERAL PURPOSE TOWER II

User side

penalty_tx = **020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000**

commitment_txid = **4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33**

16 MSB → **locator**

cipher = CHACHA20POLY1305

sk = SHA256(commitment_txid)

IV = 0

encrypt (penalty_tx, sk, IV) → **encrypted blob**

GENERAL PURPOSE TOWER II

User side

penalty_tx = 0200000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → locator

cipher = CHACHA20POLY1305

sk = SHA256(commitment_txid)

IV = 0

encrypt (penalty_tx, sk, IV) → encrypted blob

GENERAL PURPOSE TOWER II

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → $\boxed{\text{locator}}$

SEND TO THE TOWER

cipher = CHACHA20POLY1305

sk = SHA256(commitment_txid)

IV = 0

encrypt (penalty_tx, sk, IV) → $\boxed{\text{encrypted blob}}$

GENERAL PURPOSE TOWER III

Tower side

GENERAL PURPOSE TOWER III

Tower side

for every transaction_id in every block

locator = 16 MSB transaction_id

GENERAL PURPOSE TOWER III

Tower side

for every transaction_id in every block

    locator = 16 MSB transaction_id

        if locator in appointments:

            sk = SHA256(transaction_id)

            IV = 0

GENERAL PURPOSE TOWER III

Tower side

for every transaction_id in every block

locator = 16 MSB transaction_id

if locator in appointments:

sk = SHA256(transaction_id)

IV = 0

decrypt (encrypted blob, sk, IV)

GENERAL PURPOSE TOWER III

Tower side

for every transaction_id in every block

locator = 16 MSB transaction_id

if locator in appointments:

sk = SHA256(transaction_id)

IV = 0

decrypt (encrypted blob, sk, IV) → **penalty_tx**

GENERAL PURPOSE TOWER IV

Pros:

- Privacy preserving
- Can give service to the whole network (with one or multiple towers)
- No infrastructure needed for the user

Cons:

- Design is rather complex
- O(N) storage
- Can be easily spammed (altruistic vs non-altruistic)

PERSONAL USE ONLY TOWER I

If the tower is for personal use only, the design can be highly simplified:

- Privacy is not a concern (at least not at the same level)
- Data may not need to be encrypted
- Most of the spam protections included in the design can be lifted
- Storage can be O(1)

PERSONAL USE ONLY TOWER II

Examples:

- Channel ids can be shared with the tower, so no more need for locators (only renovation data) (**O(N) but simpler design**)
- Even revocation keys could be shared, so the storage is drastically reduced (**O(1) but riskier**)

Haven’t seen any tower like this in the wild. I’ve seen some devs proposing / pursuing them though (h/t Antoine Riard)

PERKS OF USING ANY KIND OF TOWER I

CSV in to_local outputs **COULD** be reduced:

	CSV (default)	~ time
c-lightning	144	1D
LND	144-2016	1D - 2W
Eclair	720	5D
rust-lightning	144	1D

PERKS OF USING ANY KIND OF TOWER II

Why?

Unresponsive peers. If you have to close unilaterally you are forced to wait the CSV delay.

However

It will be unsafe to set a too low CSV if no-one is watching your channels

PERKS OF USING ANY KIND OF TOWER III

Also

Having a too big CSV may hurt UX

Finally

This is not tradeoff free. Depending on the setup you may be trusting the tower to respond

SPECIAL THANKS TO MY SPONSORS

Square Crypto

8 sponsors are funding sr-gi's work.

sr_gi

Talaia Labs

Q&A

sr_gi

Talaia Labs