Watchtowers: from design to deployment

Sergi Delgado

sr_gi

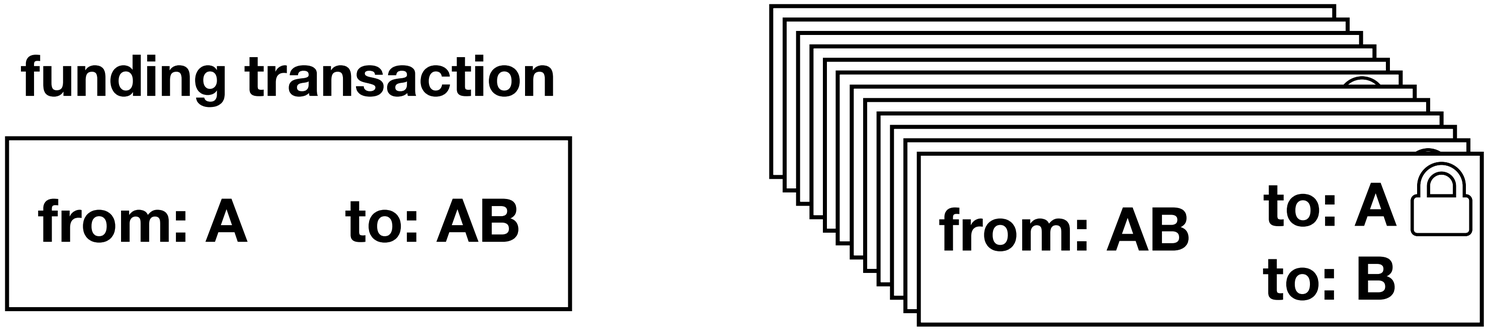
Talaia Labs

A BIT OF BACKGROUND

LIGHTNING CHANNEL LIFE CYCLE

commitment transactions

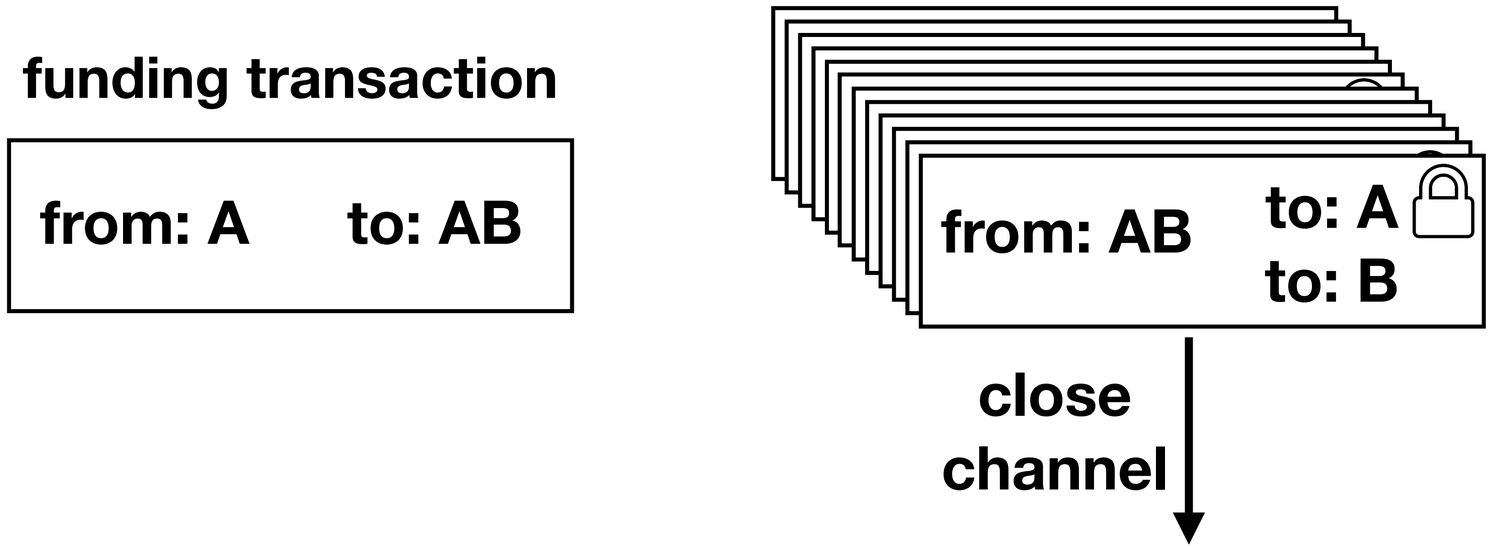
funding transaction



LIGHTNING CHANNEL LIFE CYCLE

commitment transactions

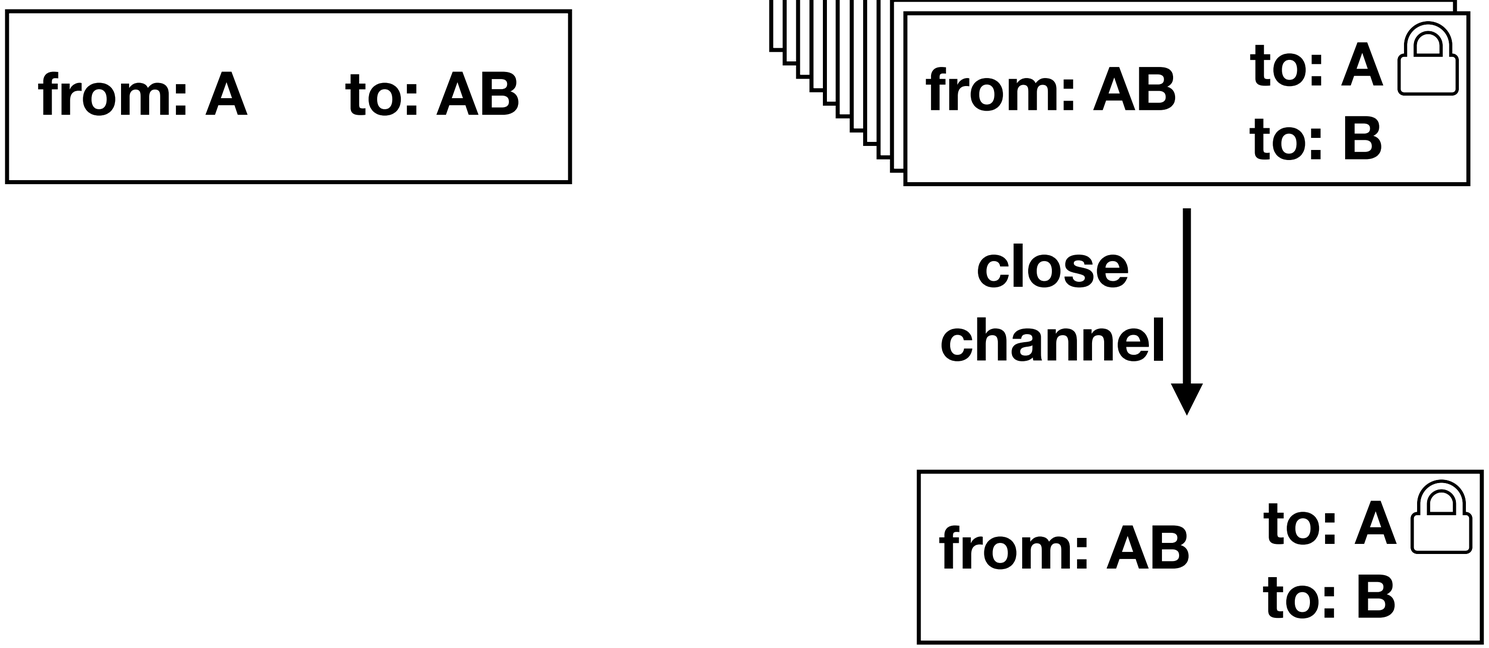
funding transaction



LIGHTNING CHANNEL LIFE CYCLE

commitment transactions

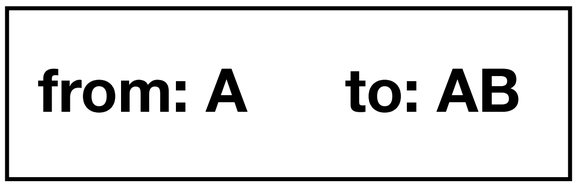
funding transaction

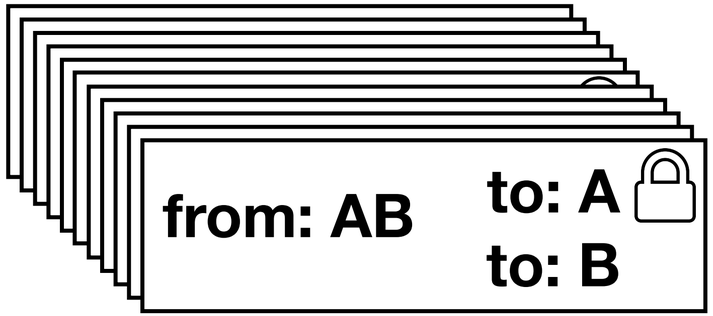


LIGHTNING CHANNEL LIFE CYCLE

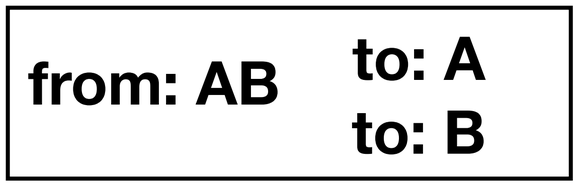
commitment transactions

funding transaction





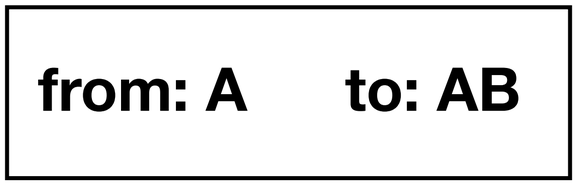
close channel



closing transaction

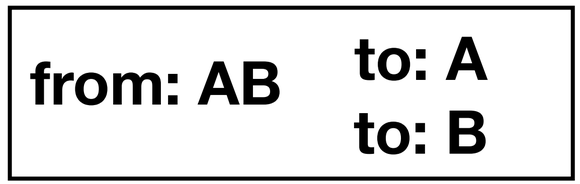
LIGHTNING CHANNEL LIFE CYCLE

funding transaction



commitment transactions

close channel

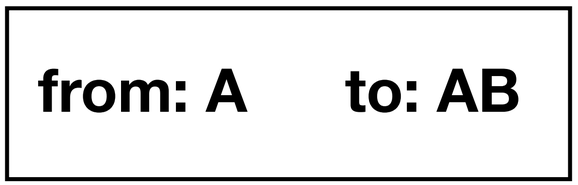


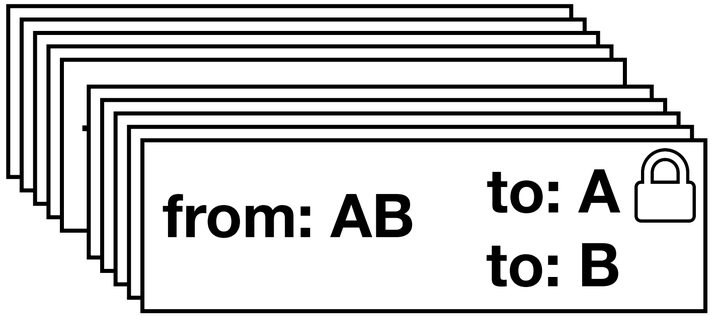
closing transaction

LIGHTNING CHANNEL LIFE CYCLE

commitment transactions

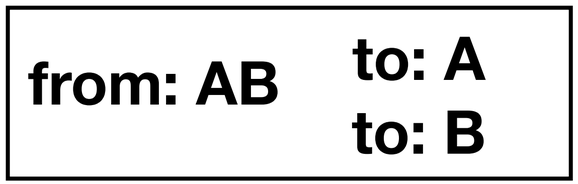
funding transaction





close channel

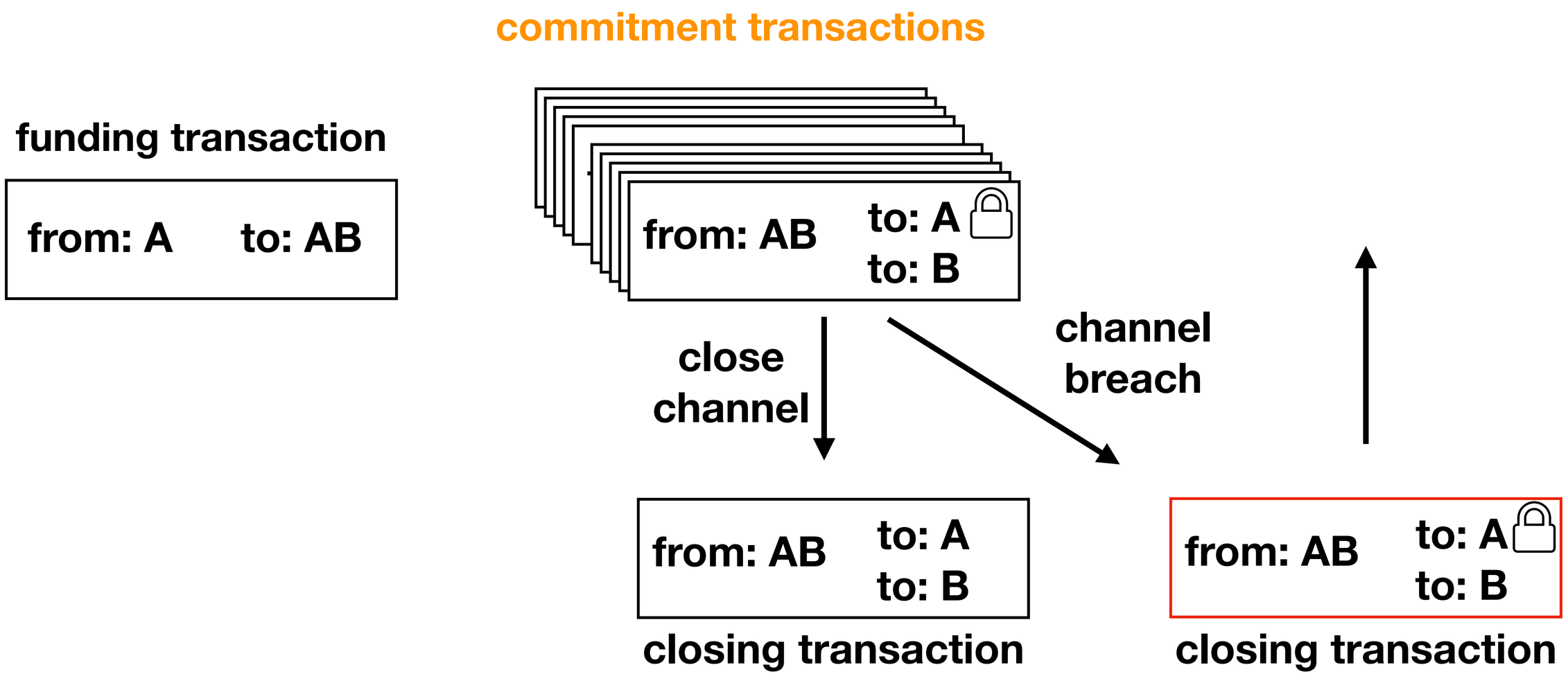
channel breach



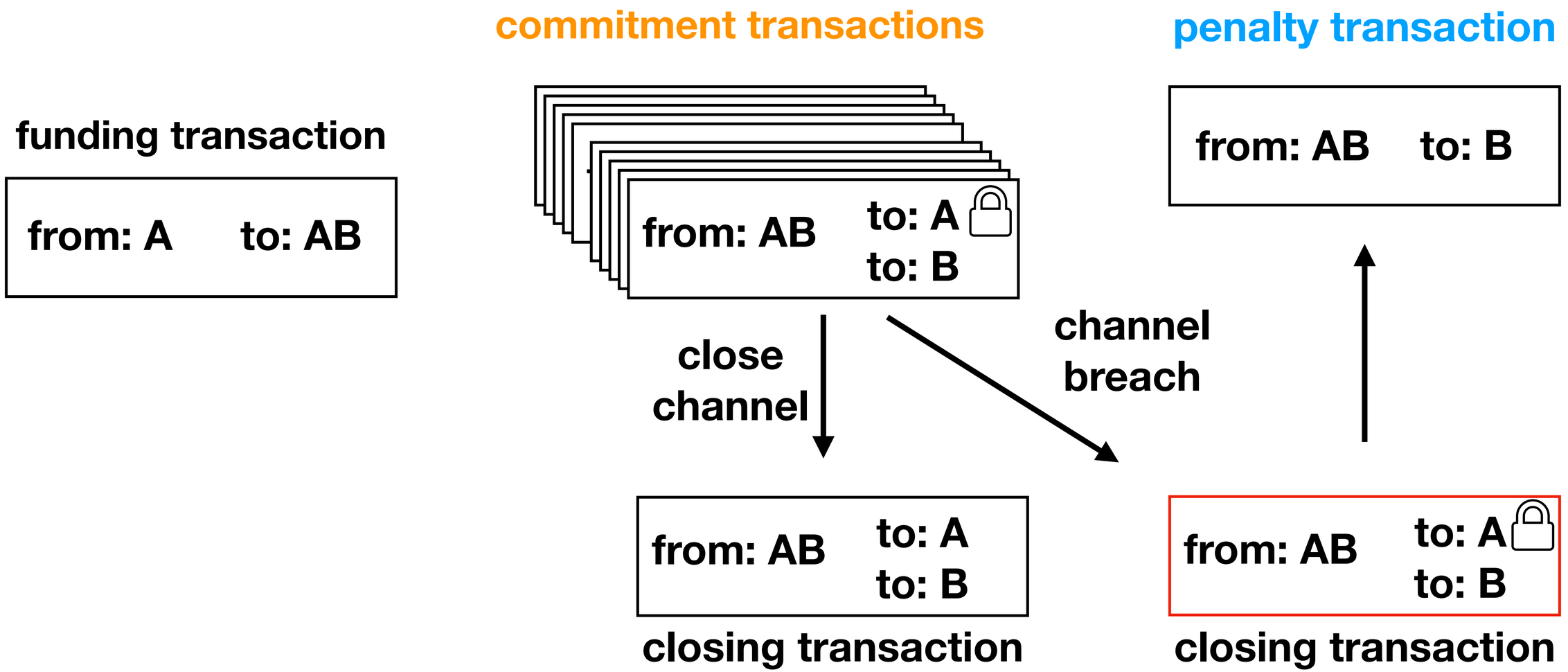
closing transaction

closing transaction

LIGHTNING CHANNEL LIFE CYCLE



LIGHTNING CHANNEL LIFE CYCLE



THE THEORY

WATCHTOWERS

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

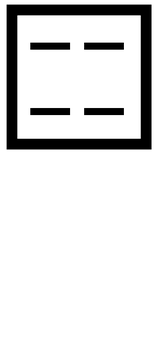
[...]
commitment_txid,
penalty_tx,
[...]

appointment

BASIC WATCHTOWER PROTOCOL

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

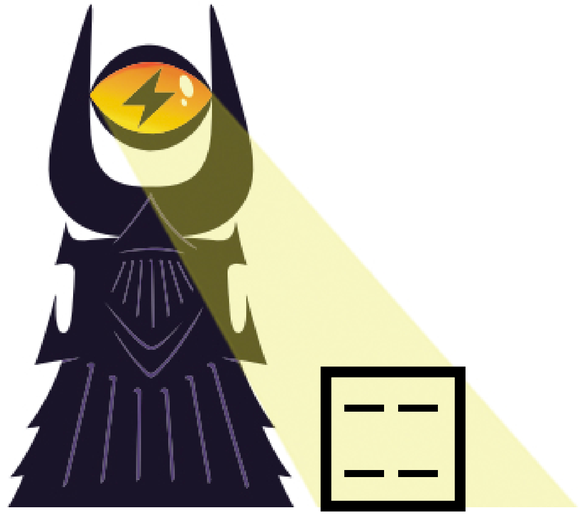
appointment



BASIC WATCHTOWER PROTOCOL

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

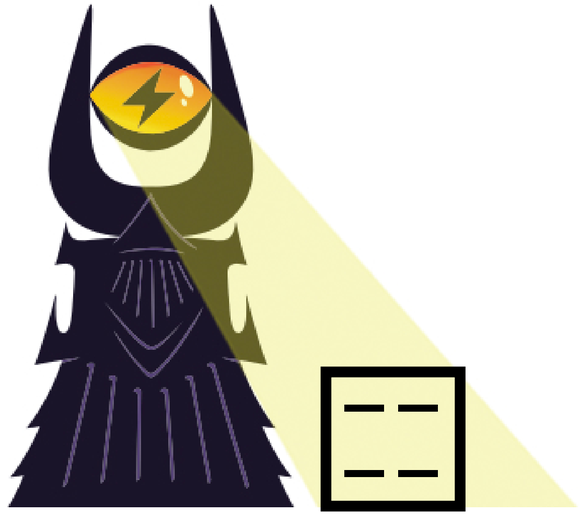
appointment



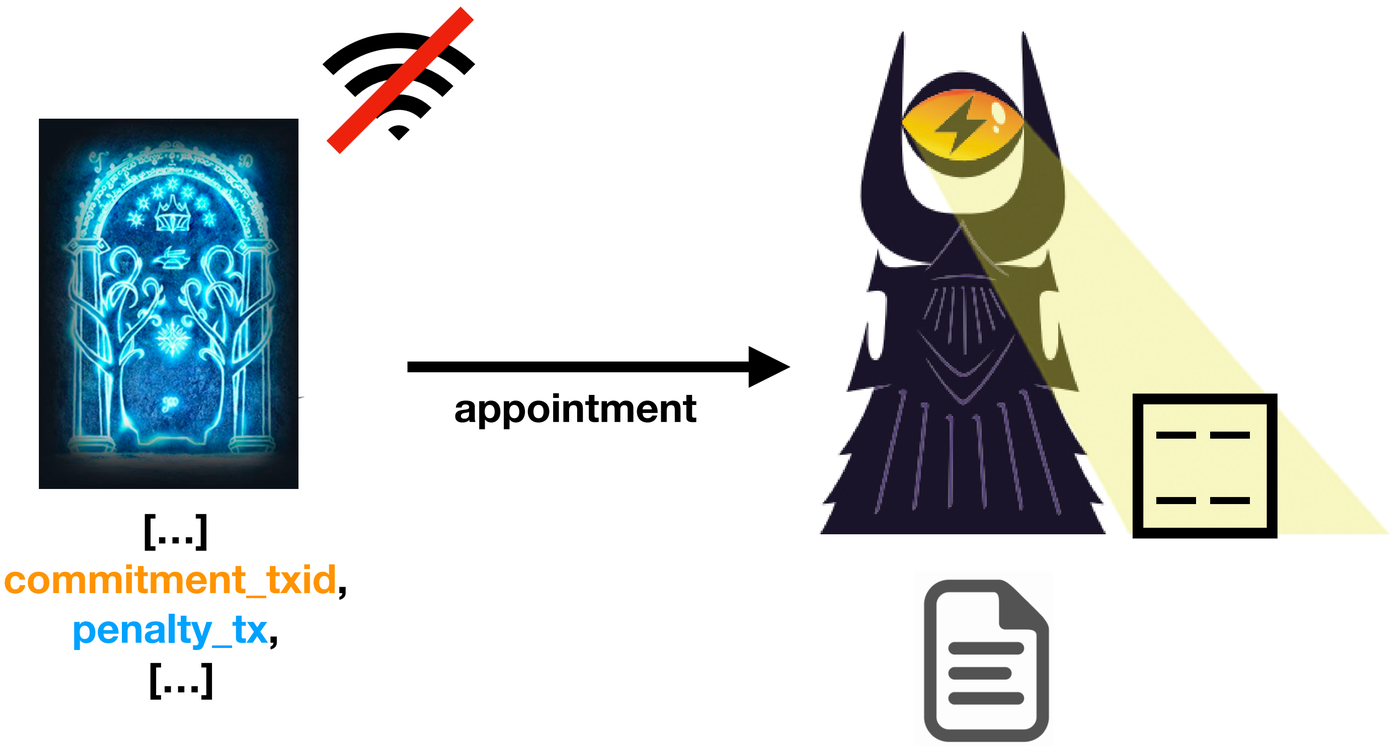
BASIC WATCHTOWER PROTOCOL

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

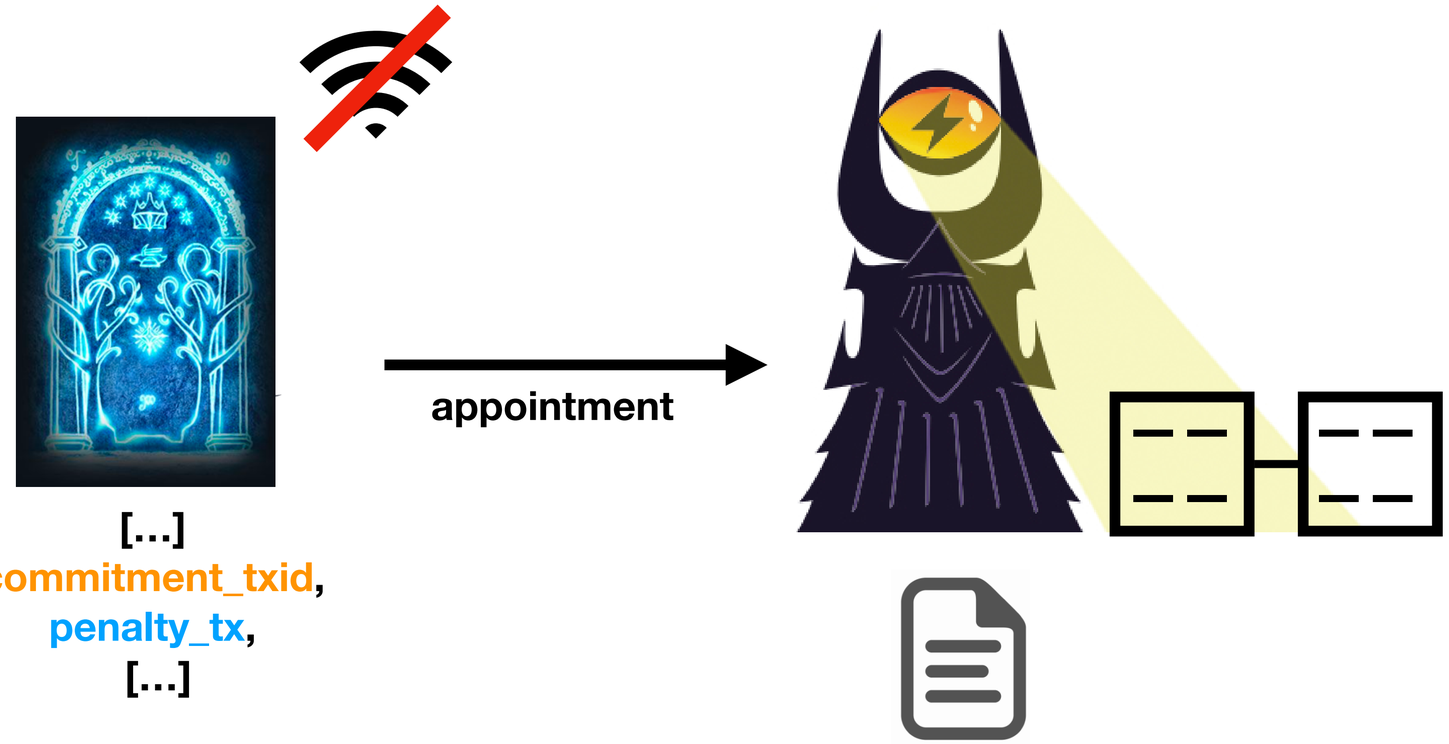
appointment



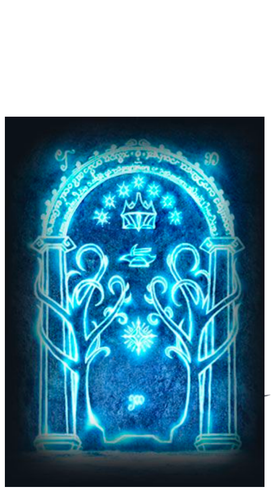
BASIC WATCHTOWER PROTOCOL



BASIC WATCHTOWER PROTOCOL

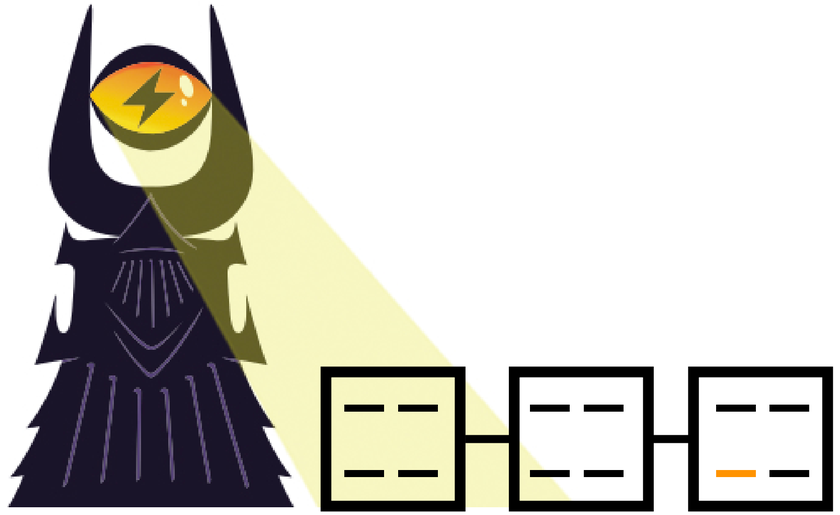


BASIC WATCHTOWER PROTOCOL

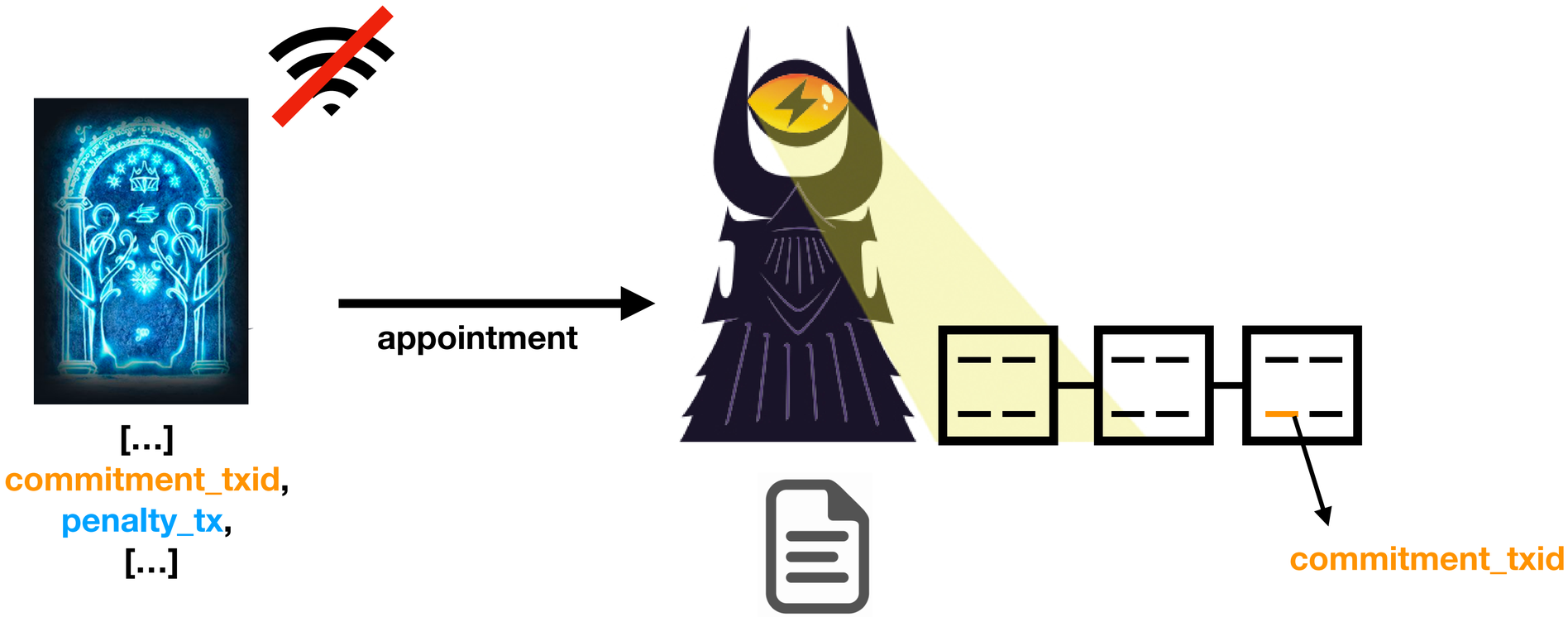


[...]
commitment_txid,
penalty_tx,
[...]

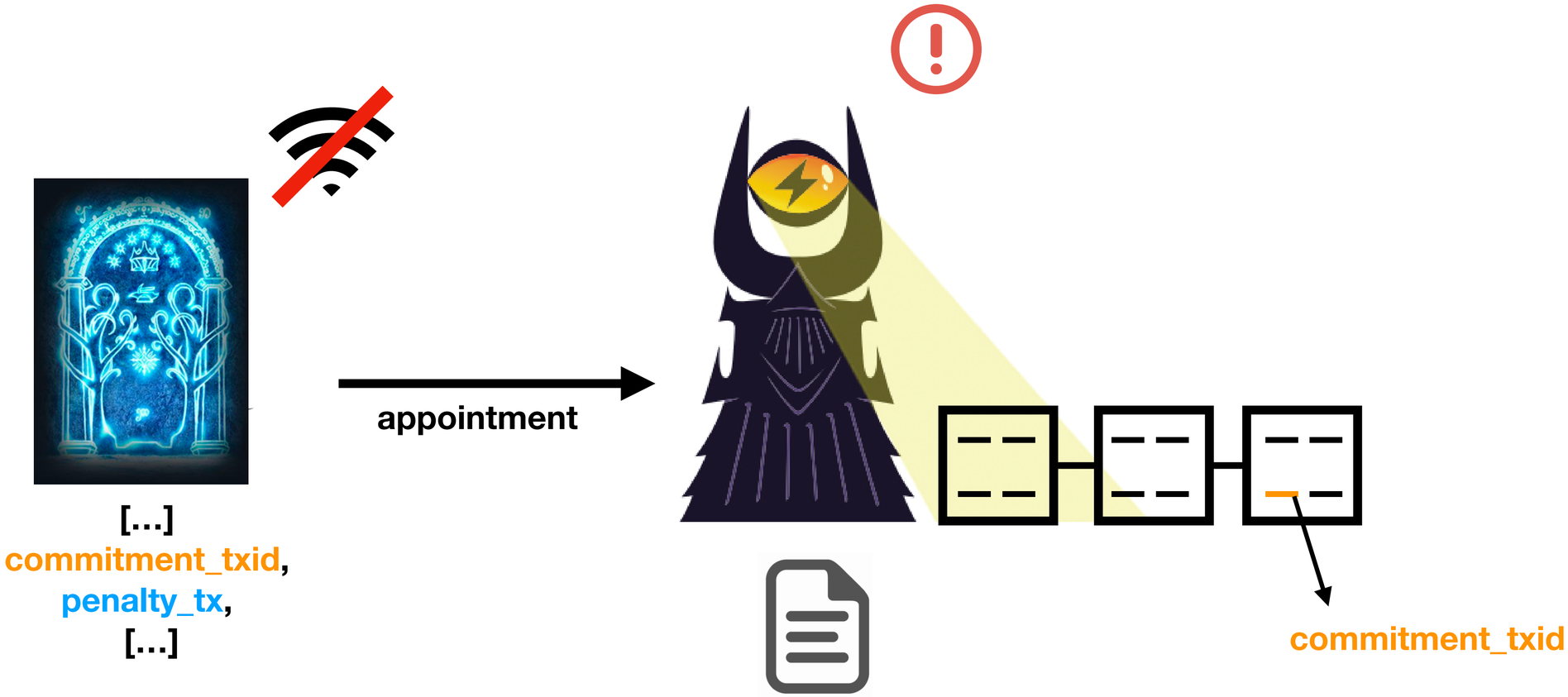
appointment



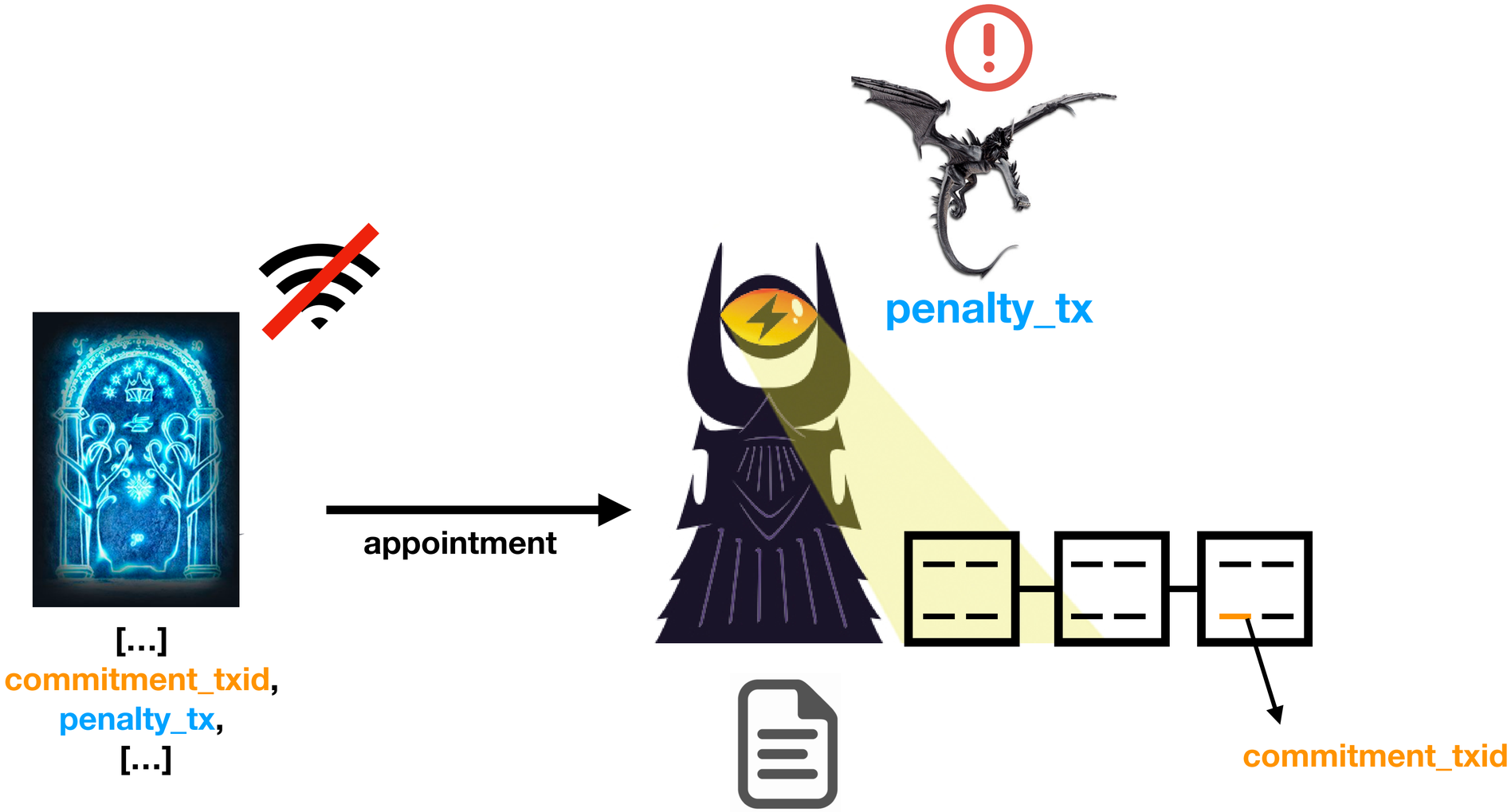
BASIC WATCHTOWER PROTOCOL



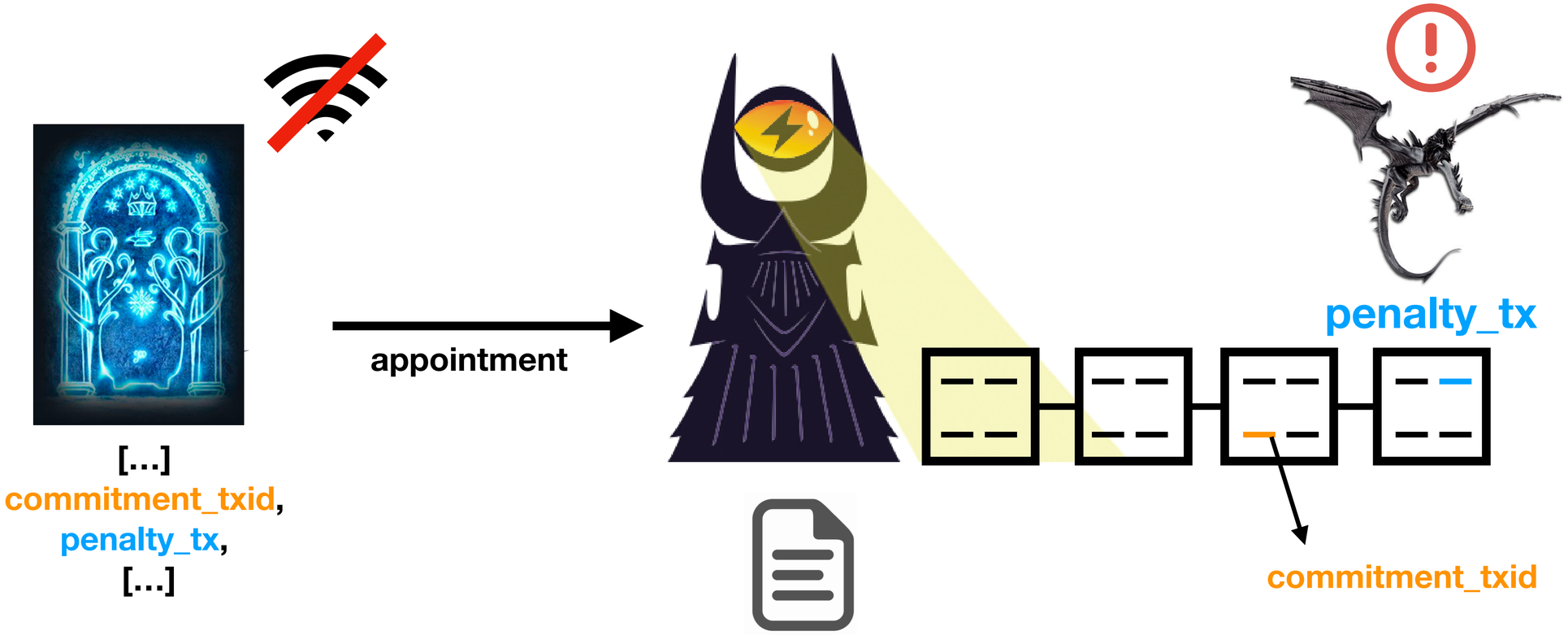
BASIC WATCHTOWER PROTOCOL



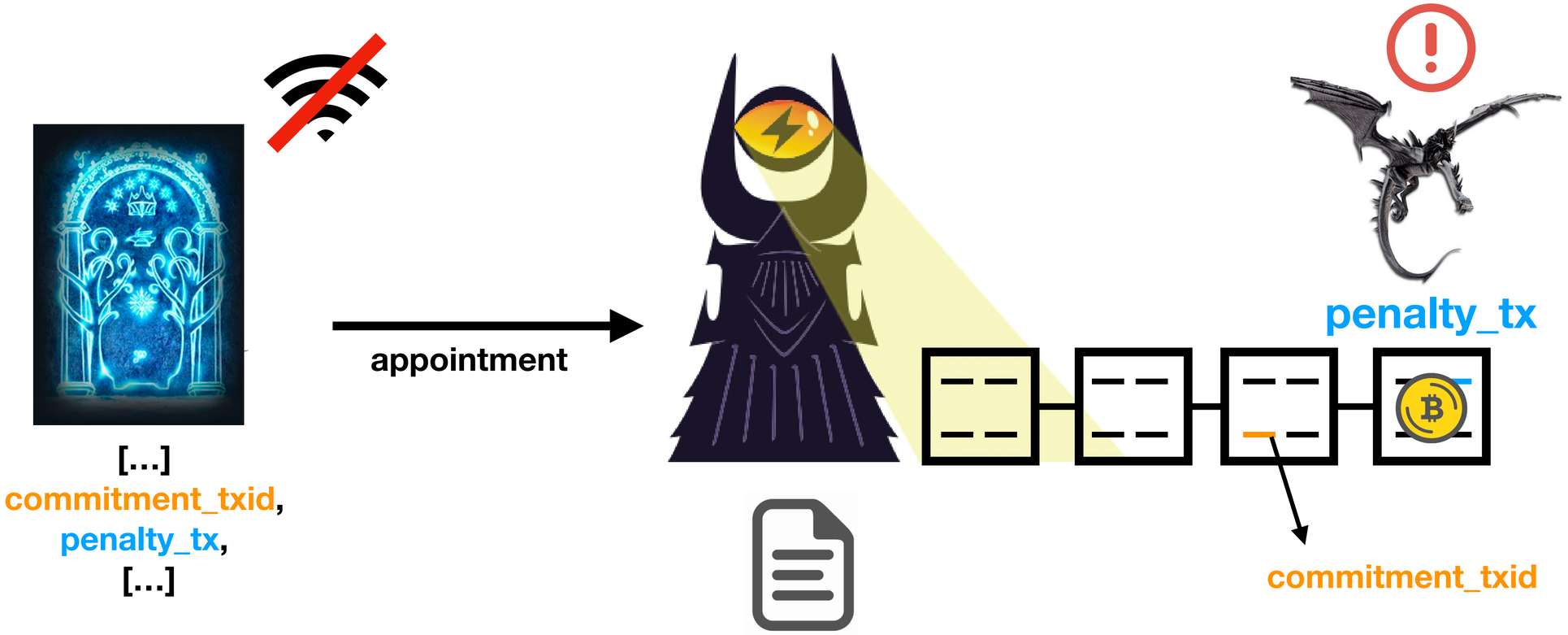
BASIC WATCHTOWER PROTOCOL



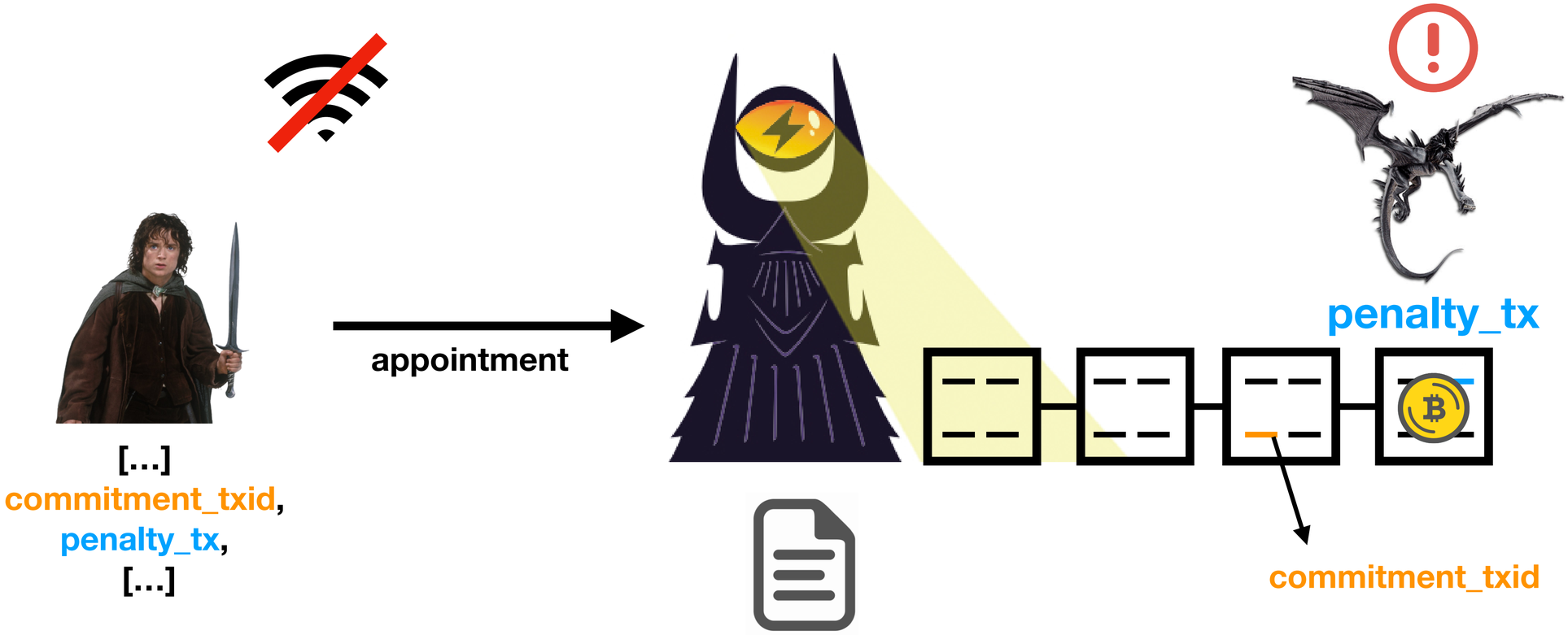
BASIC WATCHTOWER PROTOCOL



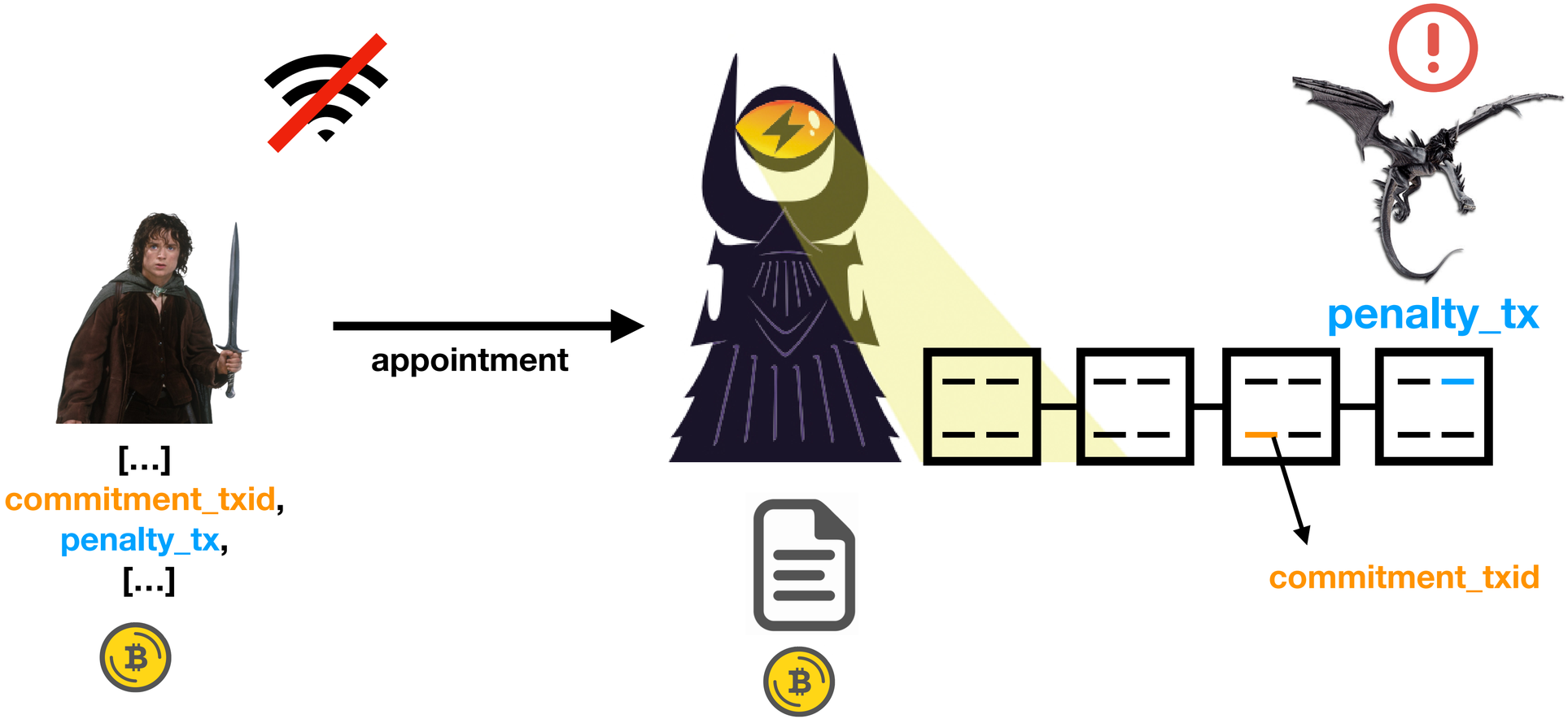
BASIC WATCHTOWER PROTOCOL



BASIC WATCHTOWER PROTOCOL



BASIC WATCHTOWER PROTOCOL



WATCHTOWERS: HOW DO THEY WORK (I)

For every channel update:

- The penalty transaction is **encrypted** under a key derived from the commitment transaction id **(sk and iv)**
- 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

WATCHTOWERS: HOW DO THEY WORK (II)

User side

WATCHTOWERS: HOW DO THEY WORK (II)

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

WATCHTOWERS: HOW DO THEY WORK (II)

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

WATCHTOWERS: HOW DO THEY WORK (II)

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB

WATCHTOWERS: HOW DO THEY WORK (II)

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → **locator**

WATCHTOWERS: HOW DO THEY WORK (II)

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → **locator**

cipher = CHACHA20POLY1305

sk = SHA256(commitment_txid)

IV = 0

WATCHTOWERS: HOW DO THEY WORK (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) →

WATCHTOWERS: HOW DO THEY WORK (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**

WATCHTOWERS: HOW DO THEY WORK (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**

WATCHTOWERS: HOW DO THEY WORK (II)

User side

penalty_tx = 020000000001010d8b7512b1f530338ca886…1f9624914fb8a6800000000

commitment_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → **locator**

APPOINTMENT (SEND TO TOWER)

cipher = CHACHA20POLY1305

sk = SHA256(commitment_txid)

IV = 0

encrypt (penalty_tx, sk, IV) →

encrypted blob

WATCHTOWERS: HOW DO THEY WORK (III)

Tower side

WATCHTOWERS: HOW DO THEY WORK (III)

Tower side

for every transaction_id in every block

locator = 16 MSB transaction_id

WATCHTOWERS: HOW DO THEY WORK (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

WATCHTOWERS: HOW DO THEY WORK (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)

WATCHTOWERS: HOW DO THEY WORK (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**

RECAP: CONCEPTS

appointment: collection of information sent to the tower from every payment. Contains an encrypted penalty transaction.

locator: user-picked identifier for appointments. Corresponds to the first half of the commitment_txid.

channel breach: a unilateral channel closed using an old commitment_tx.

commitment_txid: txid of a given commitment transaction. May correspond to a channel breach.

penalty_tx: transaction sent in response to a channel breach.

DESIGN TRADEOFFS

(NON CUSTODIAL) TOWER DESIGN TRADEOFFS

PRIVACY

What does the Watchtower know about the node?

ACCESS

Who can use the Watchtower?

STORAGE

What does the Watchtower have to store?

COST

What is the user cost to use the Watchtower?

PRIVACY

NO PRIVACY

The user sends the penalty transaction as clear text

- ✅ Can verify data is a transaction
- ❌ Cannot verify transaction is valid
- ❌ Payment information is leaked
- ❌ Data can be leaked to third parties

HIGH PRIVACY

The user sends an encrypted penalty transaction

- ✅ Data only revealed on a breach
- ✅ Harder to leak your data
- ❌ Cannot verify data is a transaction
- ❌ Heavier computation

PRIVACY

NO PRIVACY

Can sell all your data

HIGH PRIVACY

Can sell data about your breaches

PRIVACY

NO PRIVACY

Can sell all your data

HIGH PRIVACY

Can sell data about your breaches

STORAGE

PRIVACY

NO PRIVACY

Can sell all your data

HIGH PRIVACY

Can sell data about your breaches

STORAGE

Therefore, privacy by design IS a better approach

ACCESS

PRIVATE ACCESS

A limited number of (trusted) users can use the tower

- ✅ No DoS risk
- ✅ Potentially free service
- ❌ Cannot accommodate the whole network

PUBLIC ACCESS

Anyone can use the tower

- ✅ Tower as a service
- ➖ Access control required
- ❌ Paid service (high DoS surface if not properly priced)

ACCESS

PRIVATE ACCESS

A limited number of (trusted) users can use the tower

LOW STORAGE

LOW / NO COST

PUBLIC ACCESS

Anyone can use the tower

- ✅ Tower as a service
- ➖ Access control required
- ❌ Paid service (high DoS surface if not properly priced)

ACCESS

PRIVATE ACCESS

A limited number of (trusted) users can use the tower

LOW STORAGE

LOW / NO COST

PUBLIC ACCESS

Anyone can use the tower

HIGH STORAGE

LOW COST

STORAGE

STORAGE 

The required storage is always going to be big (modulo the number of channel updates).

- Highly linked to price
- Strategies to align the incentives of the user and the tower are required
- ✘ One appointment per channel update

COST

NO COST

Using the tower is free

- ✅ OK for private towers
- ❌ Highly unviable for public towers (highest cost and DoS surface)

LOW COST

The tower charges a fee

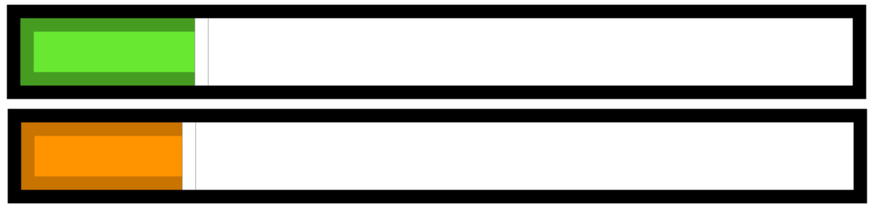
- ✅ High traffic = profit **(if properly priced)**
- ✅ Data can be deleted **(if incentives are aligned)**

COST

NO COST

Using the tower is free

PRIVATE ACCESS AND LOW STORAGE



LOW COST

The tower charges a fee

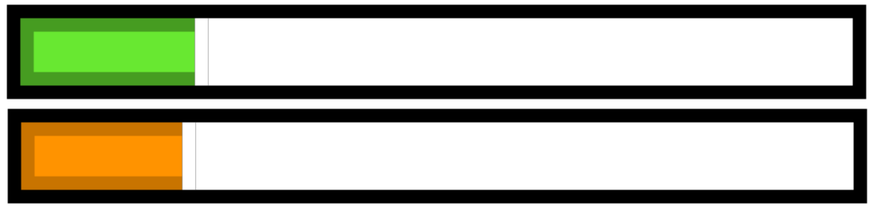
- ✅ High traffic = profit
  (if properly priced)
- ✅ Data can be deleted
  (if incentives are aligned)

COST

NO COST

Using the tower is free

PRIVATE ACCESS AND LOW STORAGE



OR
PUBLIC ACCESS AND &^%$ STORAGE

LOW COST

The tower charges a fee

- ✅ High traffic = profit
  (if properly priced)
- ✅ Data can be deleted
  (if incentives are aligned)

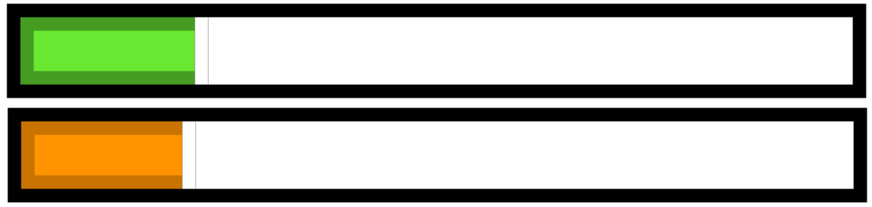
COST

NO COST

LOW COST

Using the tower is free

PRIVATE ACCESS AND LOW STORAGE



OR

PUBLIC ACCESS AND &^%$ STORAGE

The tower charges a fee

HIGH STORAGE

IDEAL WATCHTOWER (NO ELTOO)

PRIVACY		High privacy
ACCESS		Public access
STORAGE		Non-exploitable O(N) storage
COST		Low cost

IDEAL WATCHTOWER (NO ELTOO)

PRIVACY		High privacy
ACCESS		Public access
STORAGE		Non-exploitable O(N) storage
COST		Low cost

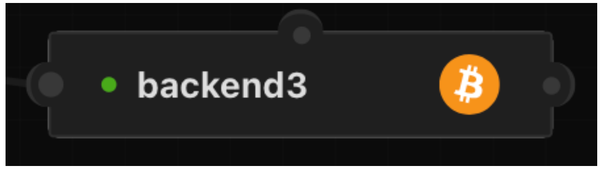
INTEROPERABLE!

THE PRACTICE

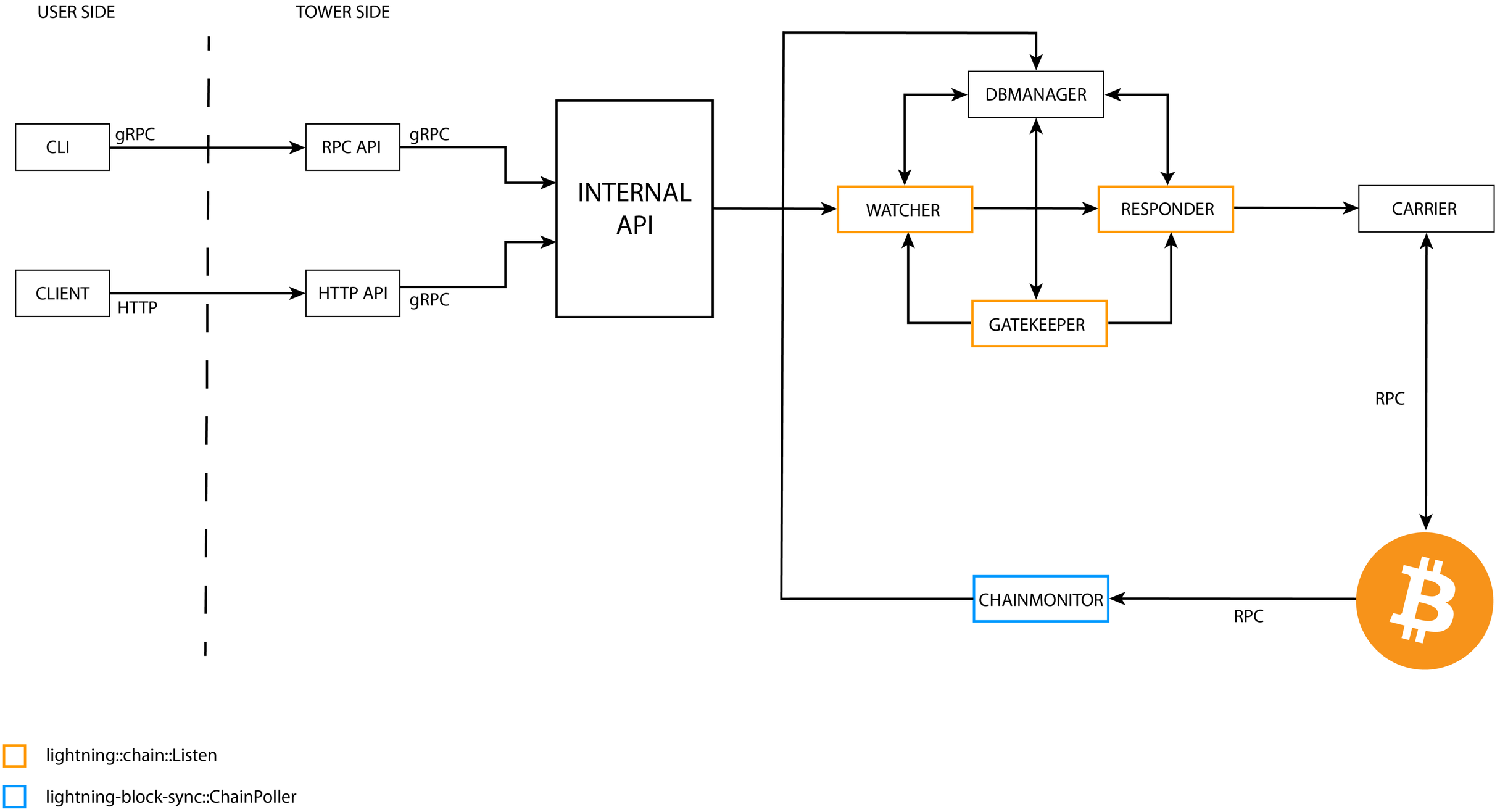
WHAT WILL WE DO?

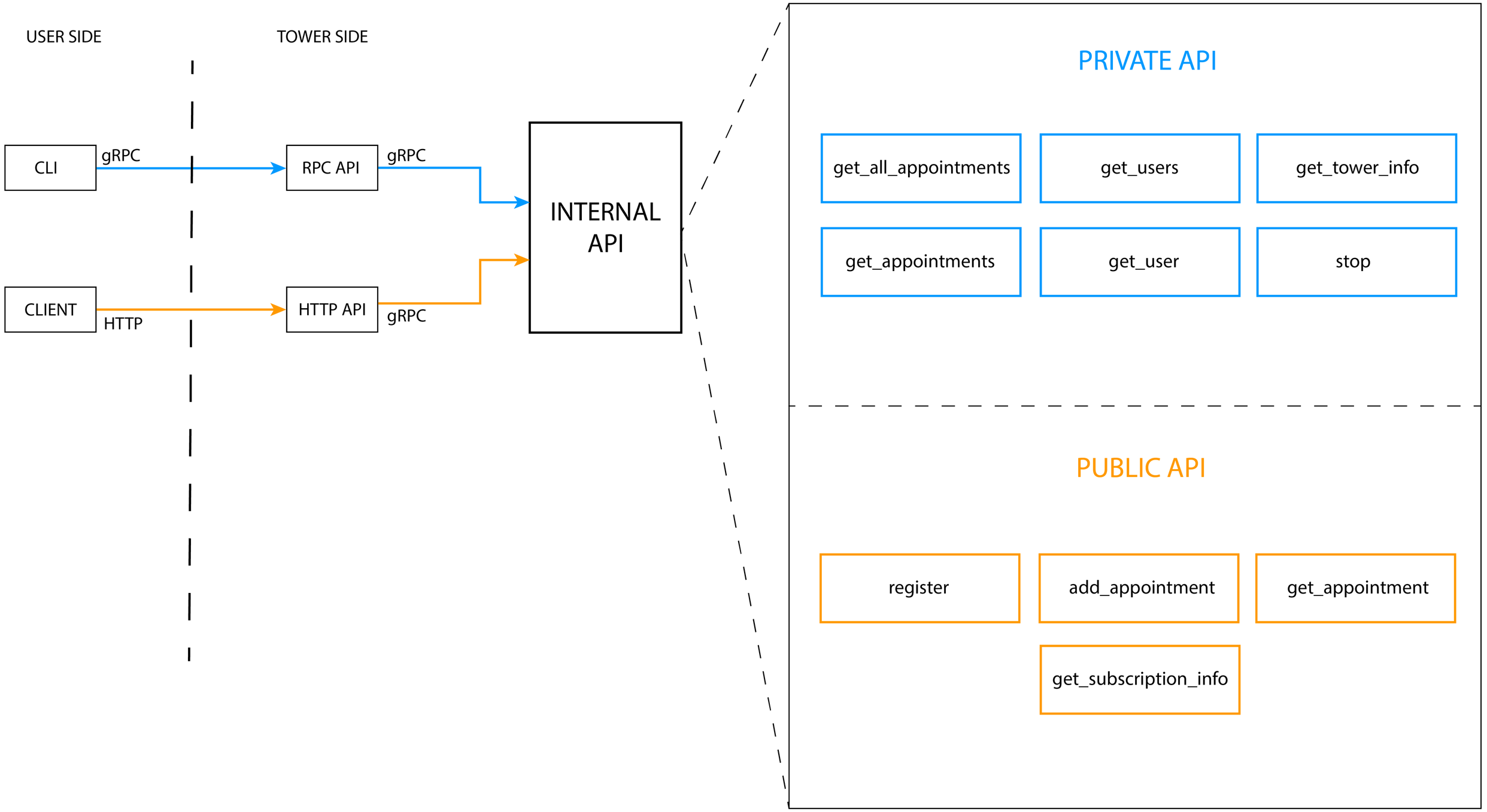
- Install The Eye of Satoshi (`rust-teos`)
- Explore the tower APIs
- Install `watchtower-client` plugin for `CoreLN` and interact with the tower
- Deploy a test network with two `CoreLN` node and see the tower / client
- See a live mainnet watchtower in action

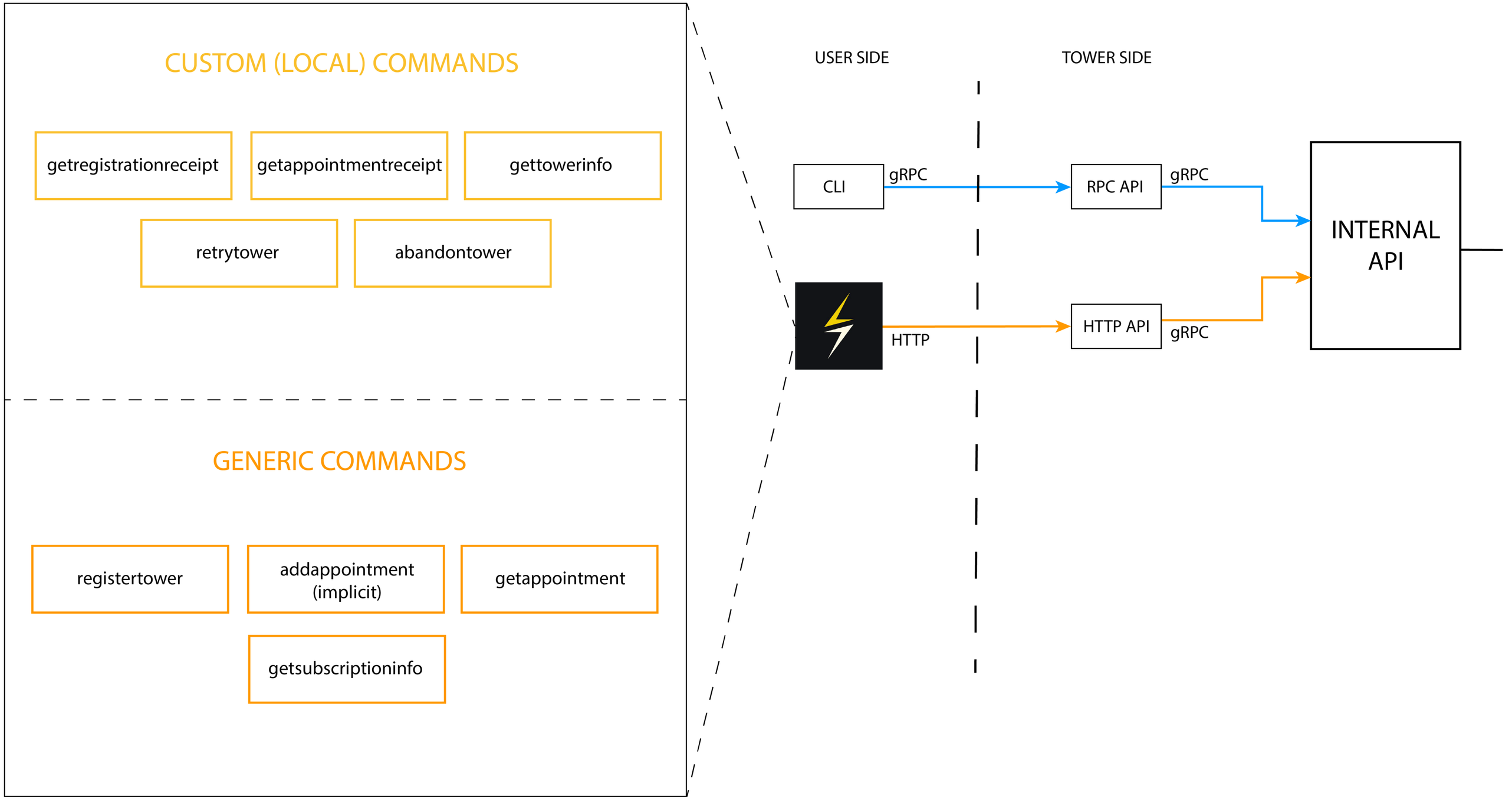

```
# Install system dependencies (as root)
apt-get update && apt-get install build-essential vim tor

# Switch to bitcoin user
su - clightning

# Install rustlang
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh -s -- -y

# Install the plugin
git clone https://github.com/talaia-labs/rust-teos.git && \
cd rust-teos && \
cargo install --locked --path watchtower-plugin

# Create the plugins folder and link the plugin
mkdir -p .lightning/plugins && \
cd .lightning/plugins && \
ln -s/home/clightning/.cargo/bin/watchtower-client
```




```
# Install system dependencies (as root)
apt-get update && apt-get install git build-essential net-tools vim tor

# Switch to bitcoin user
su - bitcoin

# Install rustlang
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh -s -- -y

# Install the tower backend
git clone https://github.com/talaia-labs/rust-teos.git && \
cd rust-teos && \
cargo install --locked --path teos
```


USER SIDE
TOWER SIDE
CLI
gRPC
RPC API
gRPC
CLIENT
HTTP
HTTP API
gRPC
INTERNAL API
DBMANAGER
WATCHER
RESPONDER
CARRIER
GATEKEEPER
RPC
CHAINMONITOR
RPC
lightning::chain::Listen
lightning-block-sync::ChainPoller


USER SIDE
TOWER SIDE
CLI
gRPC
RPC API
gRPC
INTERNAL API
CLIENT
HTTP
HTTP API
gRPC
PRIVATE API
get_all_appointments
get_users
get_tower_info
get_appointments
get_user
stop
PUBLIC API
register
add_appointment
get_appointment
get_subscription_info


CUSTOM (LOCAL) COMMANDS
getregistrationreceipt
getappointmentreceipt
gettowerinfo
retrytower
abandontower
GENERIC COMMANDS
registertower
addappointment
(implicit)
getappointment
getsubscriptioninfo
USER SIDE
TOWER SIDE
CLI
gRPC
RPC API
gRPC
INTERNAL
API
HTTP
HTTP API
gRPC

QUESTIONS

sr_gi

Talaia Labs

BONUS TRACK

REVENUE MODELS

BOUNTY APPROACH

The tower is paid only if a breach happens and the penalty makes it to the chain	✅	❌
Multiple towers can be hired for the price of one	✅	❌
The tower **can** use CPFP to bump the fee of the penalty transaction	✅	✅
It’s easy to spam/DoS the tower with junk	➖	❌

SUBSCRIPTION APPROACH

The tower is paid beforehand, even if the it does not responds to the breach	✗	✓
A rational user will only hire so many towers	–	✓
The tower **cannot** use CPFP to bump the fee of the penalty transaction	✗	✗
Spamming the tower has a cost	–	✓
Exploiting requires paying for a subscription	–	✓

SUBSCRIPTION VS BOUNTY

BOTH MODELS HAVE THEIR PROS AND CONS…

SUBSCRIPTION VS BOUNTY

BOTH MODELS HAVE THEIR PROS AND CONS…

BOUNTY & SUBSCRIPTION

The tower is paid **a fraction of the cost** beforehand, the rest is paid as a bounty	—	—
A rational user will only hire so many towers	—	✓
The tower **can** use CPFP to bump the fee of the penalty transaction	✓	✓
Spamming the tower has a cost	—	✓
Exploiting requires paying for a subscription	—	✓

ATTACKS ON TOWERS

ATTACKS ON TOWERS

ATTACKS ON TOWERS

ATTACKS ON TOWERS

ATTACKS ON TOWERS

Bounty approach

ATTACKS ON TOWERS

Bounty approach

ATTACKS ON TOWERS

Bounty approach

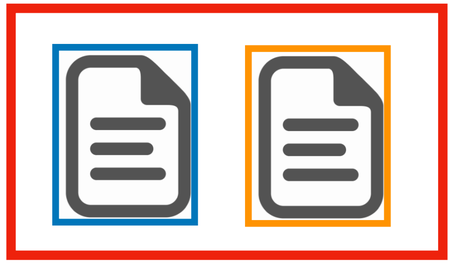
ATTACKS ON TOWERS

Bounty approach

ATTACKS ON TOWERS

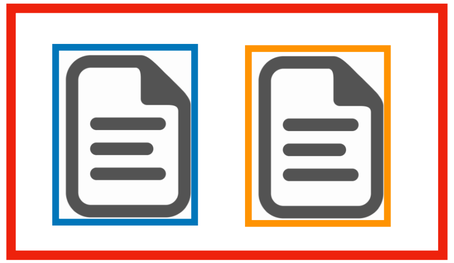
Bounty approach

ATTACKS ON TOWERS



Bounty approach

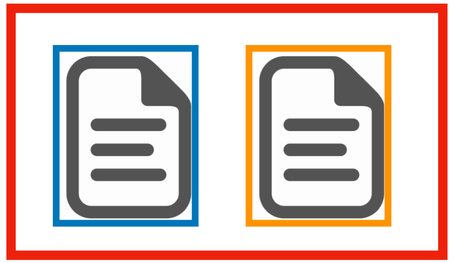
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

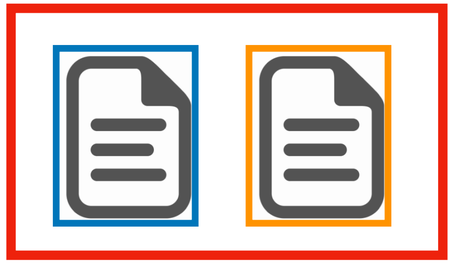
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

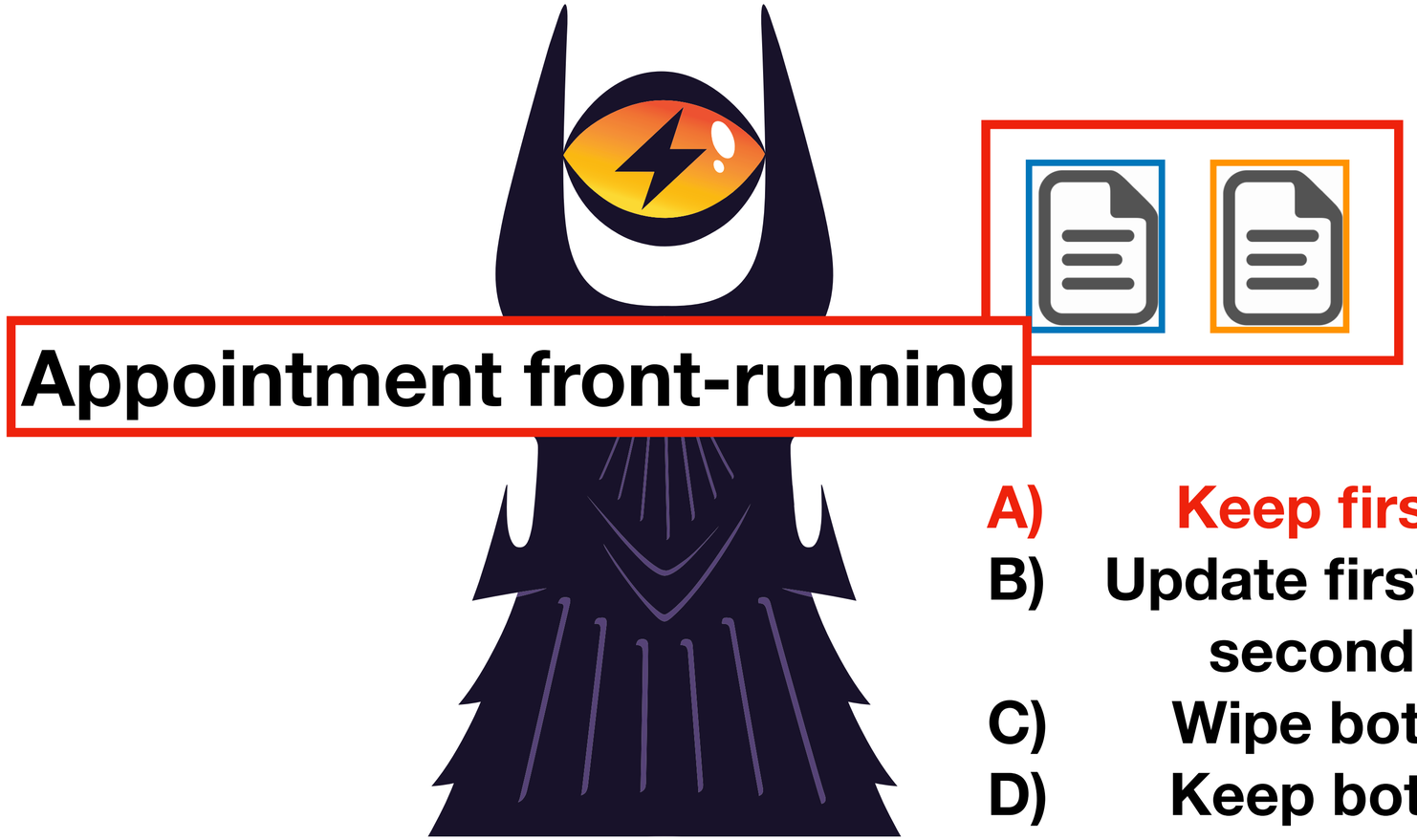
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

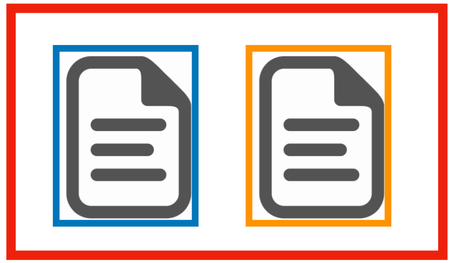
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

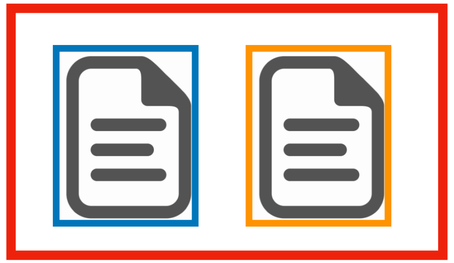
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

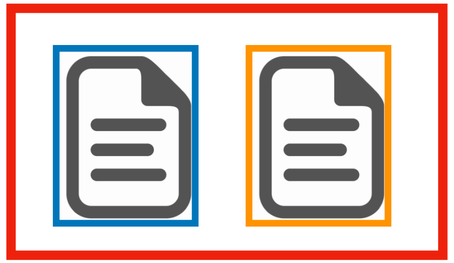
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

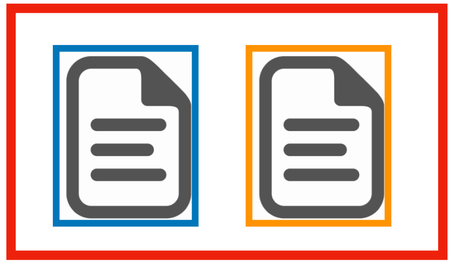
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

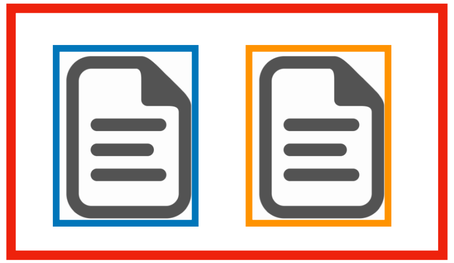
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

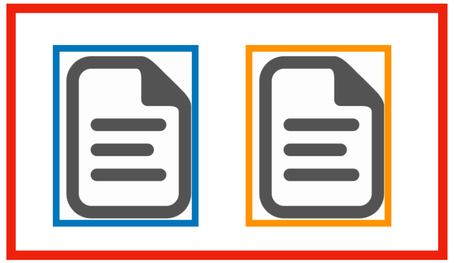
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

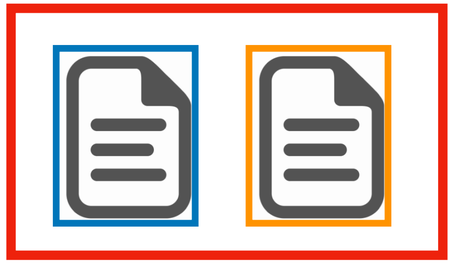
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

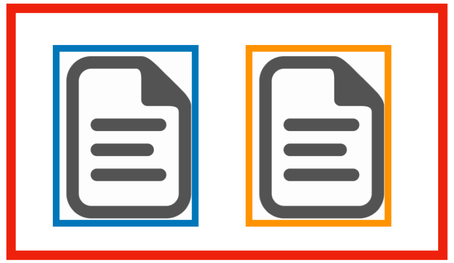
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

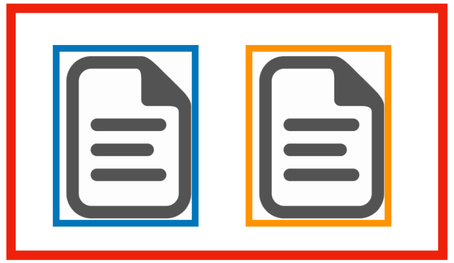
ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach

ATTACKS ON TOWERS



A) Keep first
B) Update first w/ second
C) Wipe both
D) Keep both

Bounty approach