

# Bitcoin: From Zero to Hero

---

Sergi Delgado



# **Bitcoin transactions**

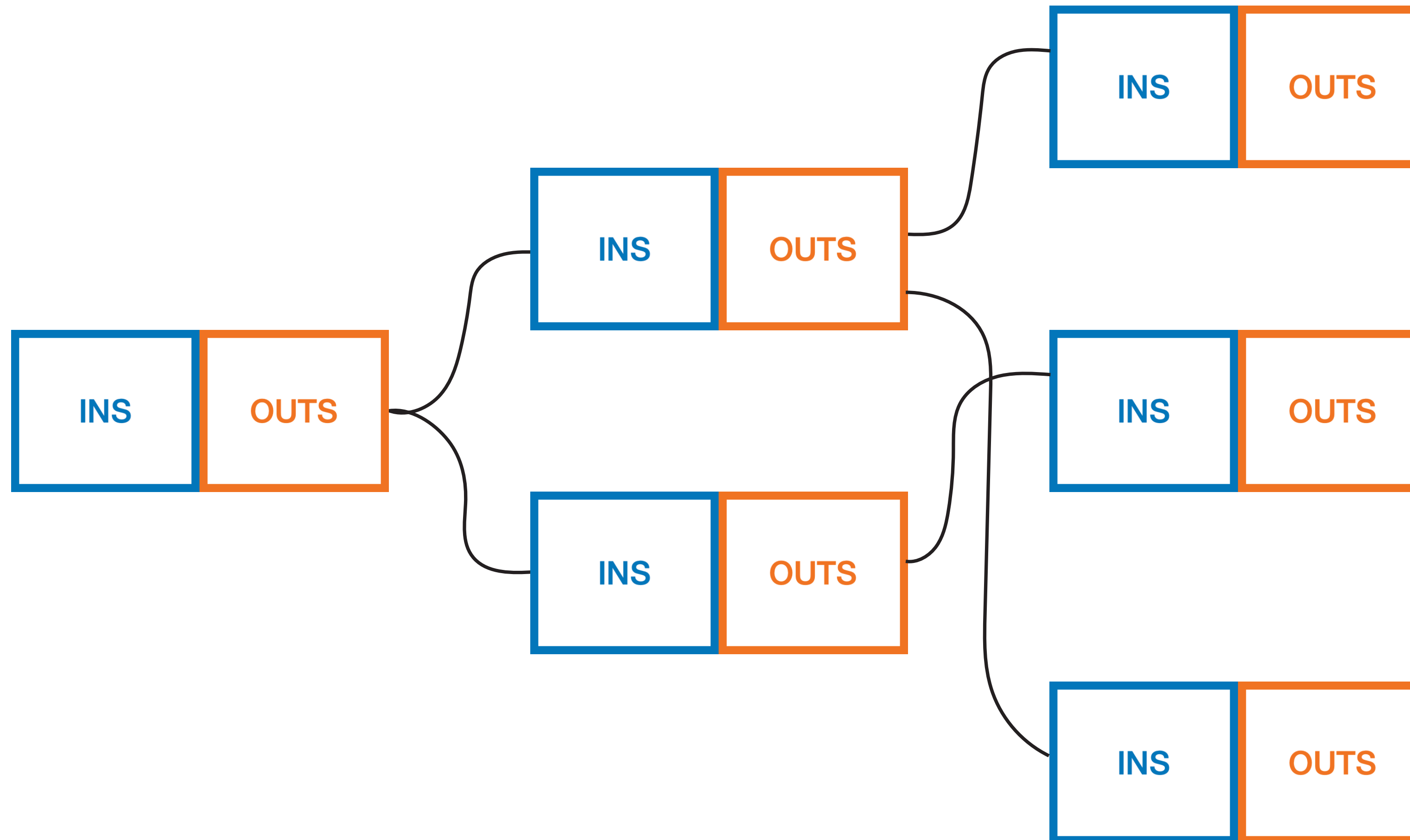


**¿Qué es una transacción de Bitcoin?**

# Bitcoin transactions



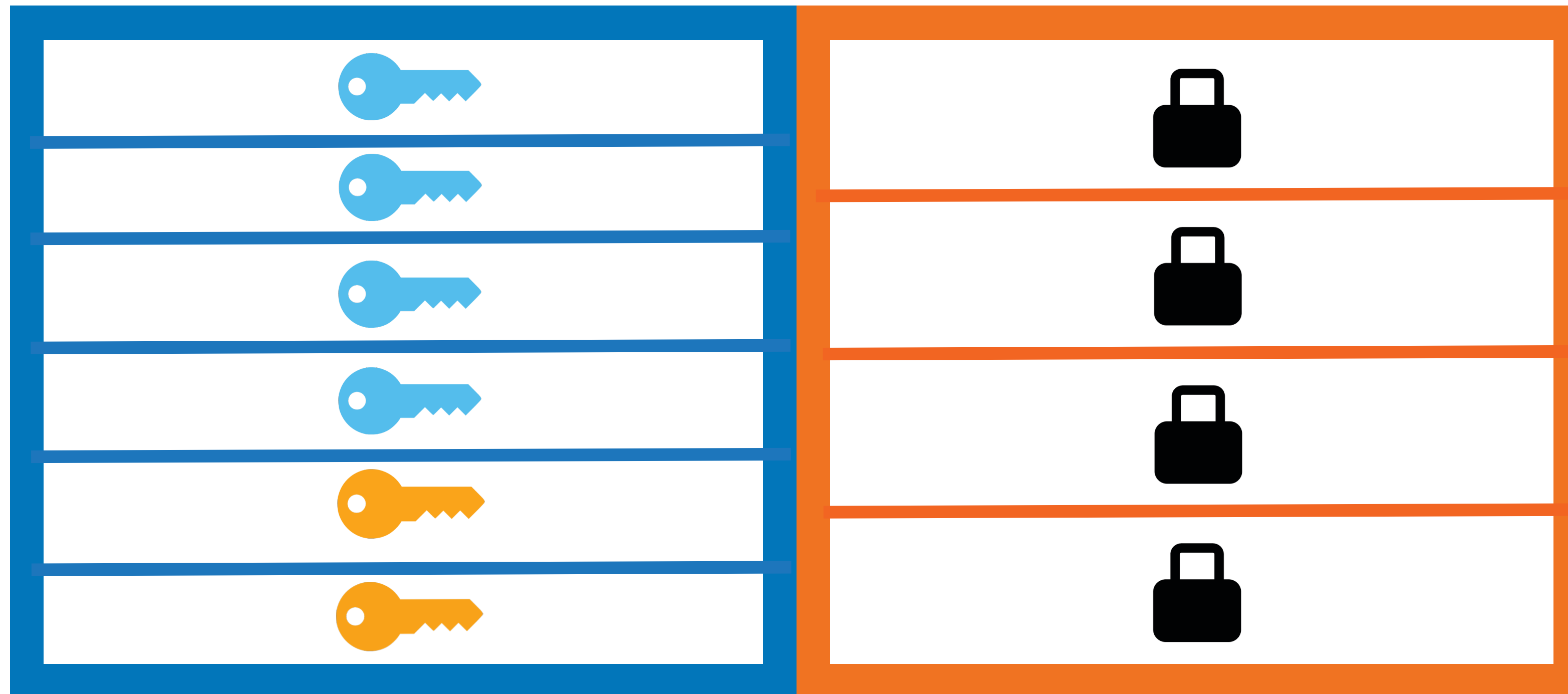
¿Qué es una transacción de Bitcoin?



# Bitcoin transactions



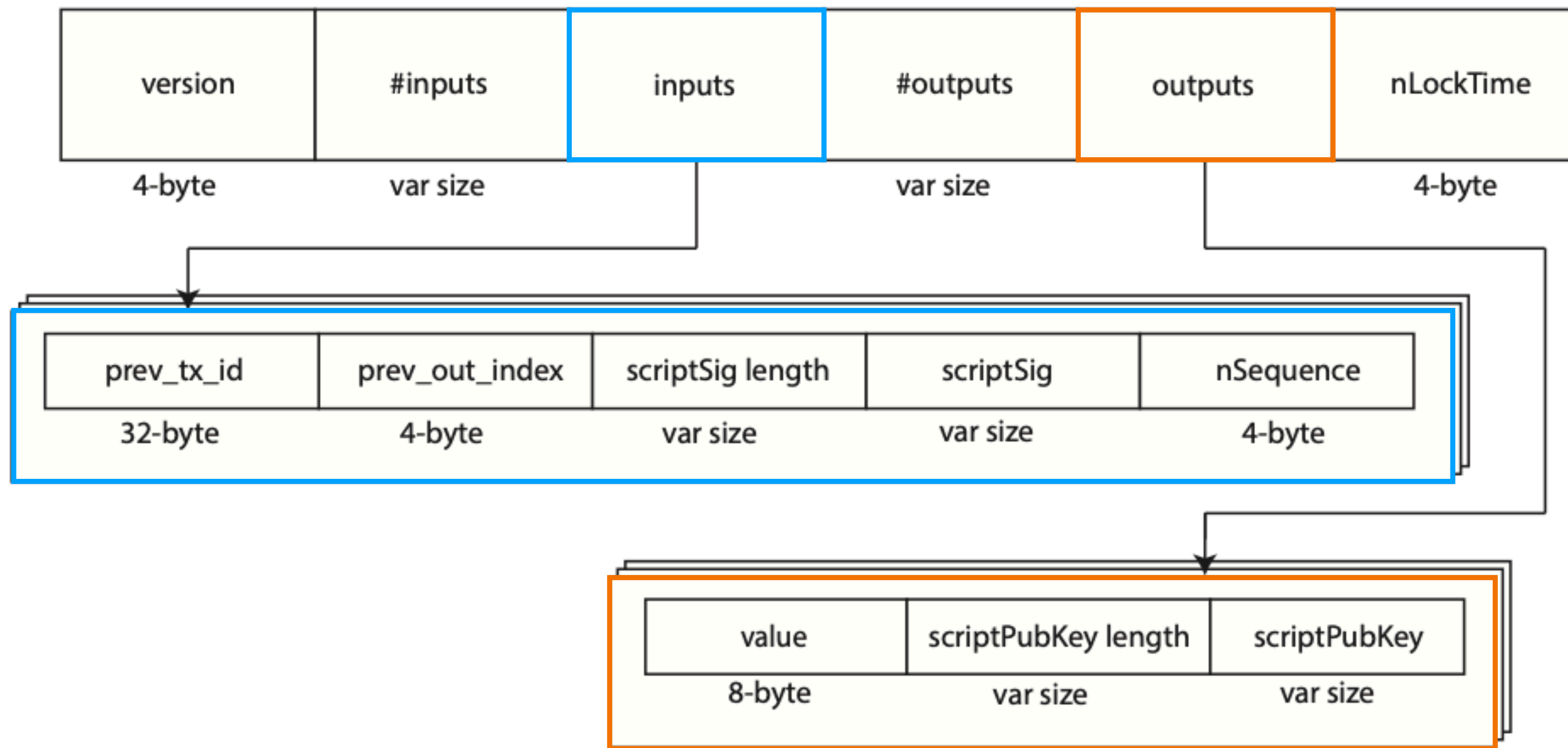
¿Qué es una transacción de Bitcoin?



# Bitcoin transactions



¿Qué es una transacción de Bitcoin?



# **Limitaciones de Bitcoin**

**¿Qué limitaciones tiene Bitcoin?**

# **Limitaciones de Bitcoin**

**¿Qué limitaciones tiene Bitcoin?**

- Espacio de bloque finito (~7 ttps)

# **Limitaciones de Bitcoin**

## **¿Qué limitaciones tiene Bitcoin?**

- Espacio de bloque finito (~7 ttps)
- Comisiones (fees) por tamaño de transacción, no por valor



# **Limitaciones de Bitcoin**

## **¿Qué limitaciones tiene Bitcoin?**

- Espacio de bloque finito (~7 ttps)
- Comisiones (fees) por tamaño de transacción, no por valor
  - Existe un valor mínimo de fee

# **Limitaciones de Bitcoin**

## **¿Qué limitaciones tiene Bitcoin?**

- Espacio de bloque finito (~7 ttps)
- Comisiones (fees) por tamaño de transacción, no por valor
  - Existe un valor mínimo de fee
- Tiempo de confirmación relativamente elevado (~10 min/conf)

# Limitaciones de Bitcoin

## ¿Qué limitaciones tiene Bitcoin?

- Espacio de bloque finito (~7 ttps)
- Comisiones (fees) por tamaño de transacción, no por valor
  - Existe un valor mínimo de fee
- Tiempo de confirmación relativamente elevado (~10 min/conf)
- Expresividad del lenguaje limitada

# Limitaciones de Bitcoin

## ¿Qué limitaciones tiene Bitcoin?

- Espacio de bloque finito (~7 ttps)
- Comisiones (fees) por tamaño de transacción, no por valor
  - Existe un valor mínimo de fee
- Tiempo de confirmación relativamente elevado (~10 min/conf)
- Expresividad del lenguaje limitada
- ...

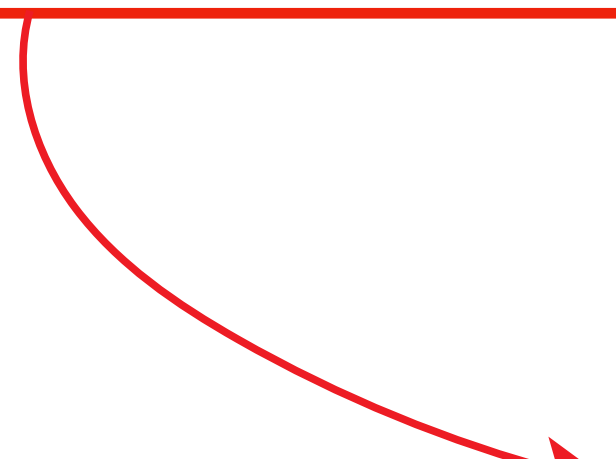
# Limitaciones de Bitcoin

## ¿Qué limitaciones tiene Bitcoin?

- Espacio de bloque finito (~7 ttps)
- Comisiones (fees) por tamaño de transacción, no por valor
  - Existe un valor mínimo de fee
- Tiempo de confirmación relativamente elevado (~10 min/conf)
- Expresividad del lenguaje limitada
- ...

# Limitaciones de Bitcoin

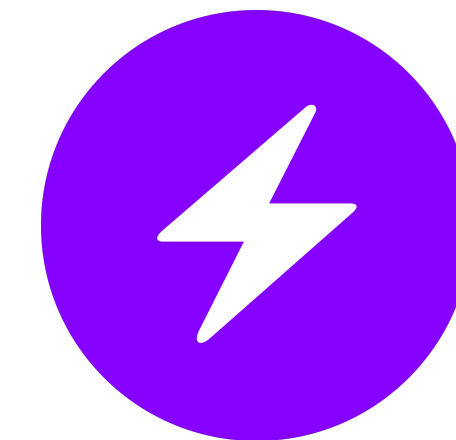
## ¿Qué limitaciones tiene Bitcoin?

- Espacio de bloque finito (~7 ttps)
  - Comisiones (fees) por tamaño de transacción, no por valor
    - Existe un valor mínimo de fee
  - Tiempo de confirmación relativamente elevado (~10 min/conf)
  - Expresividad del lenguaje limitada
  - ...
- 

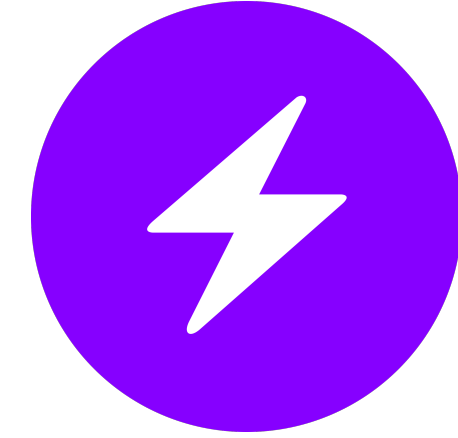
# Limitaciones de Bitcoin

## ¿Qué limitaciones tiene Bitcoin?

- Espacio de bloque finito (~7 ttps)
- Comisiones (fees) por tamaño de transacción, no por valor
  - Existe un valor mínimo de fee
- Tiempo de confirmación relativamente elevado (~10 min/conf)
- Expresividad del lenguaje limitada
- ...



# The Lightning Network



**¿Qué nos permite Lightning que no es posible en Bitcoin?**



# The Lightning Network

**¿Qué nos permite Lightning que no es posible en Bitcoin?**

- Transaccionar sin necesidad de direcciones de Bitcoin

# The Lightning Network

**¿Qué nos permite Lightning que no es posible en Bitcoin?**

- Transaccionar sin necesidad de direcciones de Bitcoin
- Reducir las comisiones prácticamente a zero

# The Lightning Network

**¿Qué nos permite Lightning que no es posible en Bitcoin?**

- Transaccionar sin necesidad de direcciones de Bitcoin
- Reducir las comisiones prácticamente a zero
- Realizar micro pagos

# The Lightning Network

**¿Qué nos permite Lightning que no es posible en Bitcoin?**

- Transaccionar sin necesidad de direcciones de Bitcoin
- Reducir las comisiones prácticamente a zero
- Realizar micro pagos
- Confirmación prácticamente instantánea

# The Lightning Network

**¿Qué nos permite Lightning que no es posible en Bitcoin?**

- Transaccionar sin necesidad de direcciones de Bitcoin
- Reducir las comisiones prácticamente a zero
- Realizar micro pagos
- Confirmación prácticamente instantánea
- tps virtualmente infinito (o al menos varias ordenes de magnitud por encima de onchain)

# The Lightning Network

**¿Qué nos permite Lightning que no es posible en Bitcoin?**

- Transaccionar sin necesidad de direcciones de Bitcoin
- Reducir las comisiones prácticamente a zero
- Realizar micro pagos
- Confirmación prácticamente instantánea
- tps virtualmente infinito (o al menos varias ordenes de magnitud por encima de onchain)
- ...

# Funcionamiento de Lightning

**¿Qué tres pasos constituyen el ciclo de vida de un canal Lightning?**

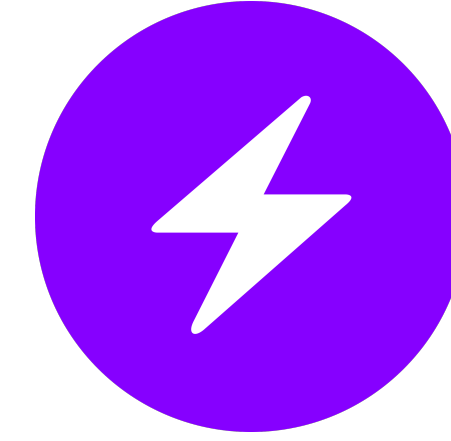
# Funcionamiento de Lightning

**¿Qué tres pasos constituyen el ciclo de vida de un canal Lightning?**

- **Apertura del canal (1x funding transaction)**



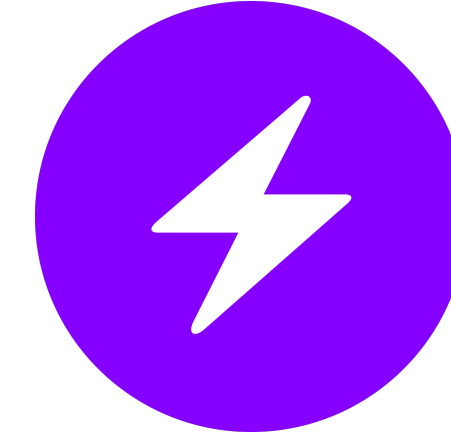
# Funcionamiento de Lightning



¿Qué tres pasos constituyen el ciclo de vida de un canal Lightning?

- Apertura del canal (**1x funding transaction**)
- Fase de operación del canal (**Nx commitment transactions**)

# Funcionamiento de Lightning



¿Qué tres pasos constituyen el ciclo de vida de un canal Lightning?

- Apertura del canal (**1x funding transaction**)
- Fase de operación del canal (**Nx commitment transactions**)
- Cierre del canal (**1x closing transaction**)

# Apertura del canal (funding transaction)



# Apertura del canal (funding transaction)



# Apertura del canal (funding transaction)

Transacción 2-2 MultiSig



# Apertura del canal (funding transaction)

## Transacción 2-2 MultiSig

- Dos entradas (una por parte) y una salida (compartida)



# Apertura del canal (funding transaction)

## Transacción 2-2 MultiSig

- Dos entradas (una por parte) y una salida (compartida)
- A nivel practico, en la mayoría de casos el funding es unidireccional (una entrada y una salida)



# Apertura del canal (funding transaction)

## Transacción 2-2 MultiSig

- Dos entradas (una por parte) y una salida (compartida)
- A nivel practico, en la mayoría de casos el funding es unidireccional (una entrada y una salida)
- Para gastar la saluda generada se necesita firmas de las dos partes





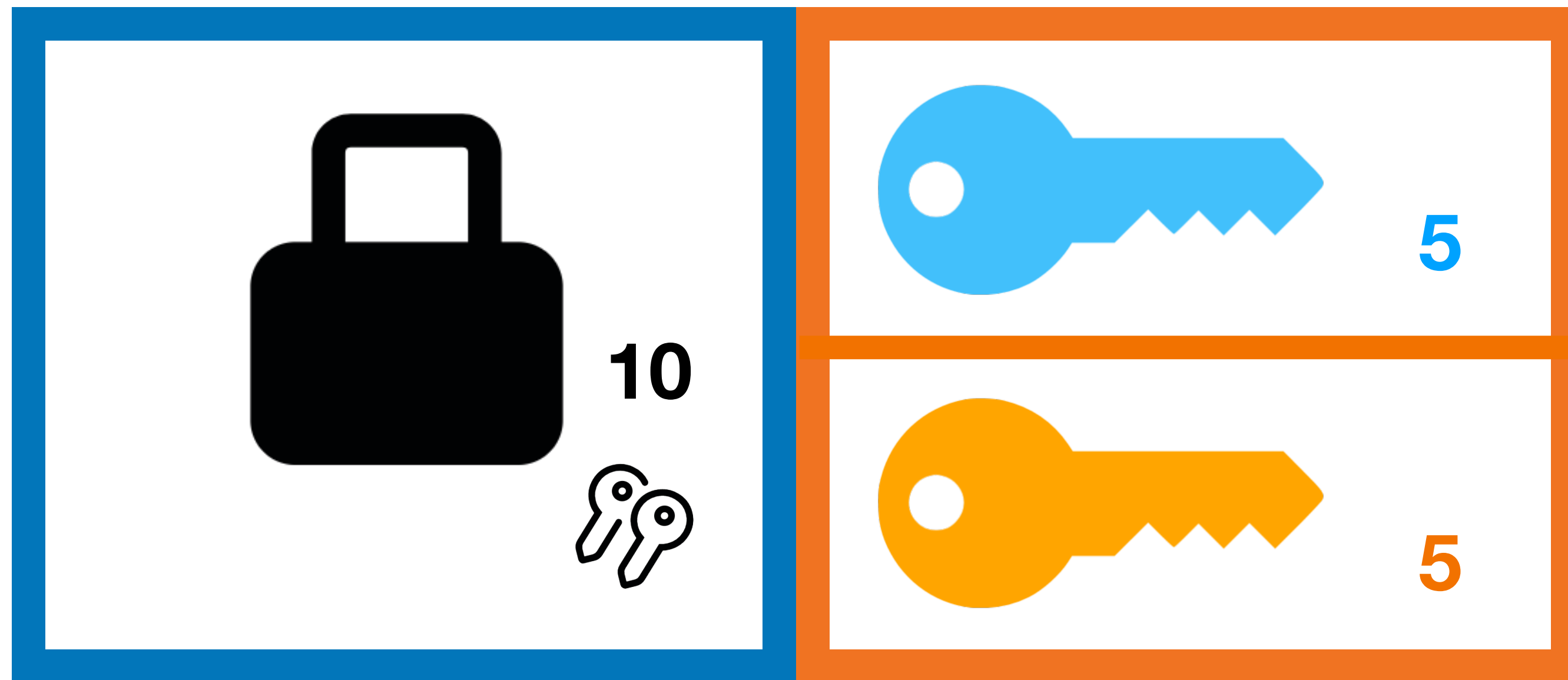
# Apertura del canal (funding transaction)

## Transacción 2-2 MultiSig

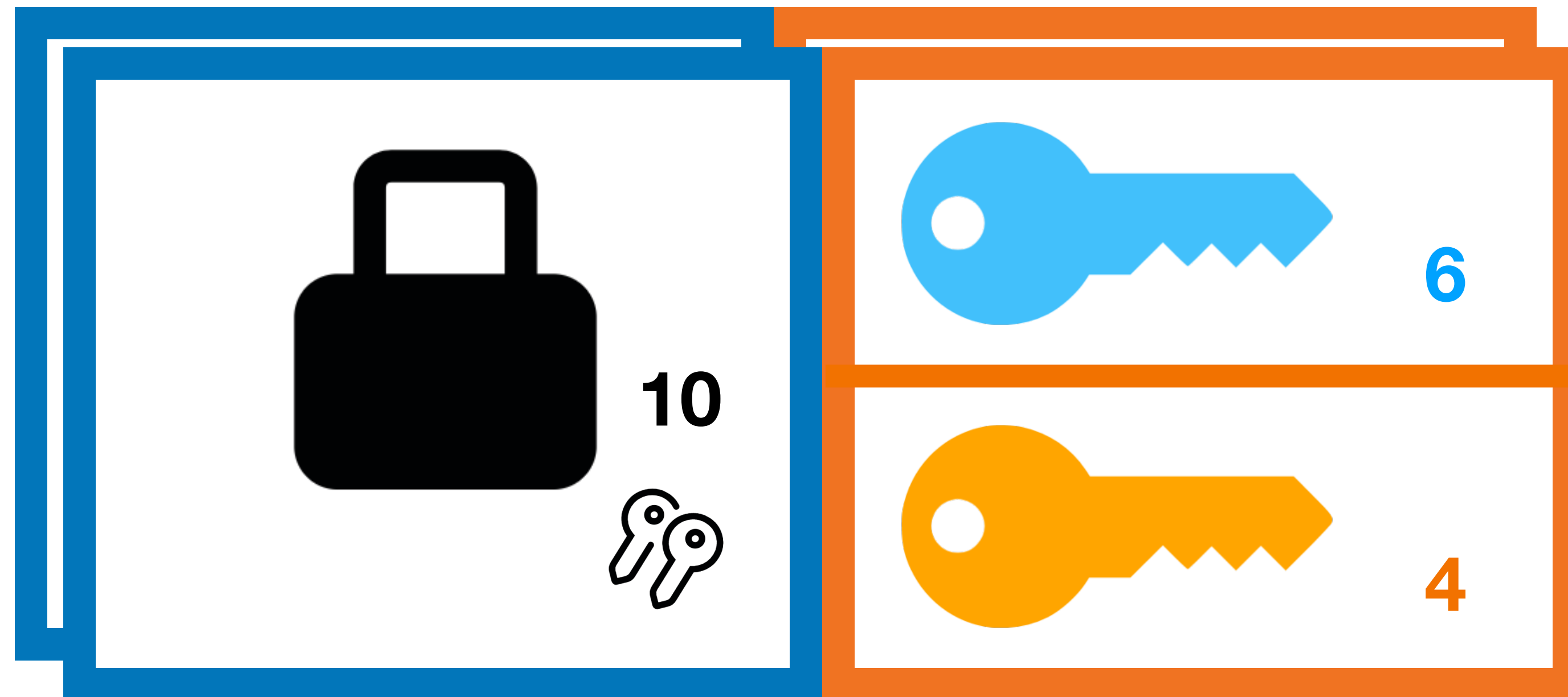
- Dos entradas (una por parte) y una salida (compartida)
- A nivel practico, en la mayoría de casos el funding es unidireccional (una entrada y una salida)
- Para gastar la saluda generada se necesita firmas de las dos partes
- Confirmada on-chain antes de empezar a operar el canal (normalmente)



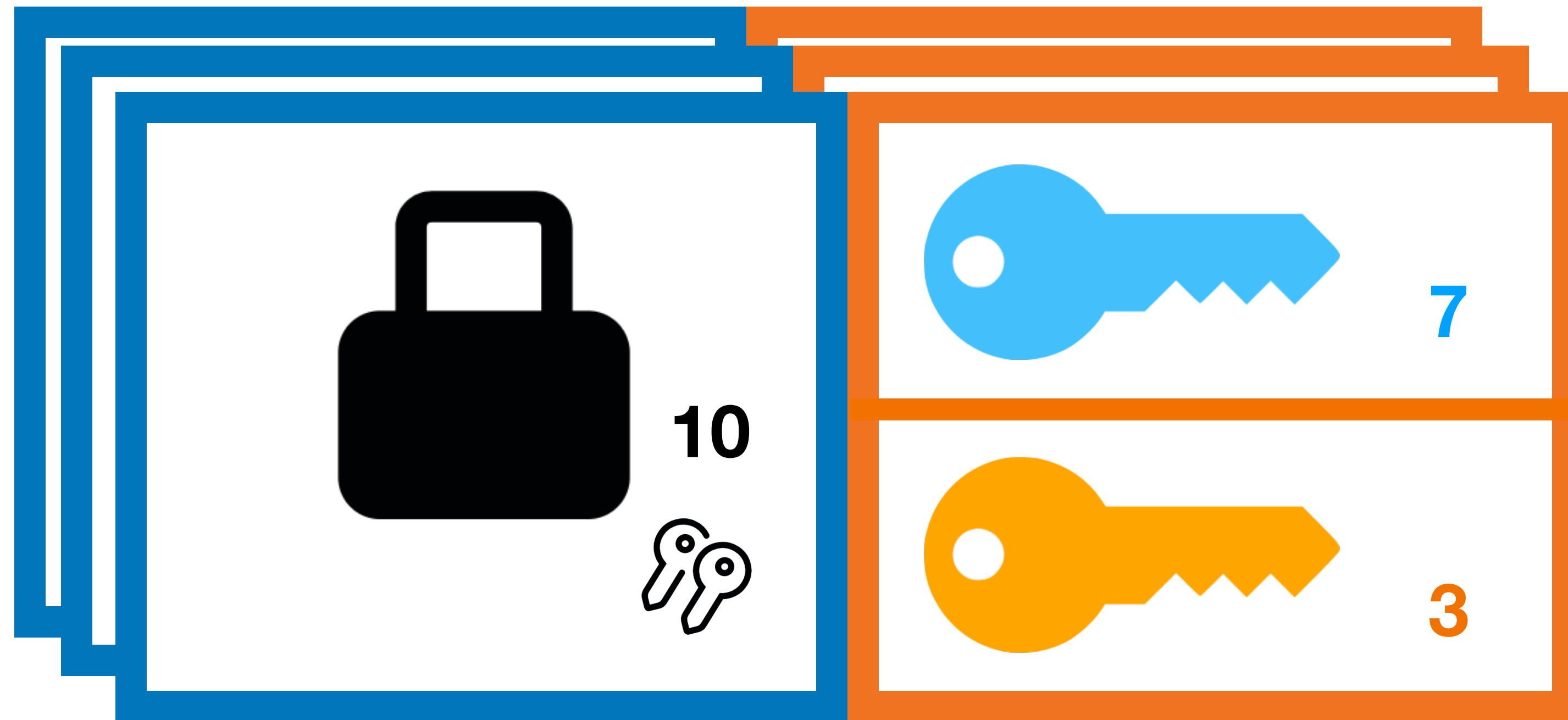
# Fase de operación (commitment transactions)



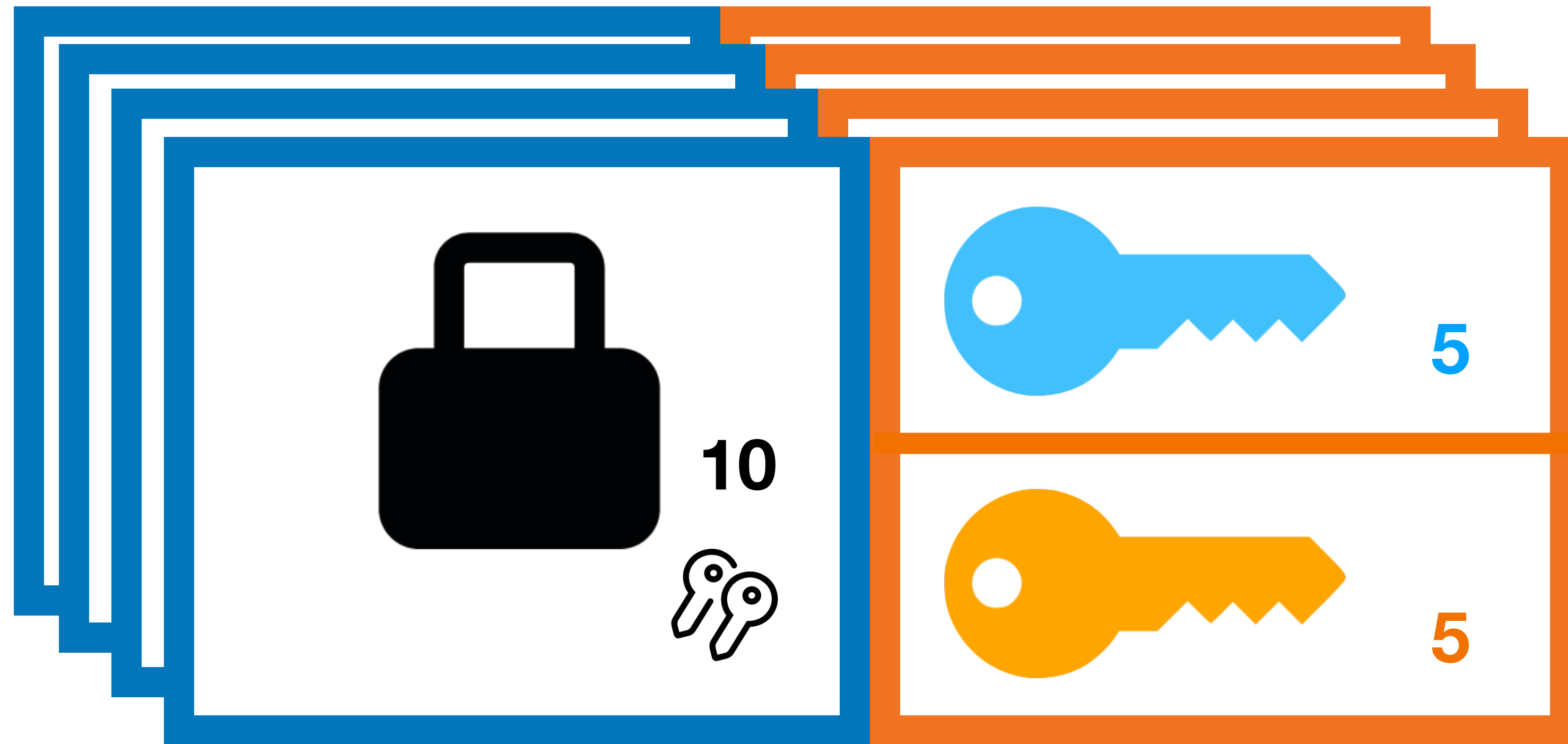
# Fase de operación (commit transactions)



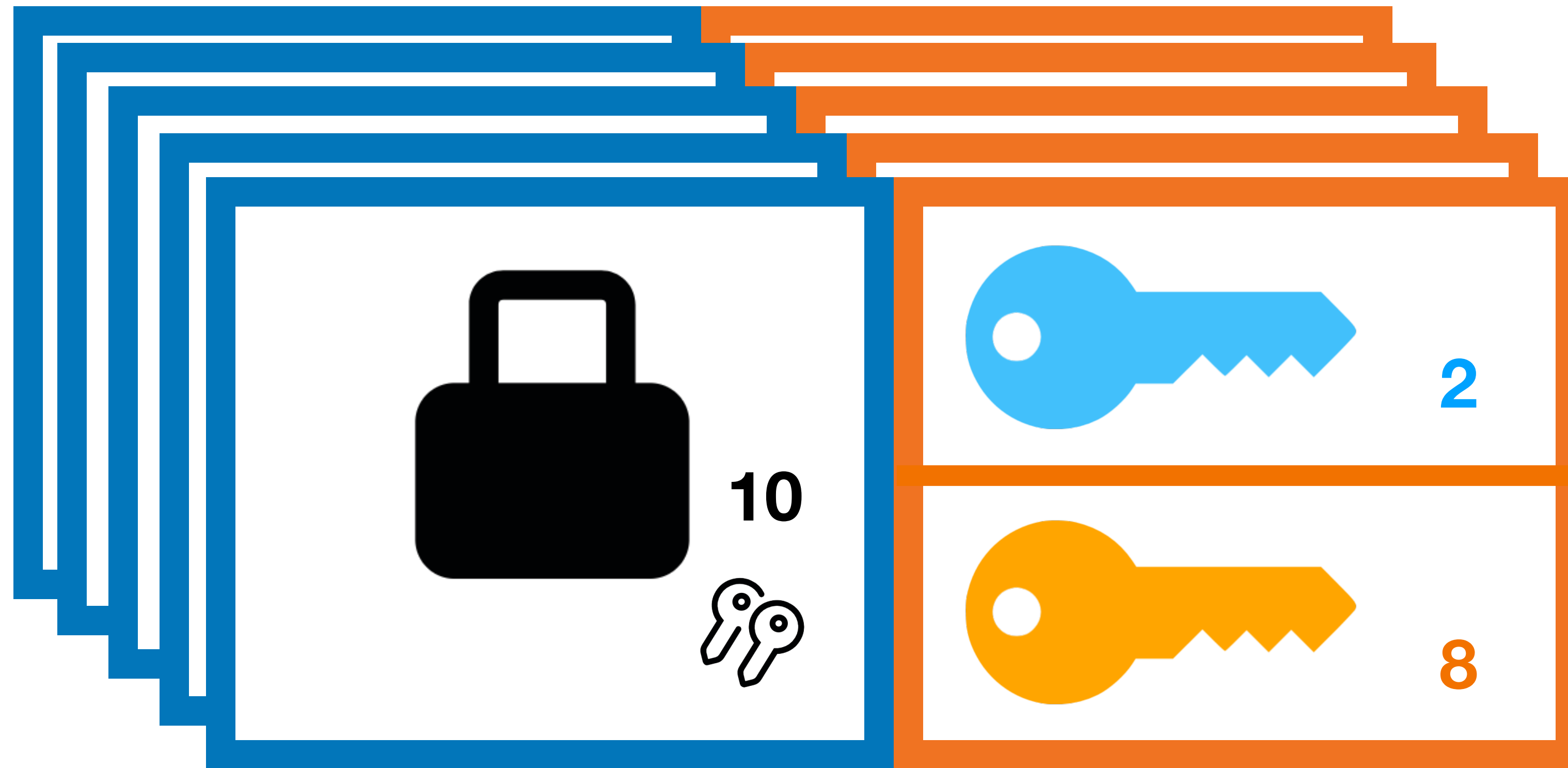
# Fase de operación (commit transactions)



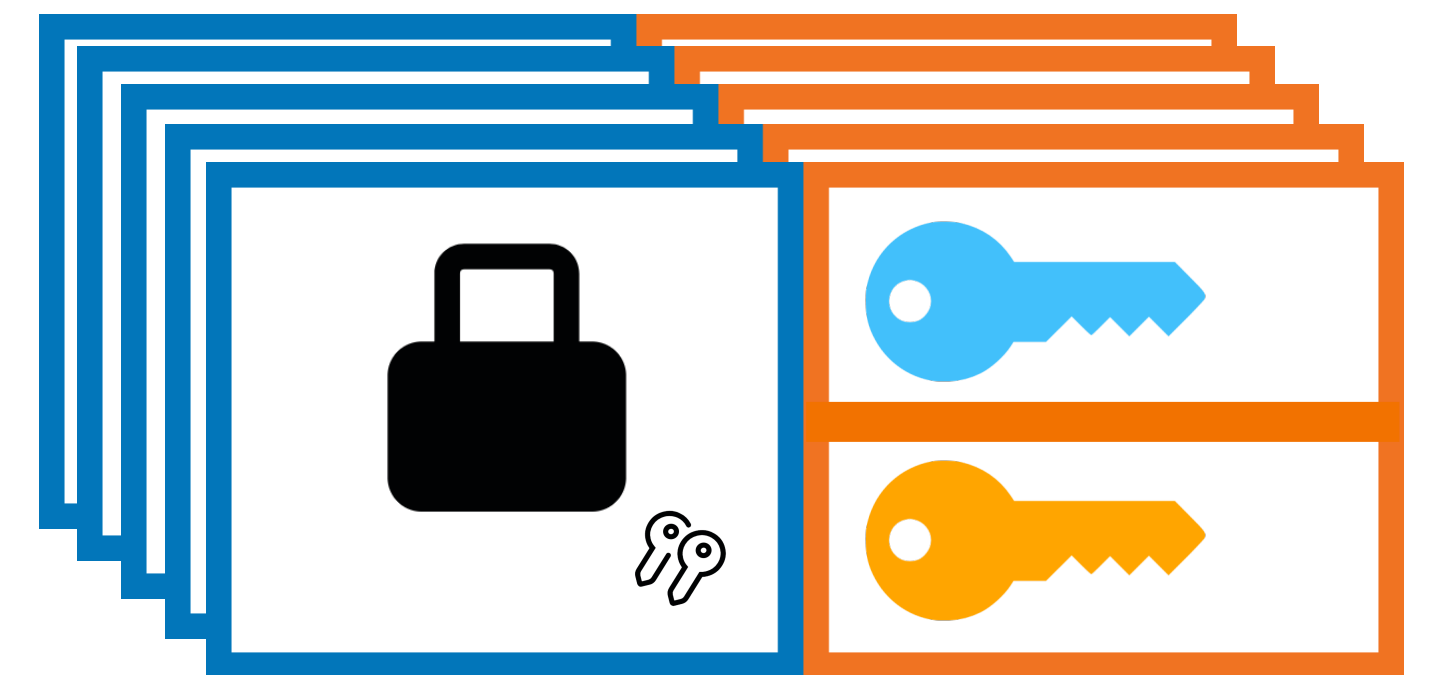
# Fase de operación (commit transactions)



# Fase de operación (commit transactions)

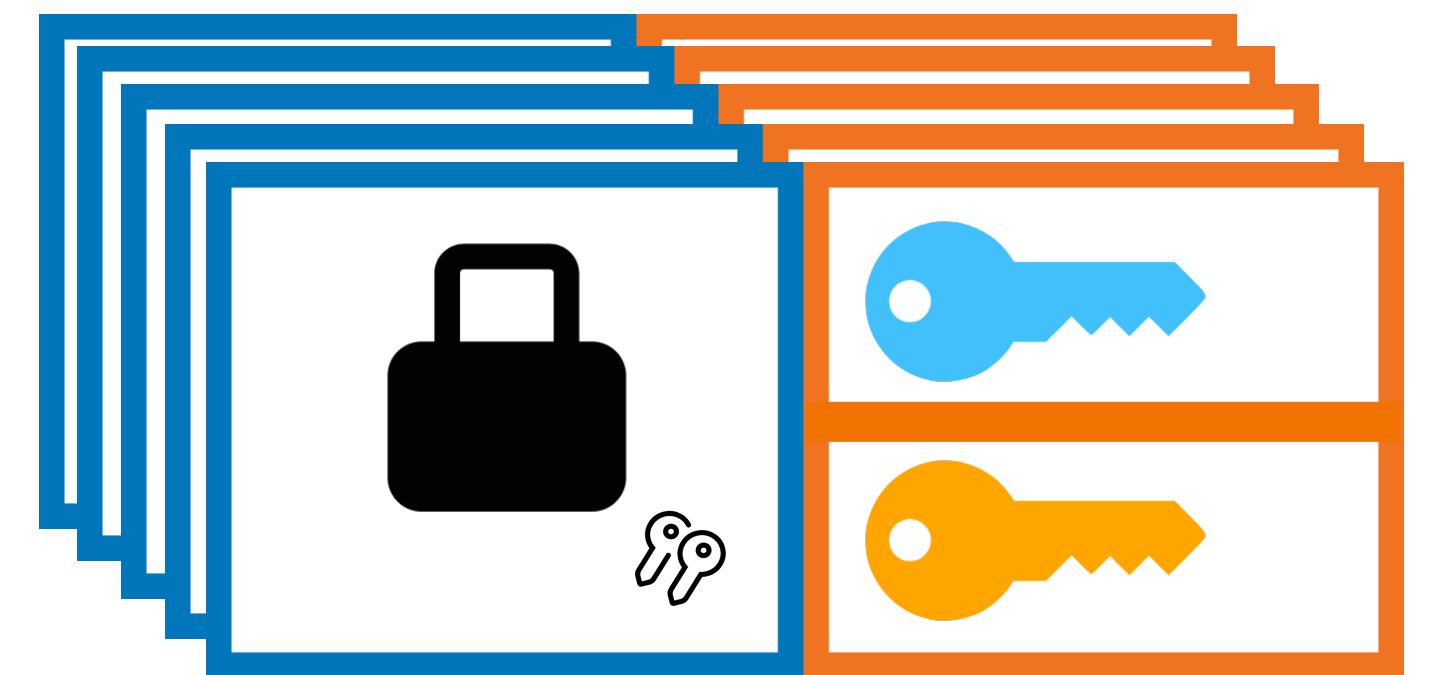


# Fase de operación (commit transactions)



# Fase de operación (commit transactions)

Multiples transacciones 1in-2out

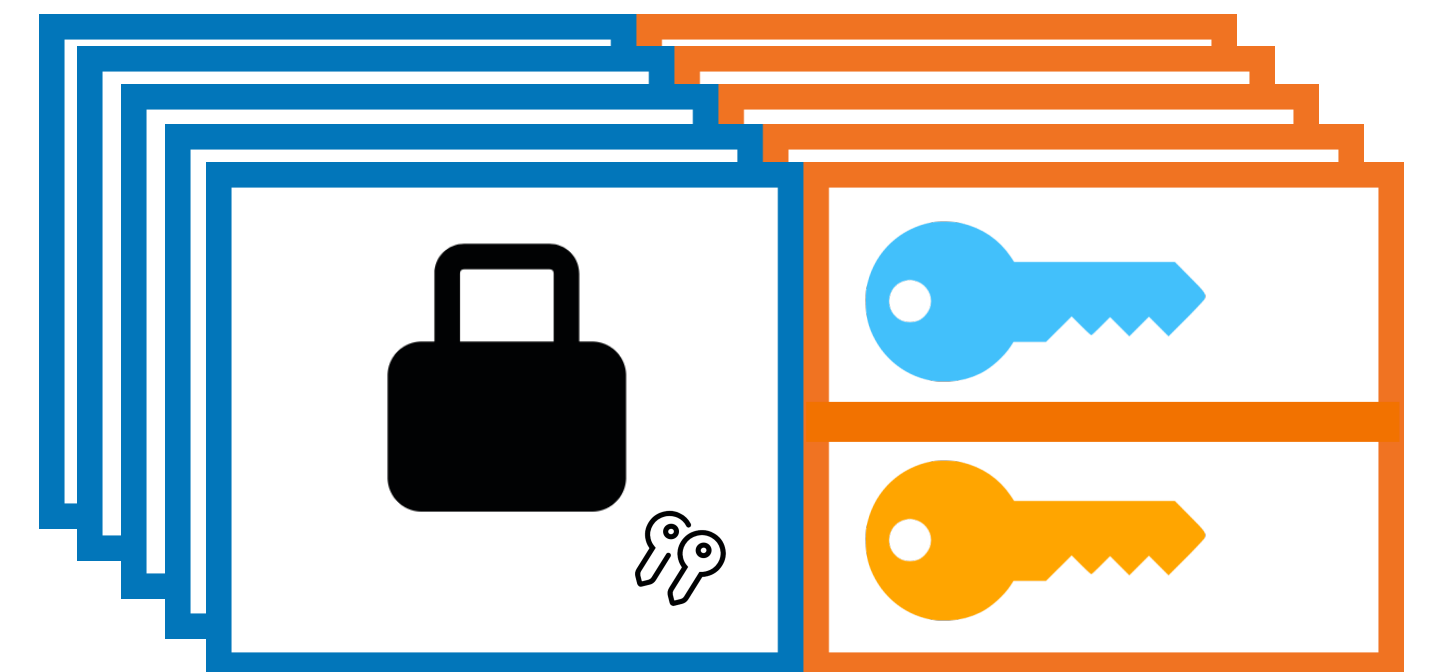




# Fase de operación (commit transactions)

## Multiples transacciones 1in-2out

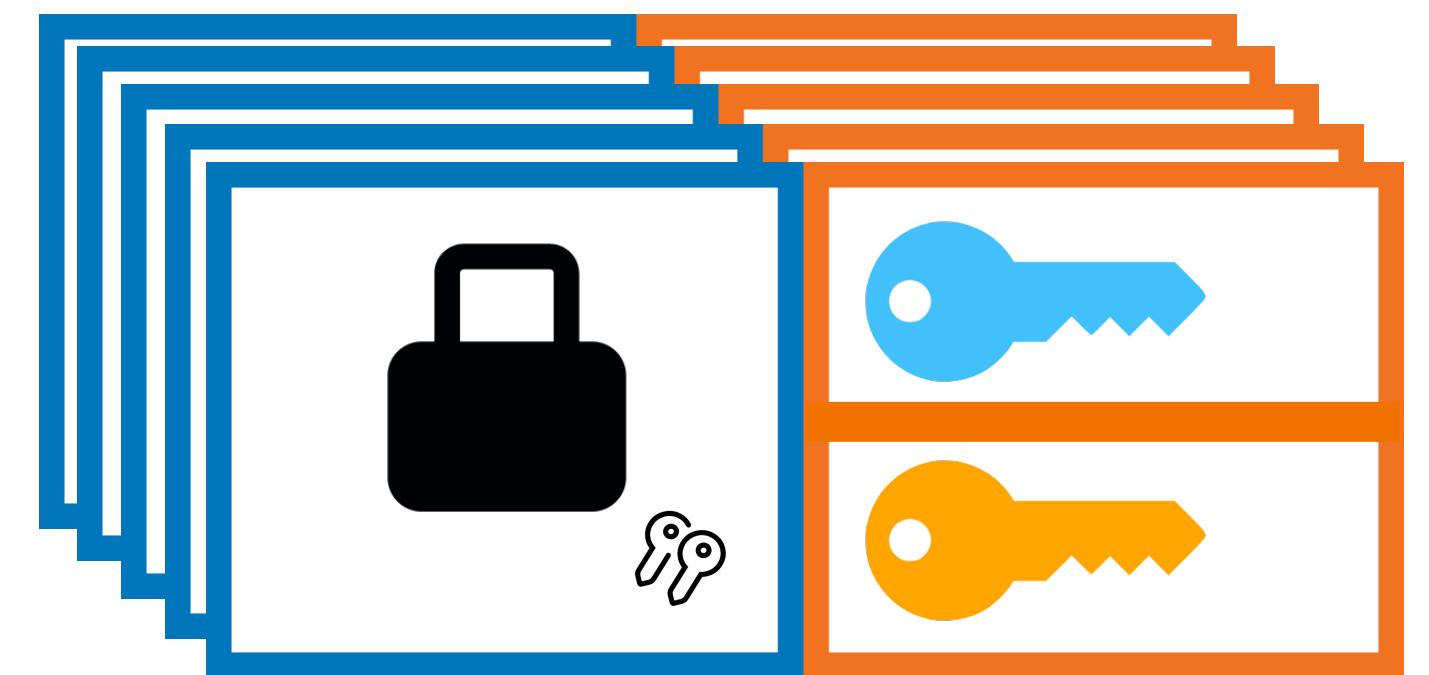
- Todas las transacciones gastan de la transacción de apertura



# Fase de operación (commit transactions)

## Multiples transacciones 1in-2out

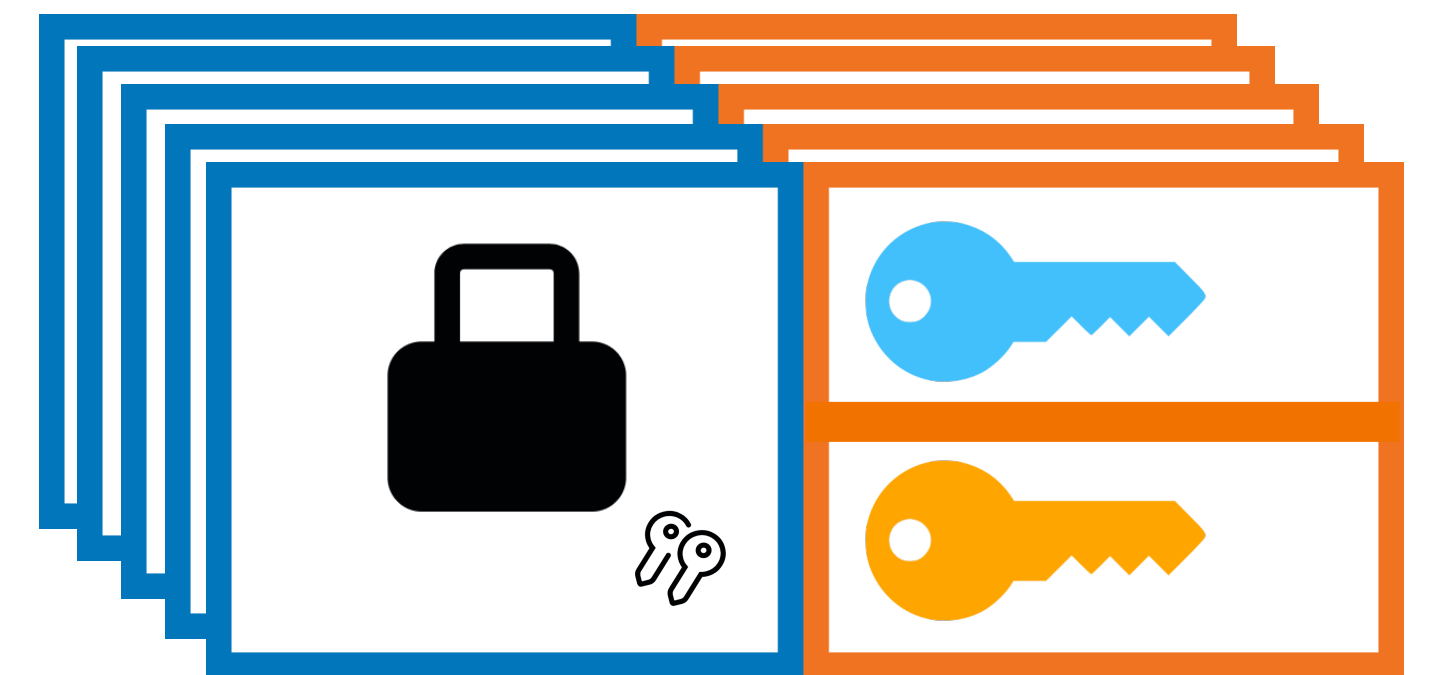
- Todas las transacciones gastan de la transacción de apertura
- Las transacciones son de doble gasto entre ellas



# Fase de operación (commit transactions)

## Multiples transacciones 1in-2out

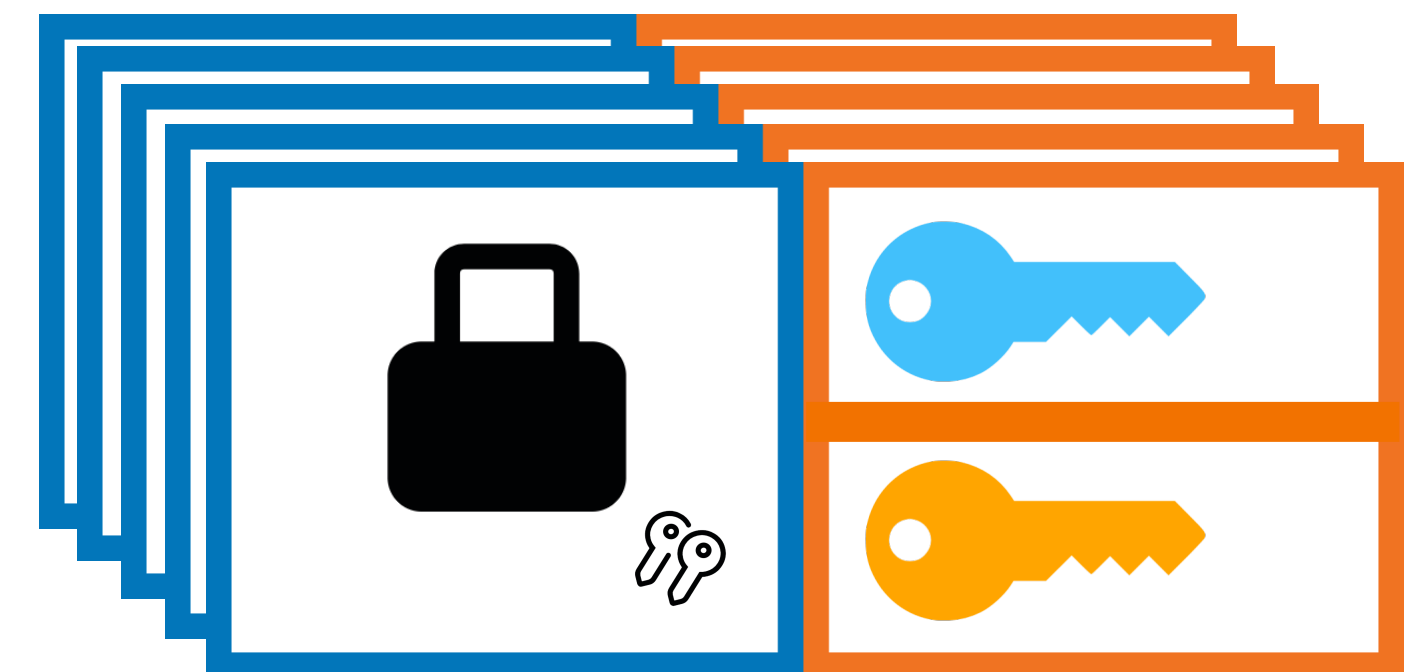
- Todas las transacciones gastan de la transacción de apertura
  - Las transacciones son de doble gasto entre ellas
- Sola la última transacción es valida, el resto son revocadas



# Fase de operación (commit transactions)

## Multiples transacciones 1in-2out

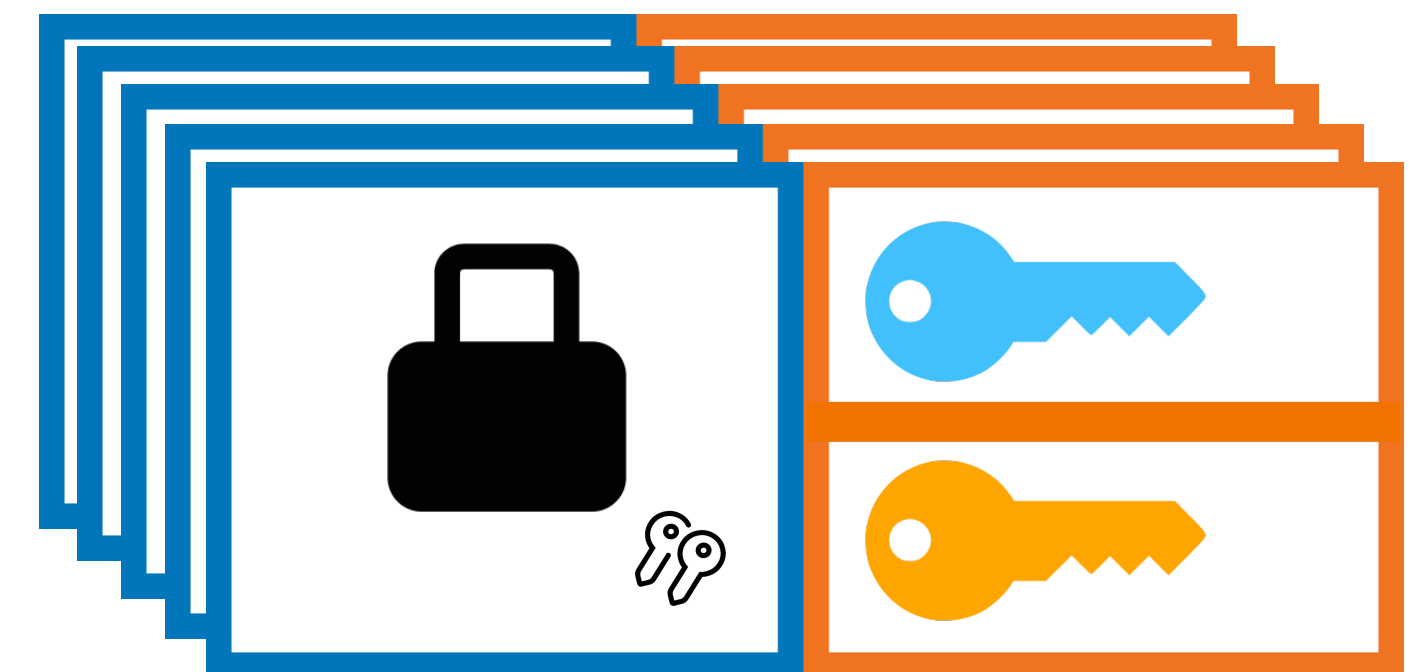
- Todas las transacciones gastan de la transacción de apertura
  - Las transacciones son de doble gasto entre ellas
- Sola la última transacción es valida, el resto son revocadas
  - Cual es el estado actual del canal es información que solo conocen las partes implicadas



# Fase de operación (commit transactions)

## Multiples transacciones 1in-2out

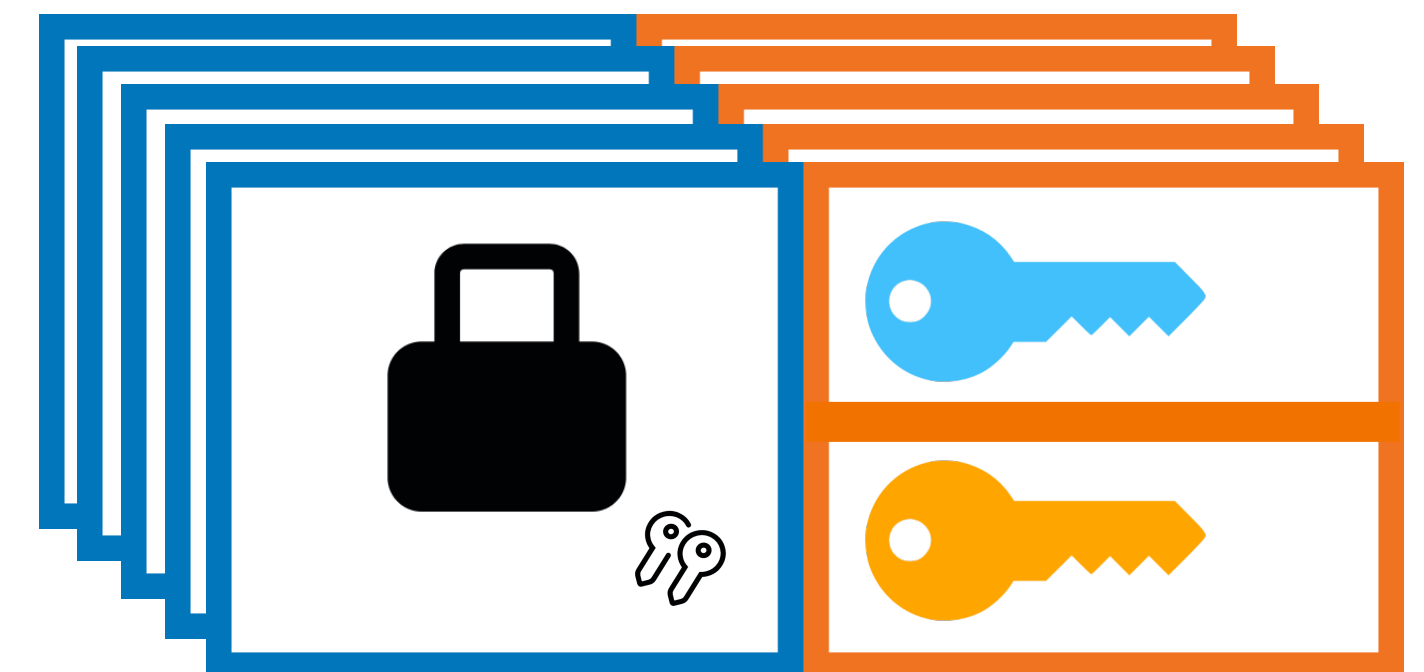
- Todas las transacciones gastan de la transacción de apertura
  - Las transacciones son de doble gasto entre ellas
- Sola la última transacción es valida, el resto son revocadas
  - Cual es el estado actual del canal es información que solo conocen las partes implicadas
- Las transacciones solo se intercambian entre las partes del canal



# Fase de operación (commitment transactions)

## Multiples transacciones 1in-2out

- Todas las transacciones gastan de la transacción de apertura
  - Las transacciones son de doble gasto entre ellas
- Sola la última transacción es valida, el resto son revocadas
  - Cual es el estado actual del canal es información que solo conocen las partes implicadas
- Las transacciones solo se intercambian entre las partes del canal
- El canal puede ser cerrado de forma unilateral utilizando cualquier commitment transaction



# **Fase de operación (commitment transactions)**

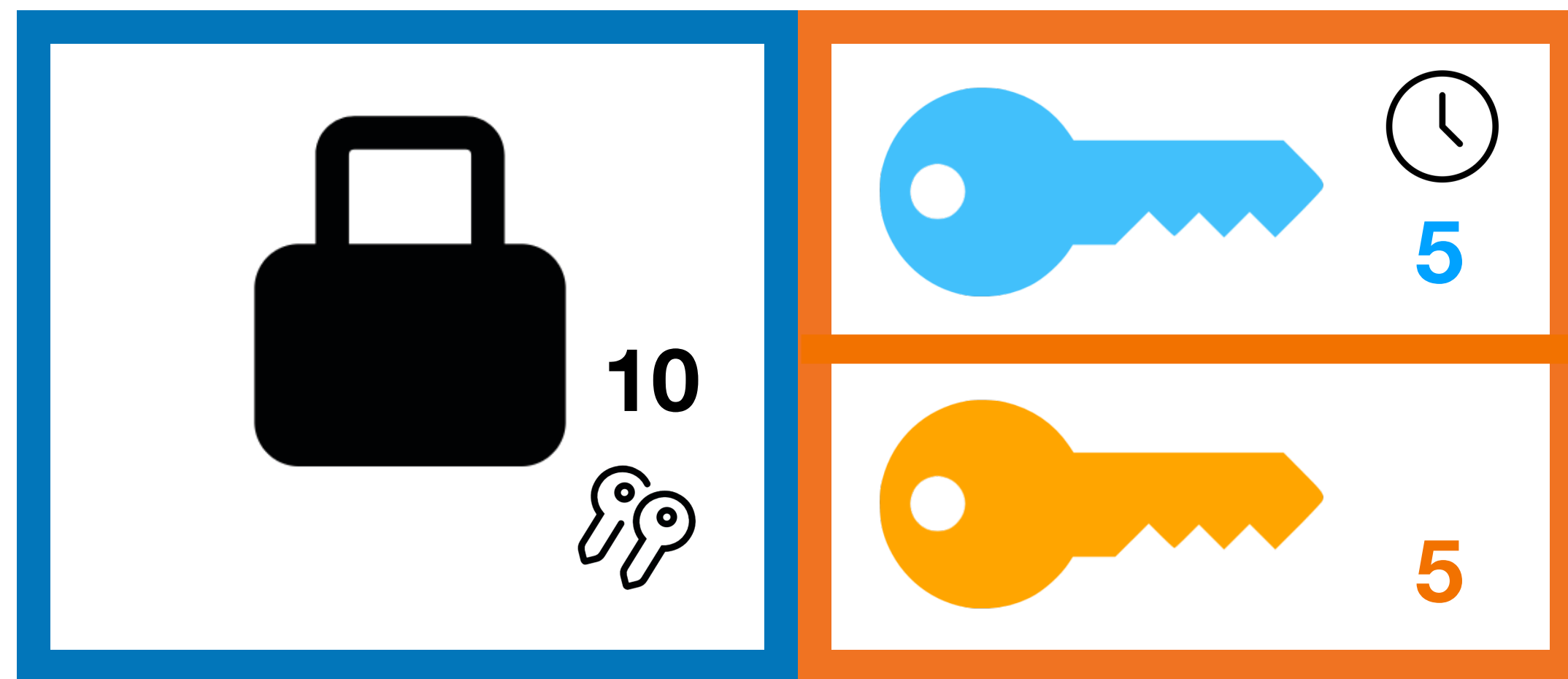
## **Asimetría de las transacciones**

Las transacciones de la fase de operación son asimétricas entre las dos partes, de tal forma que si una de las partes intenta cerrar el canal sin cooperar con su contraparte, esta deberá esperar un cierto tiempo de disputa hasta poder reclamar su parte de los fondos.

# Fase de operación (commit transactions)

## Asimetría de las transacciones

Las transacciones de la fase de operación son asimétricas entre las dos partes, de tal forma que si una de las partes intenta cerrar el canal sin cooperar con su contraparte, esta deberá esperar un cierto tiempo de disputa hasta poder reclamar su parte de los fondos.



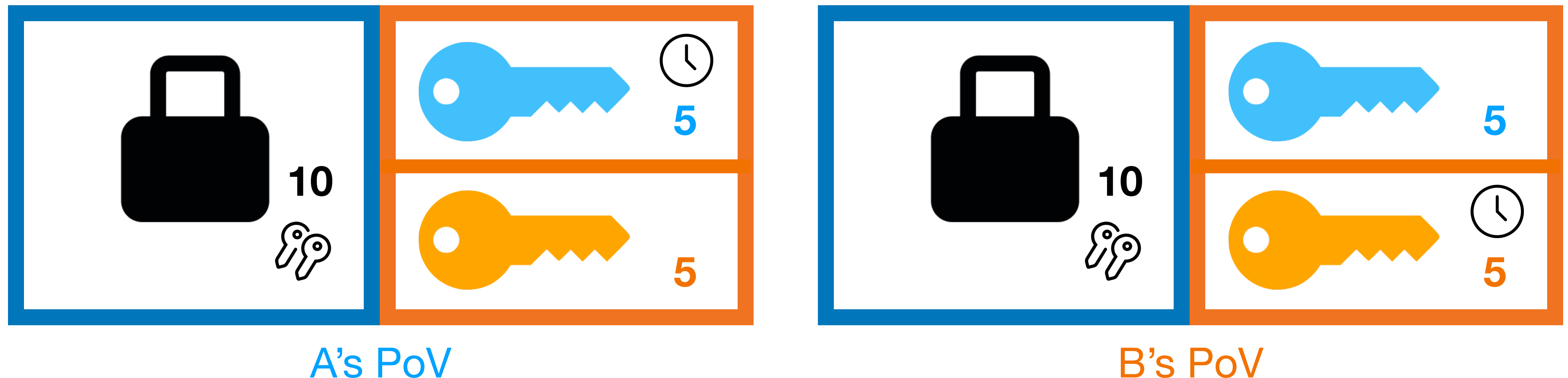
A's PoV



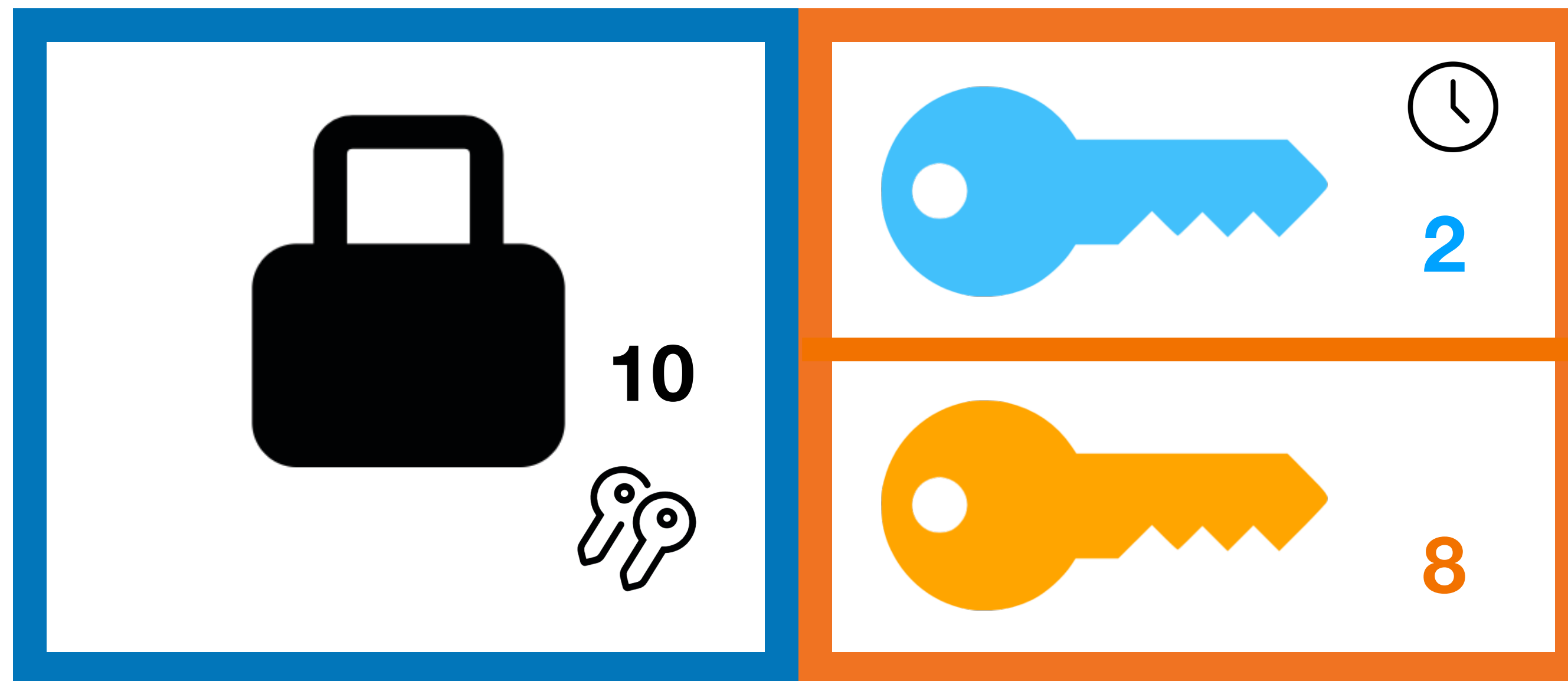
# Fase de operación (commitment transactions)

## Asimetría de las transacciones

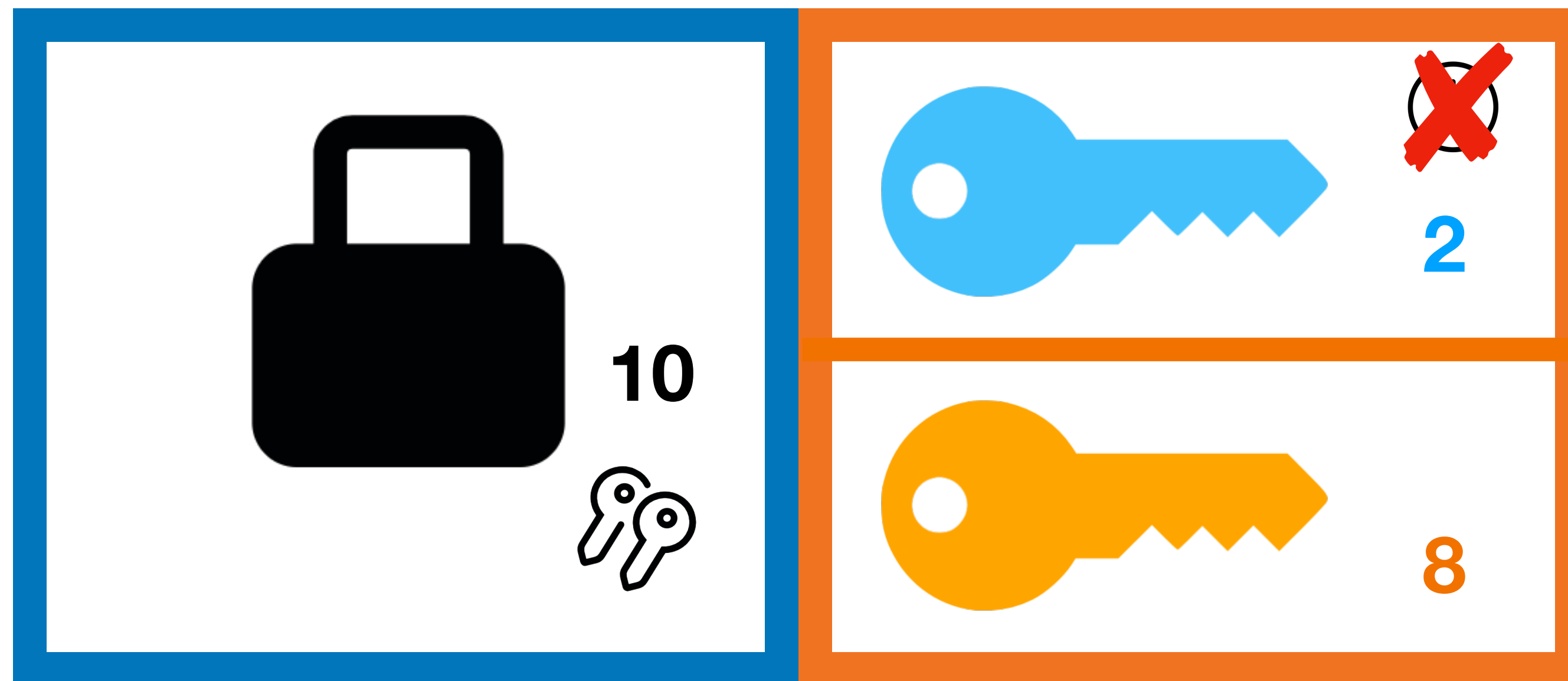
Las transacciones de la fase de operación son asimétricas entre las dos partes, de tal forma que si una de las partes intenta cerrar el canal sin cooperar con su contraparte, esta deberá esperar un cierto tiempo de disputa hasta poder reclamar su parte de los fondos.



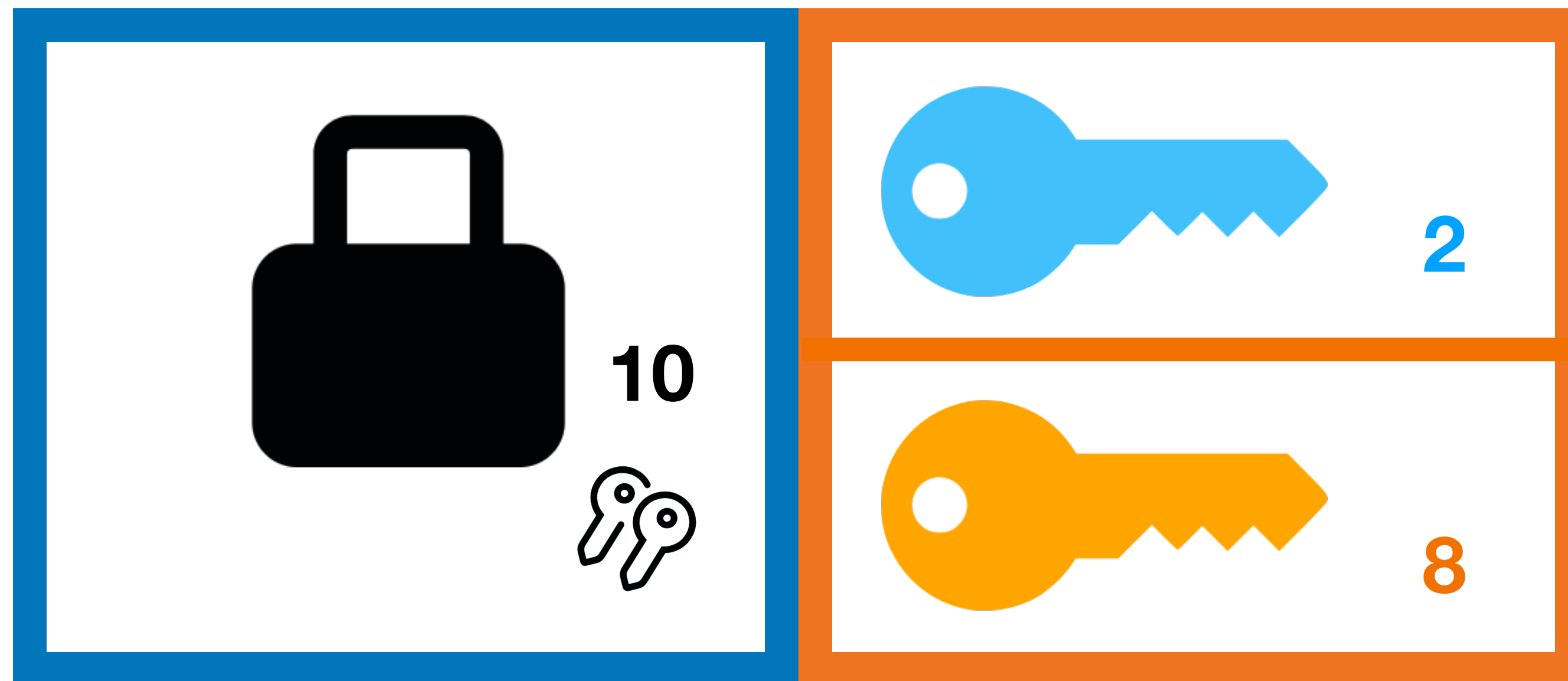
# Cierre del canal (closing transaction)



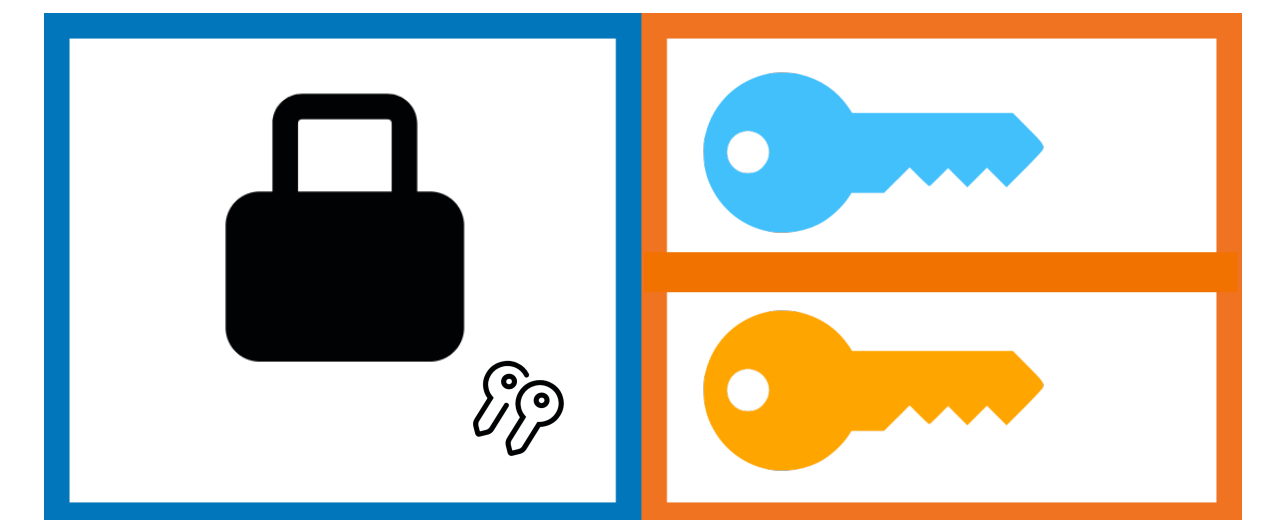
# Cierre del canal (closing transaction)



# Cierre del canal (closing transaction)

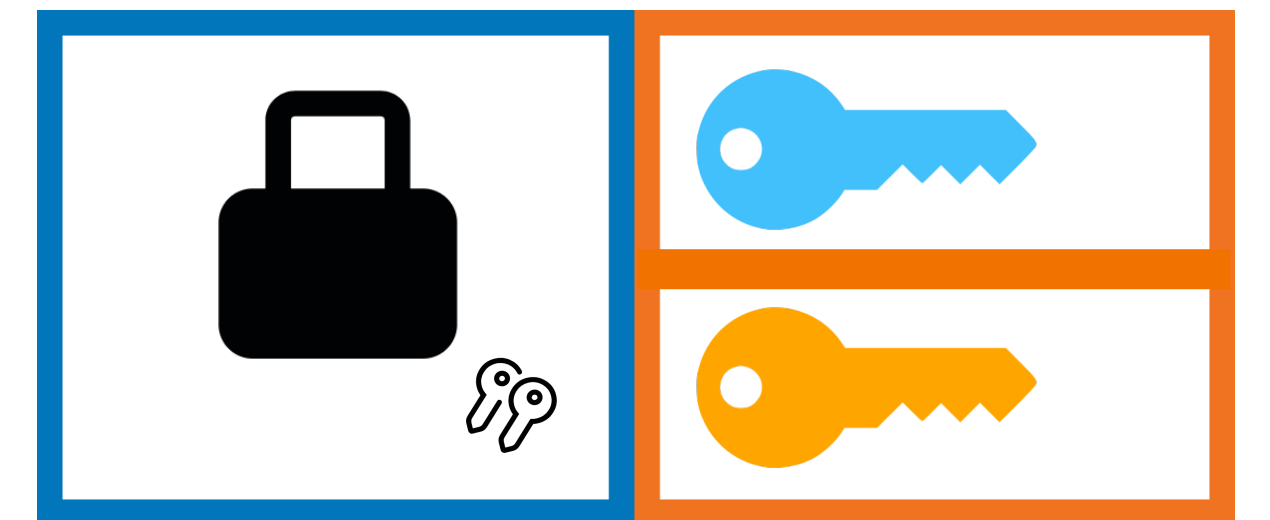


# Cierre del canal (closing transaction)



# Cierre del canal (closing transaction)

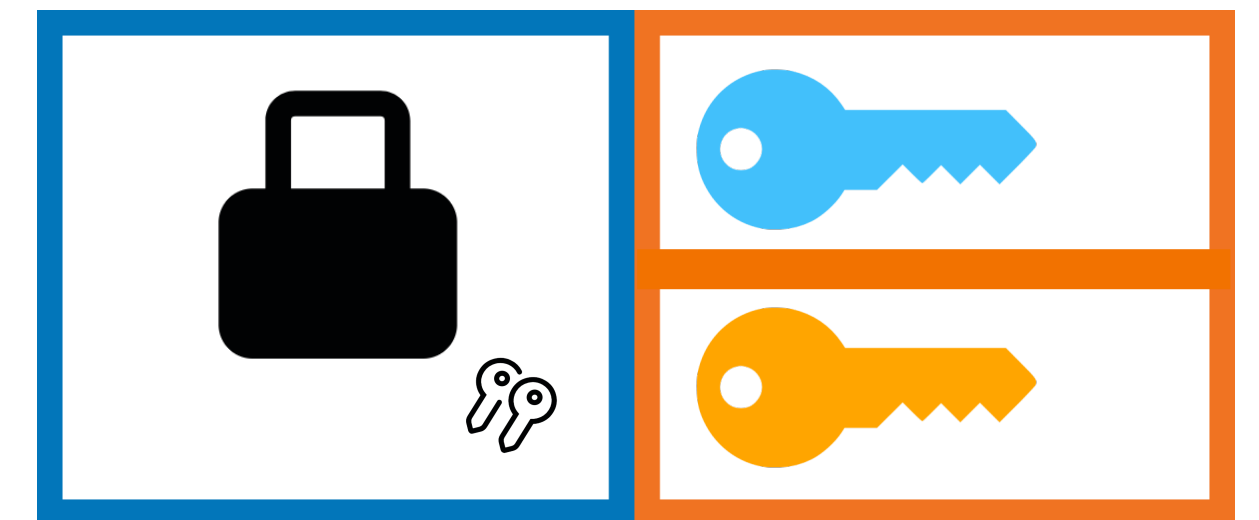
Una transacción 1in-2out



# Cierre del canal (closing transaction)

## Una transacción 1in-2out

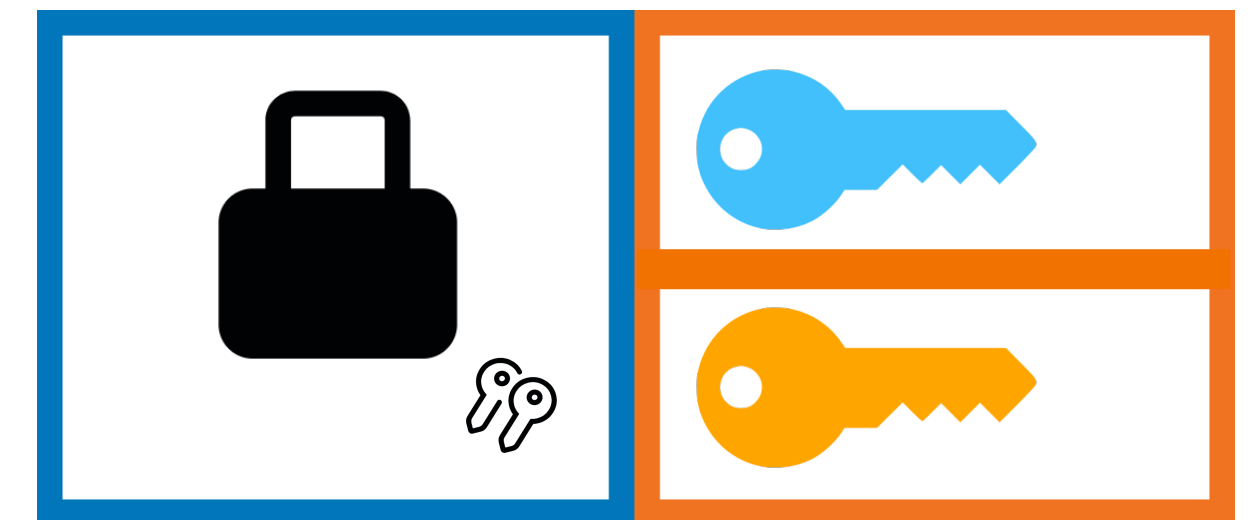
- Si ambas partes están de acuerdo, se genera un cierre cooperativo



# Cierre del canal (closing transaction)

## Una transacción 1in-2out

- Si ambas partes están de acuerdo, se genera un cierre cooperativo
- Las restricciones condicionales son eliminadas y ambas partes pueden reclamar sus fondos directamente

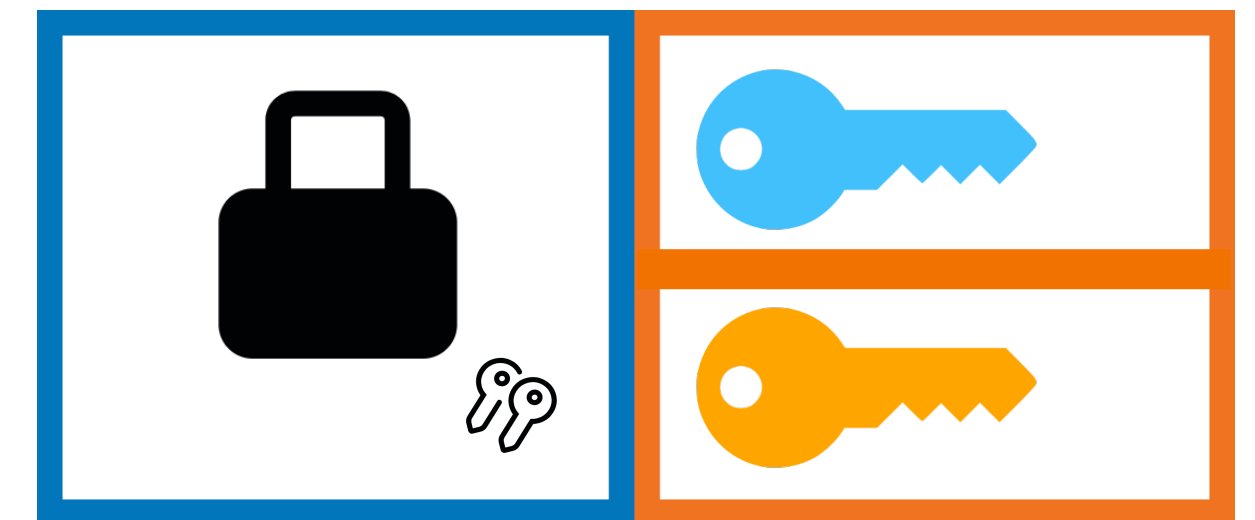




# Cierre del canal (closing transaction)

## Una transacción 1in-2out

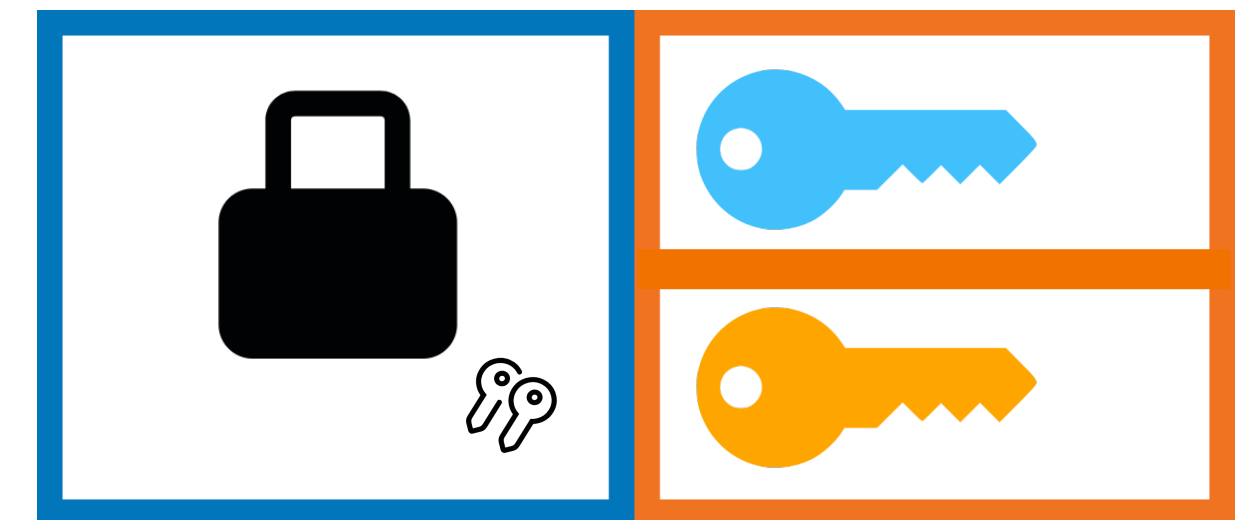
- Si ambas partes están de acuerdo, se genera un cierre cooperativo
- Las restricciones condicionales son eliminadas y ambas partes pueden reclamar sus fondos directamente
- Si las partes no se ponen de acuerdo, las restricciones se mantienen y el canal se cierra con un proceso de disputa onchain



# Cierre del canal (closing transaction)

## Una transacción 1in-2out

- Si ambas partes están de acuerdo, se genera un cierre cooperativo
- Las restricciones condicionales son eliminadas y ambas partes pueden reclamar sus fondos directamente
- Si las partes no se ponen de acuerdo, las restricciones se mantienen y el canal se cierra con un proceso de disputa onchain
- Toda la interacción entre las partes queda simplificada en dos únicas transacciones, la de apertura y la de cierre



# **Cierre del canal (closing transaction)**

**¿Pero qué sucede si el canal se cierra con proceso de disputa y una de las partes no esta presente (online)?**

- Si el proceso de disputa termina sin recurso, el cierre se hace efectivo
- Esto puede resultar en robo de fondos (intentar confirmar un estado antiguo del canal)
- Esto conlleva que las partes deben estar siempre presentes... ¿O no?

# Cierre del canal (closing transaction)

**¿Pero qué sucede si el canal se cierra con proceso de disputa y una de las partes no esta presente (online)?**

- Si el proceso de disputa termina sin recurso, el cierre se hace efectivo
- Esto puede resultar en robo de fondos (intentar confirmar un estado antiguo del canal)
- Esto conlleva que las partes deben estar siempre presentes... ¿O no?



# Watchtowers



# Watchtowers



**¿Cuál es el paradigma en el que se basan los servicios de monitorización de cadena (¿Qué son las Watchtowers?)?**

# Watchtowers



**¿Cuál es el paradigma en el que se basan los servicios de monitorización de cadena (¿Qué son las Watchtowers?)?**

# Watchtowers

**¿Cuál es el paradigma en el que se basan los servicios de monitorización de cadena (¿Qué son las Watchtowers?)?**

Usuario:



# Watchtowers

¿Cuál es el paradigma en el que se basan los servicios de monitorización de cadena (¿Qué son las Watchtowers?)?

Usuario:

- Envía **datos** a un servidor conjuntamente con una condición de acción (**trigger**)

# Watchtowers

¿Cuál es el paradigma en el que se basan los servicios de monitorización de cadena (¿Qué son las Watchtowers?)?

Usuario:

- Envía **datos** a un servidor conjuntamente con una condición de acción (**trigger**)

# Watchtowers

¿Cuál es el paradigma en el que se basan los servicios de monitorización de cadena (¿Qué son las Watchtowers?)?

Usuario:

- Envía **datos** a un servidor conjuntamente con una condición de acción (**trigger**)

Servidor:

# Watchtowers

¿Cuál es el paradigma en el que se basan los servicios de monitorización de cadena (¿Qué son las Watchtowers?)?

Usuario:

- Envía **datos** a un servidor conjuntamente con una condición de acción (**trigger**)

Servidor:

- Monitoriza un cierto canal de comunicación en busca de **triggers**

# Watchtowers

¿Cuál es el paradigma en el que se basan los servicios de monitorización de cadena (¿Qué son las Watchtowers?)?

Usuario:

- Envía **datos** a un servidor conjuntamente con una condición de acción (**trigger**)

Servidor:

- Monitoriza un cierto canal de comunicación en busca de **triggers**
- Si un trigger es identificado, realiza **una acción** con los datos recibidos anteriormente

# **Protocolo básico de watchtowers**

# Protocolo básico de watchtowers



# Protocolo básico de watchtowers





# Protocolo básico de watchtowers



[...]  
commitment\_txid,  
penalty\_tx,  
[...]



# Protocolo básico de watchtowers

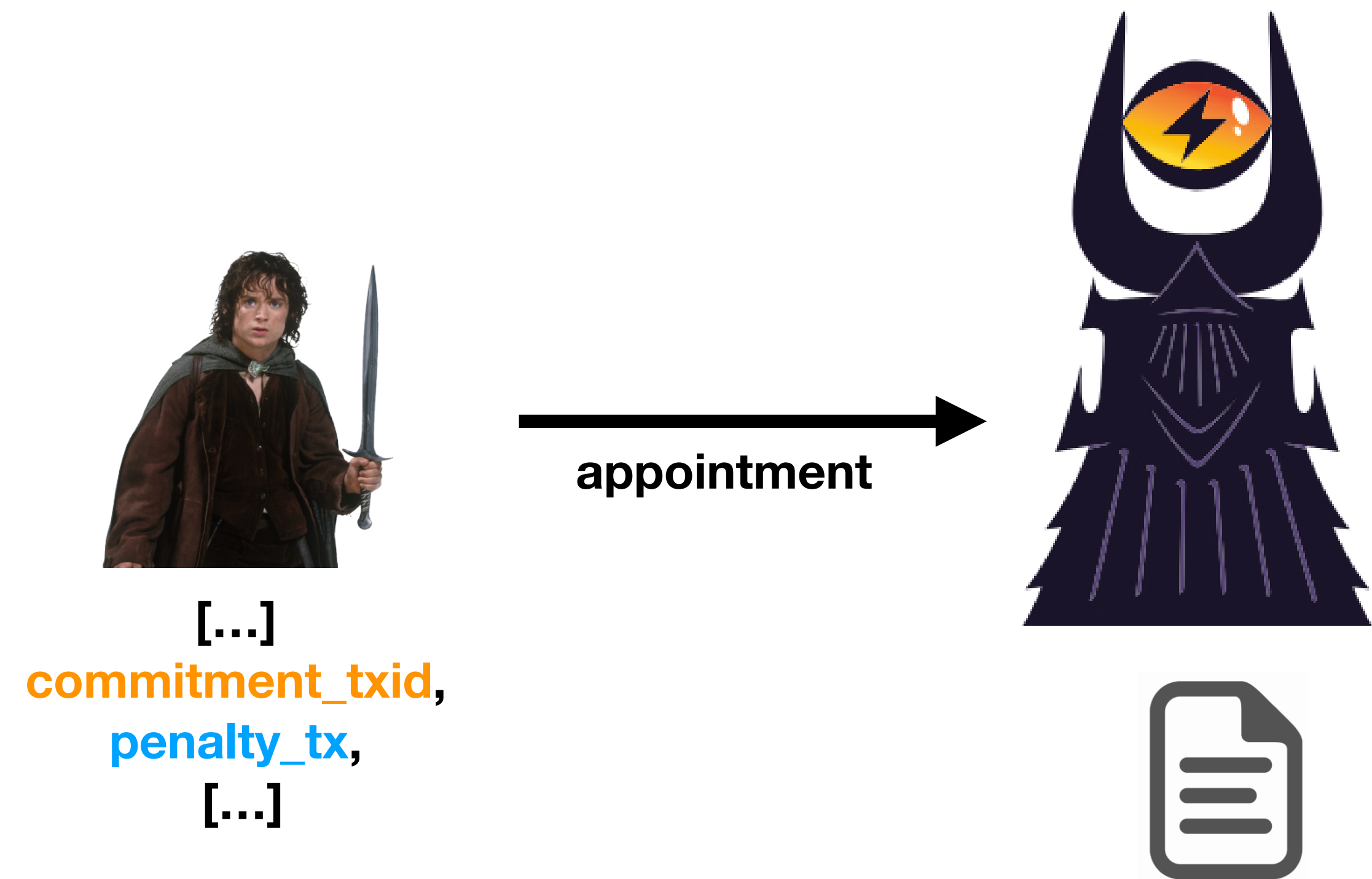


[...]

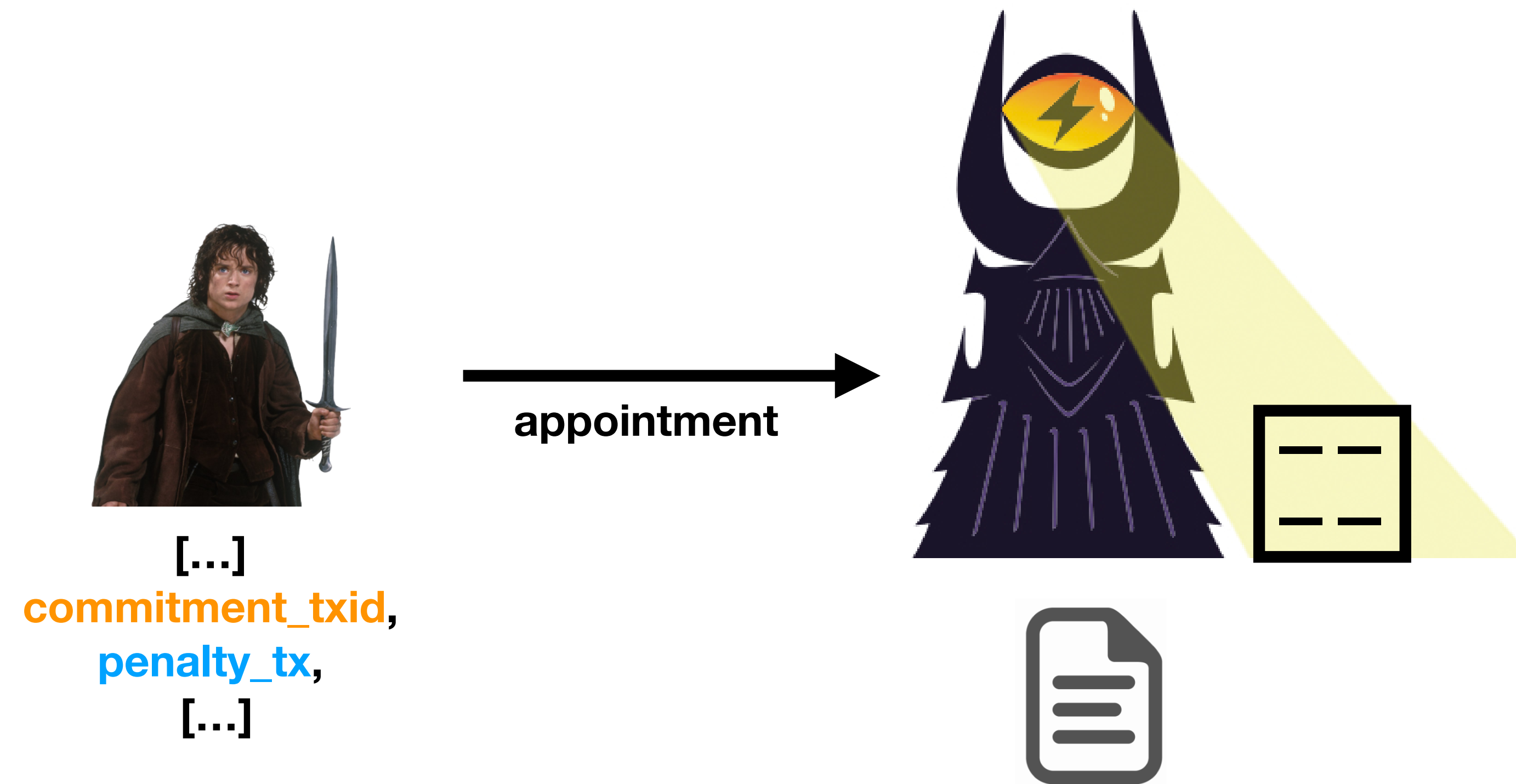
commitment\_txid,  
penalty\_tx,  
[...]



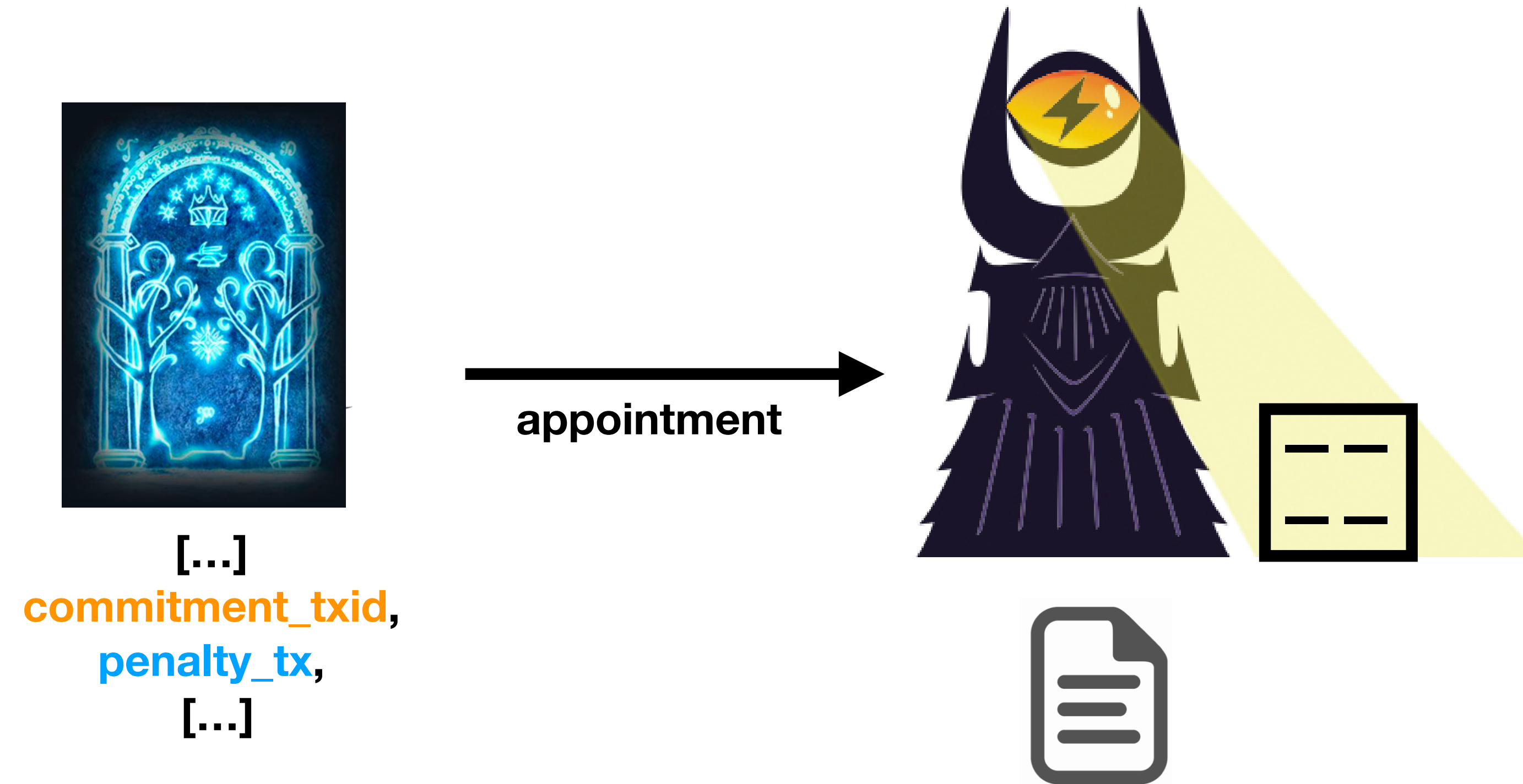
# Protocolo básico de watchtowers



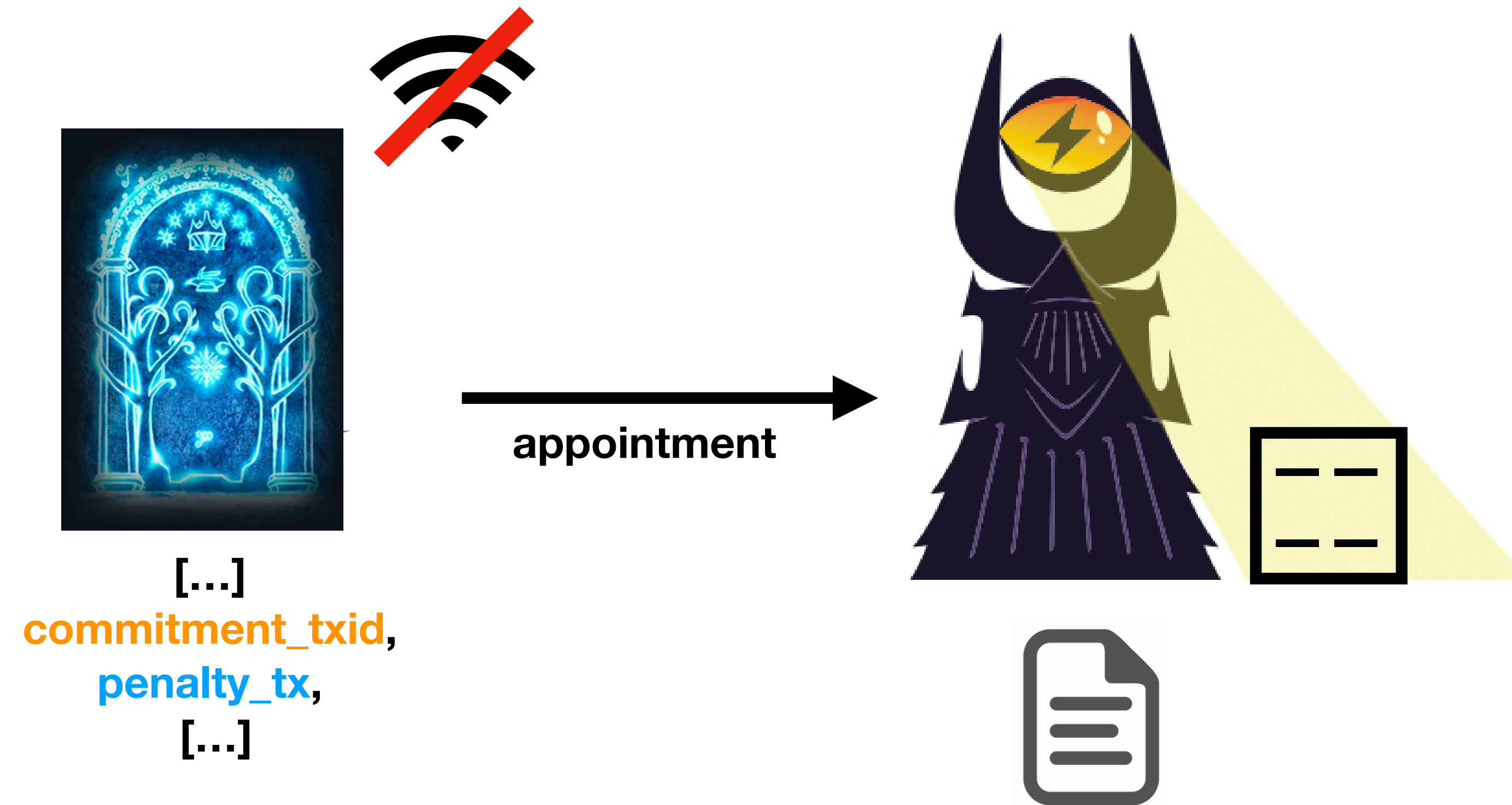
# Protocolo básico de watchtowers



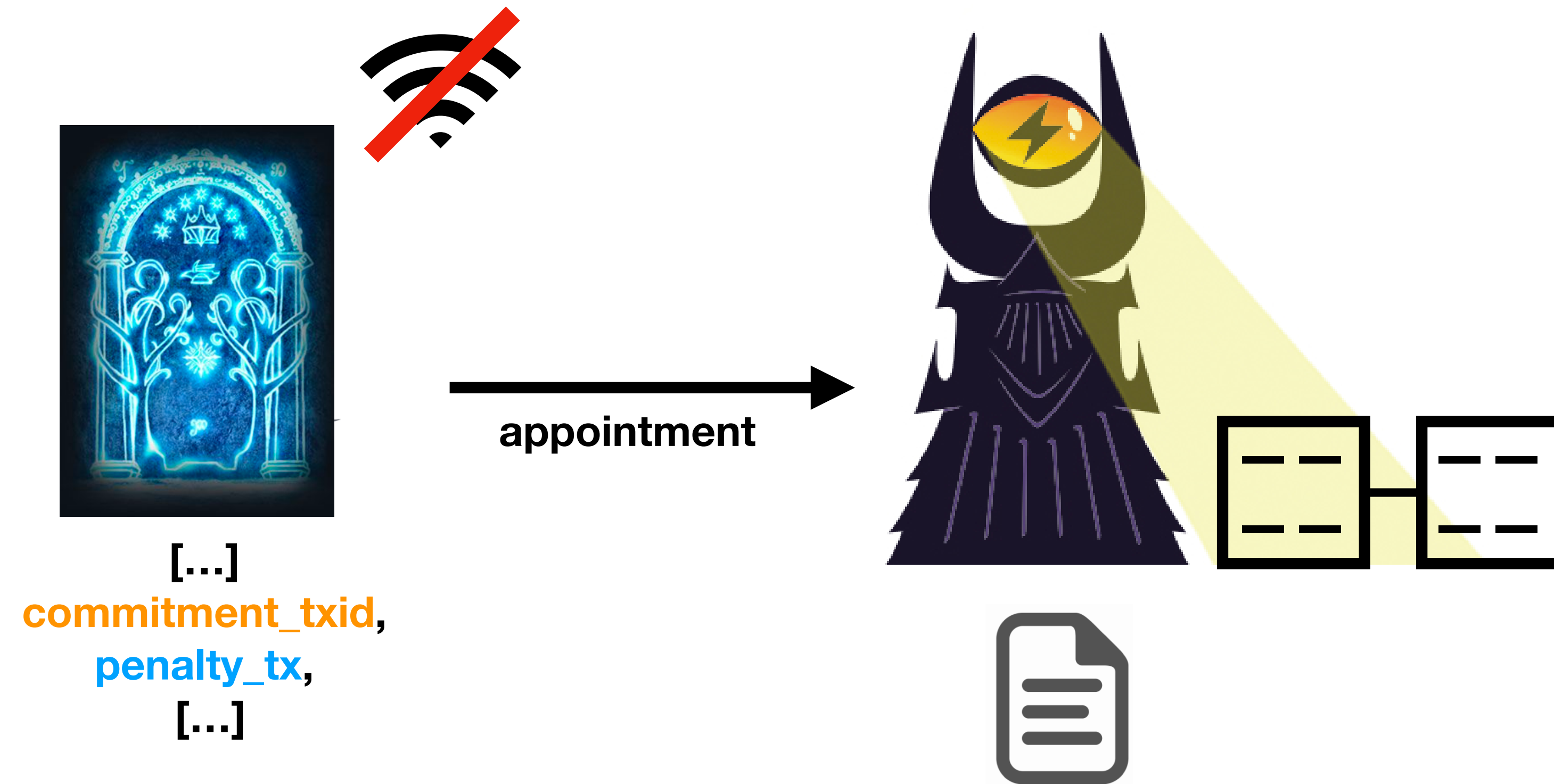
# Protocolo básico de watchtowers



# Protocolo básico de watchtowers

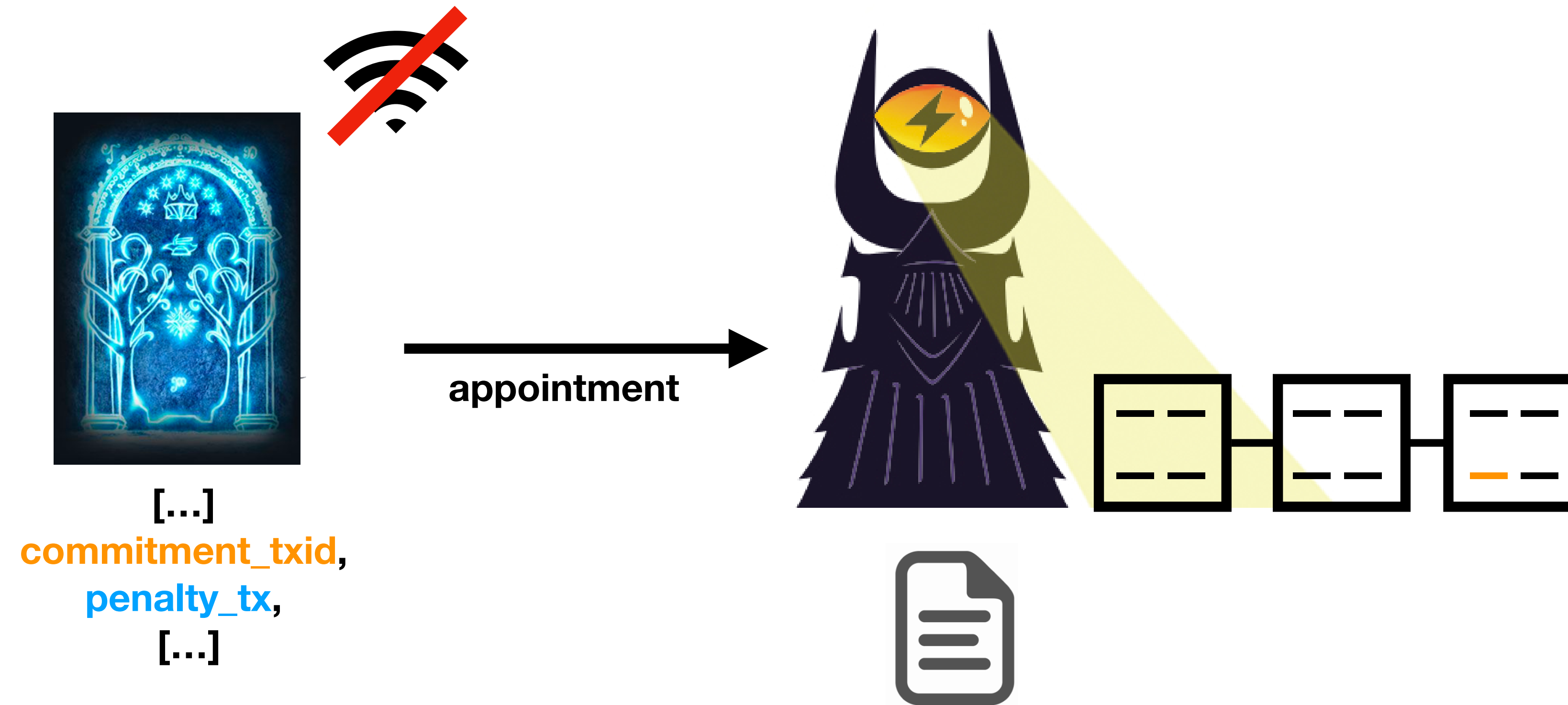


# Protocolo básico de watchtowers



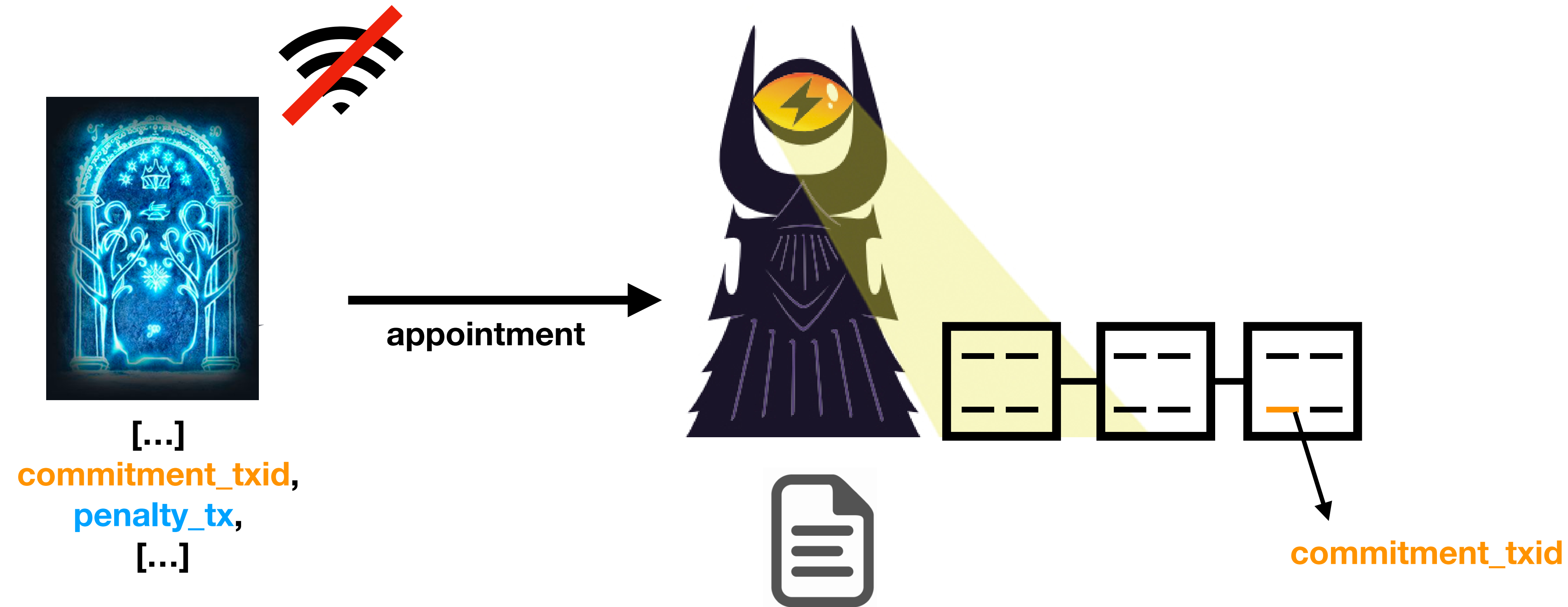


# Protocolo básico de watchtowers

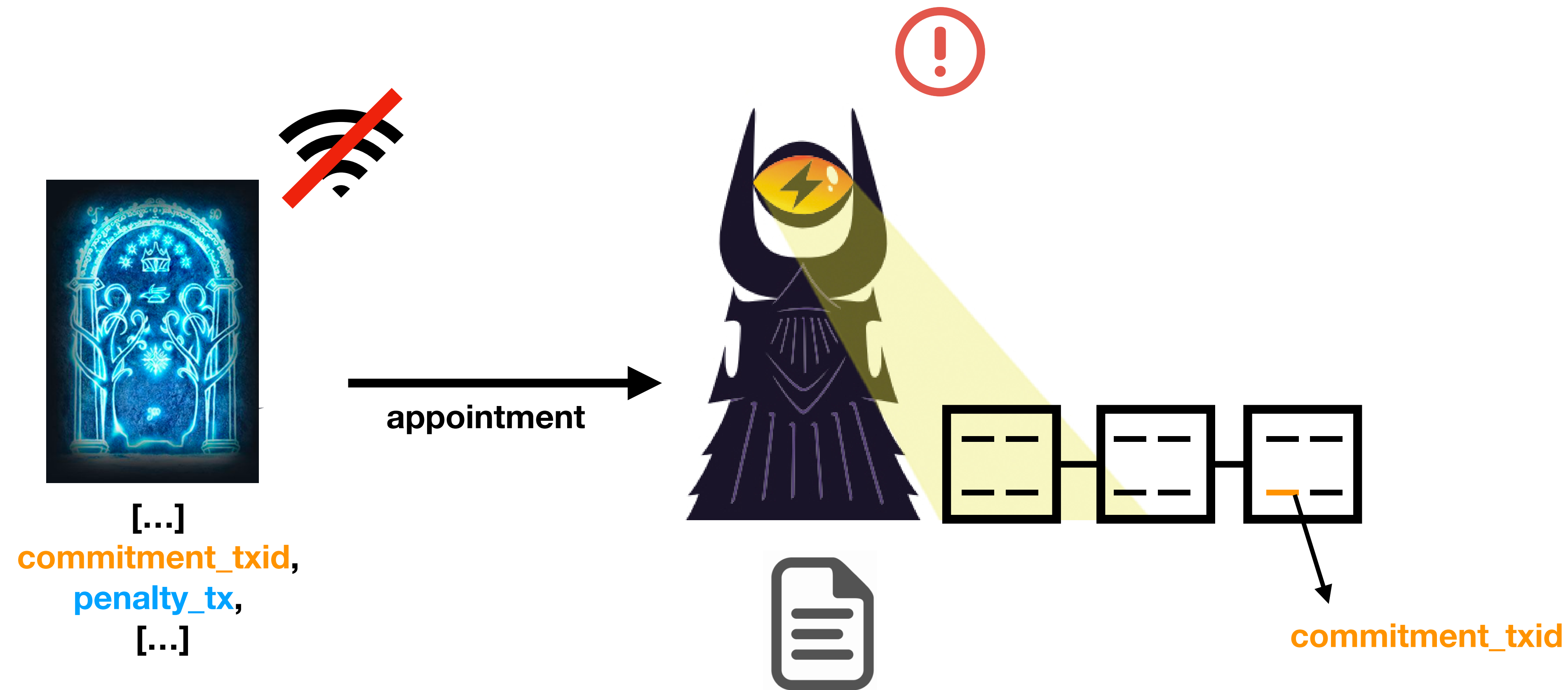




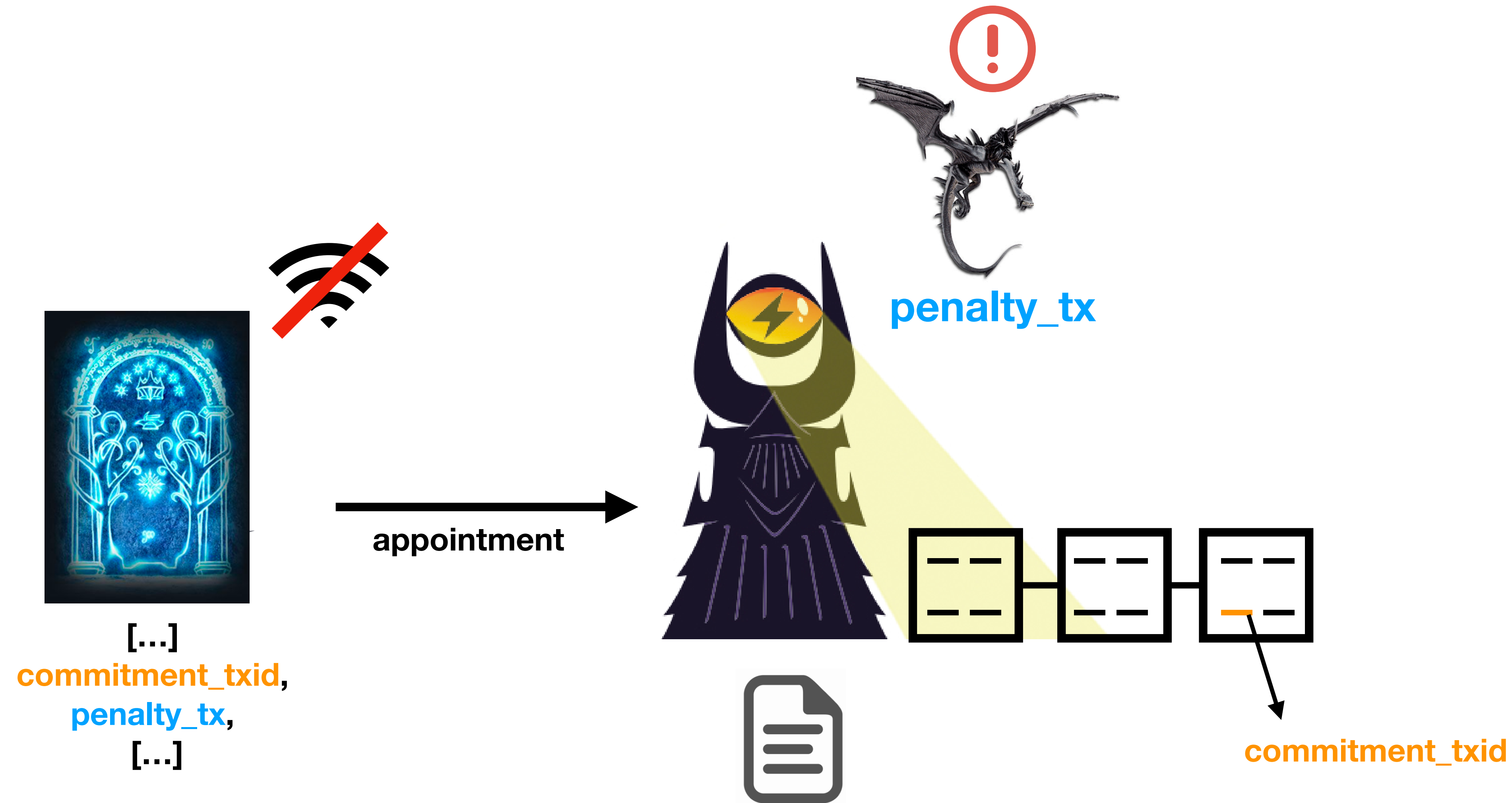
# Protocolo básico de watchtowers



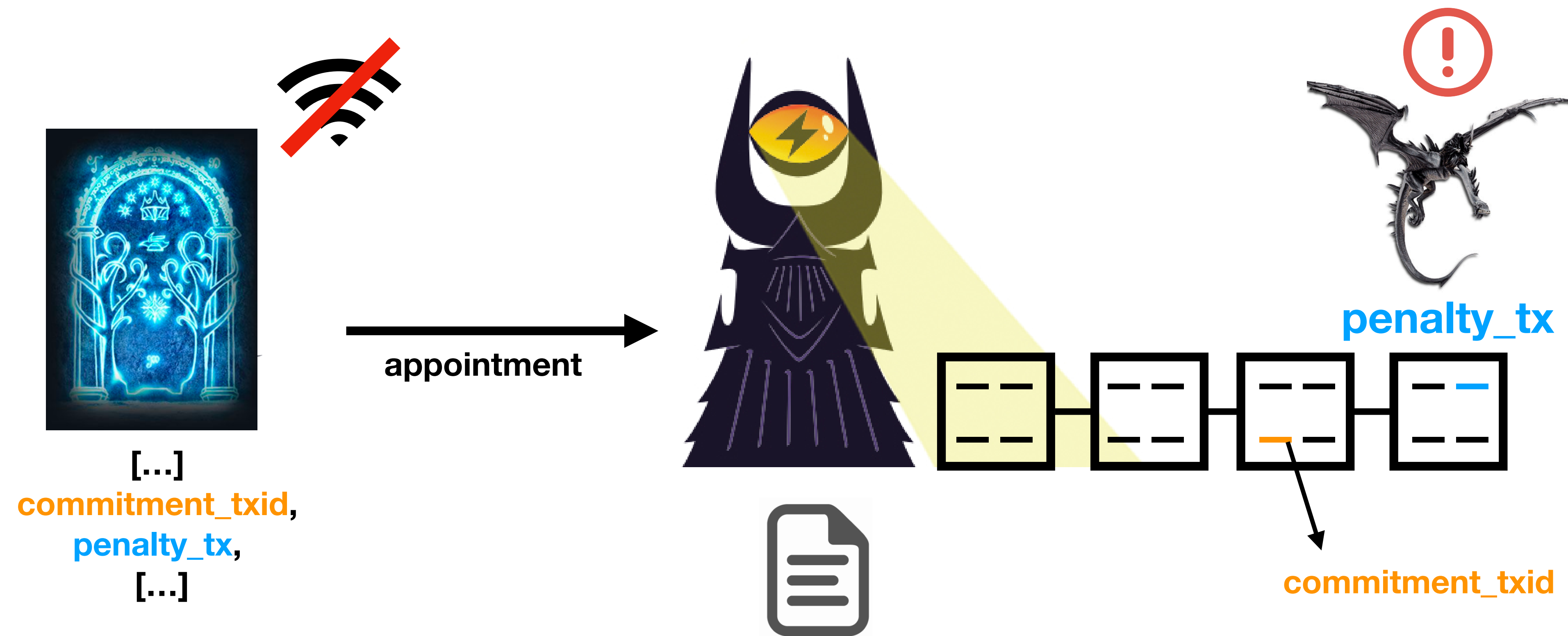
# Protocolo básico de watchtowers



# Protocolo básico de watchtowers

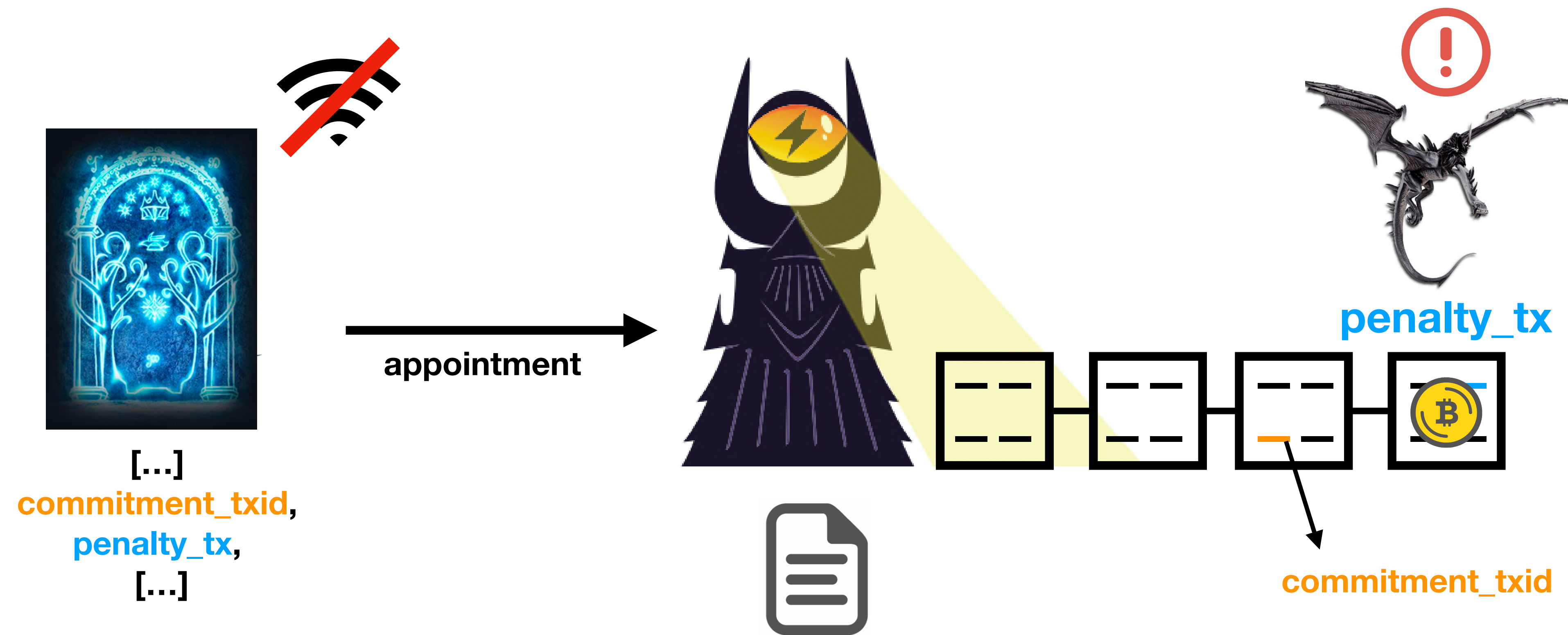


# Protocolo básico de watchtowers

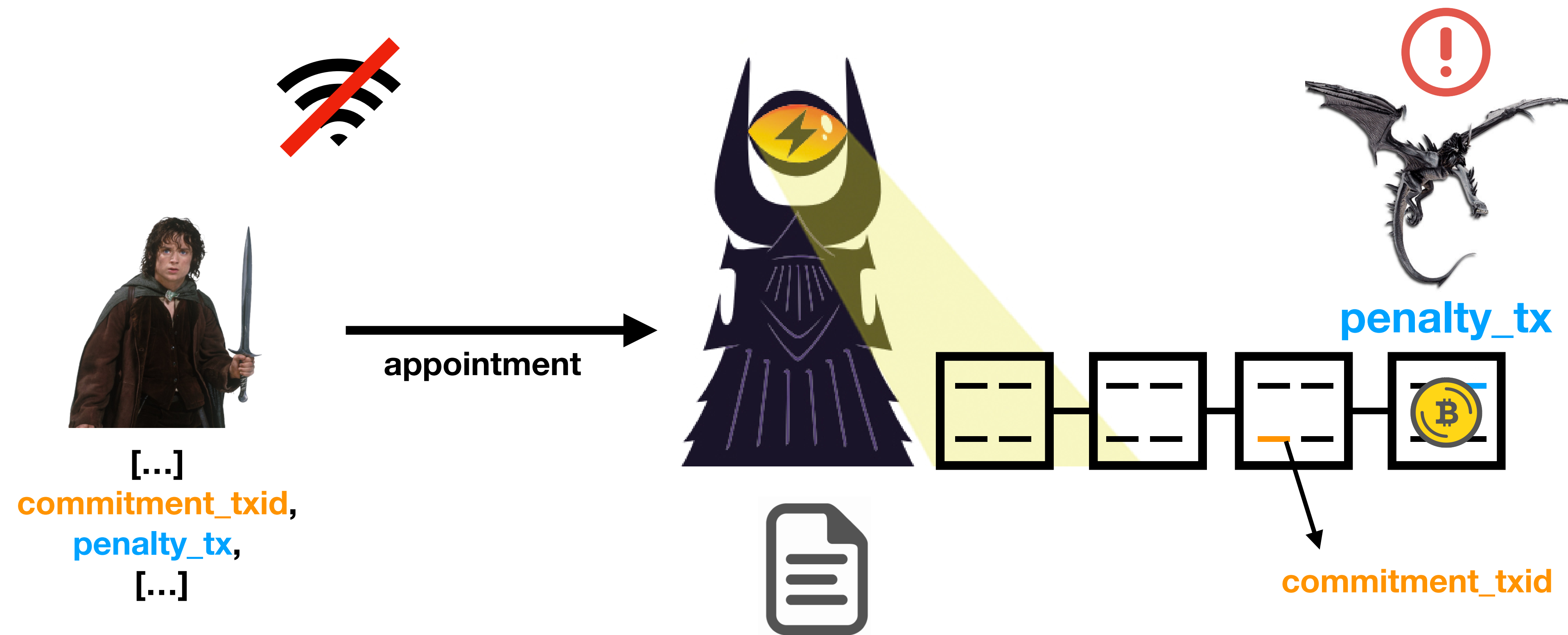




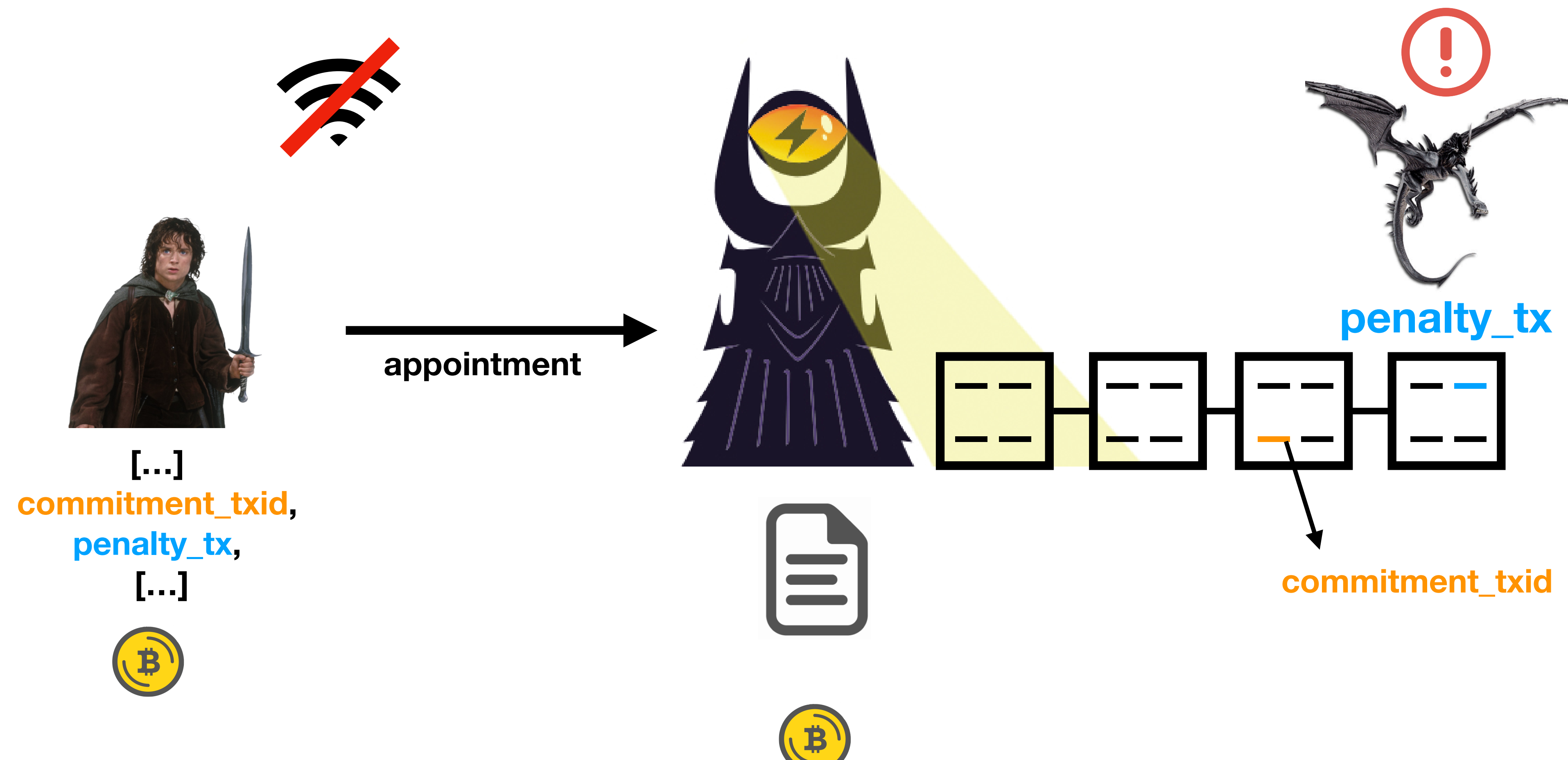
# Protocolo básico de watchtowers



# Protocolo básico de watchtowers



# Protocolo básico de watchtowers



# Watchtowers: Funcionamiento (I)

User side





# Watchtowers: Funcionamiento (I)

User side



**penalty\_tx = 020000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000**

# Watchtowers: Funcionamiento (I)

User side



**penalty\_tx = 020000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000**

**commitment\_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33**

# Watchtowers: Funcionamiento (I)

User side



penalty\_tx = 0200000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000

commitment\_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB

# Watchtowers: Funcionamiento (I)

User side



penalty\_tx = 020000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000

commitment\_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → locator

# Watchtowers: Funcionamiento (I)

User side



penalty\_tx = 0200000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000

commitment\_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → locator

cipher = CHACHA20POLY1305

sk = SHA256(commitment\_txid)

IV = 0

# Watchtowers: Funcionamiento (I)

User side



penalty\_tx = 0200000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000

commitment\_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → locator

cipher = CHACHA20POLY1305

sk = SHA256(commitment\_txid)      encrypt (penalty\_tx, sk, IV)

—————→

IV = 0

# Watchtowers: Funcionamiento (I)

User side



penalty\_tx = 0200000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000

commitment\_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB → locator

cipher = CHACHA20POLY1305

sk = SHA256(commitment\_txid)      encrypt (penalty\_tx, sk, IV) → encrypted blob

IV = 0



# Watchtowers: Funcionamiento (I)

User side



penalty\_tx = 0200000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000

commitment\_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB



locator

cipher = CHACHA20POLY1305

sk = SHA256(commitment\_txid)

encrypt (penalty\_tx, sk, IV)



encrypted blob

IV = 0



# Watchtowers: Funcionamiento (I)

User side



penalty\_tx = 0200000000001010d8b7512b1f530338ca886...1f9624914fb8a68000000000

commitment\_txid = 4a5e1e4baab89f3a32518...cc77ab2127b7afdeda33

16 MSB



locator

SENT TO THE TOWER

cipher = CHACHA20POLY1305

sk = SHA256(commitment\_txid)

encrypt (penalty\_tx, sk, IV)



encrypted blob

IV = 0

# Watchtowers: Funcionamiento (II)

Tower side



# Watchtowers: Funcionamiento (II)

Tower side

for every **transaction\_id** in every block

**locator** = 16 MSB **transaction\_id**



# Watchtowers: Funcionamiento (II)

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: Funcionamiento (II)



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**)

A red arrow starts from the left side of the 'decrypt' line, goes down, and then points horizontally to the right.

# Watchtowers: Funcionamiento (II)



## 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**

# Recursos

## **The Eye of Satoshi**

<https://github.com/talaia-labs/rust-teos>

## **BOLT13**

<https://github.com/sr-gi/bolt13/blob/master/13-watchtowers.md>

## **c-lightning plugin**

<https://github.com/talaia-labs/rust-teos/tree/master/watchtower-plugin>

# Bitcoin: From Zero to Hero

---

Sergi Delgado

