Atomic Swaps on ⚡

Atomic Swaps on ⚡ (not only)

Atomic Swaps?

@kilrau

Recap – Atomic Swaps

1. Alice creates secret
2. Alice pays Bob (pending, Bob doesn't have secret)
3. Bob pays Alice (pending, Alice needs to "release" secret)
4. Alice releases secret = both payments execute atomically because depend on the same secret.

@kilrau

What we are building @exchange_union

A decentralized exchange protocol

on payment channels.

 @kilrau

What we are building @exchange_union

A decentralized exchange protocol

on ⚡lightning⚡ & raiden.

 @kilrau

Combining two worlds

Trading via Atomic Swaps between

BTC, LTC & any ERC20 token (w-ETH, DAI, USDT..)

@kilrau

Differences between ⚡ & Raiden

1. Invoices vs. direct payments
    - LN: payee creates secret
    - Raiden: payer creates secret
2. Routing
    - LN: Source Routing
    - Raiden: dynamic routing (like the internet)
3. Hash Algorithms
    - LN: SHA256
    - Raiden: Keccak256

@kilrau

Combining two worlds

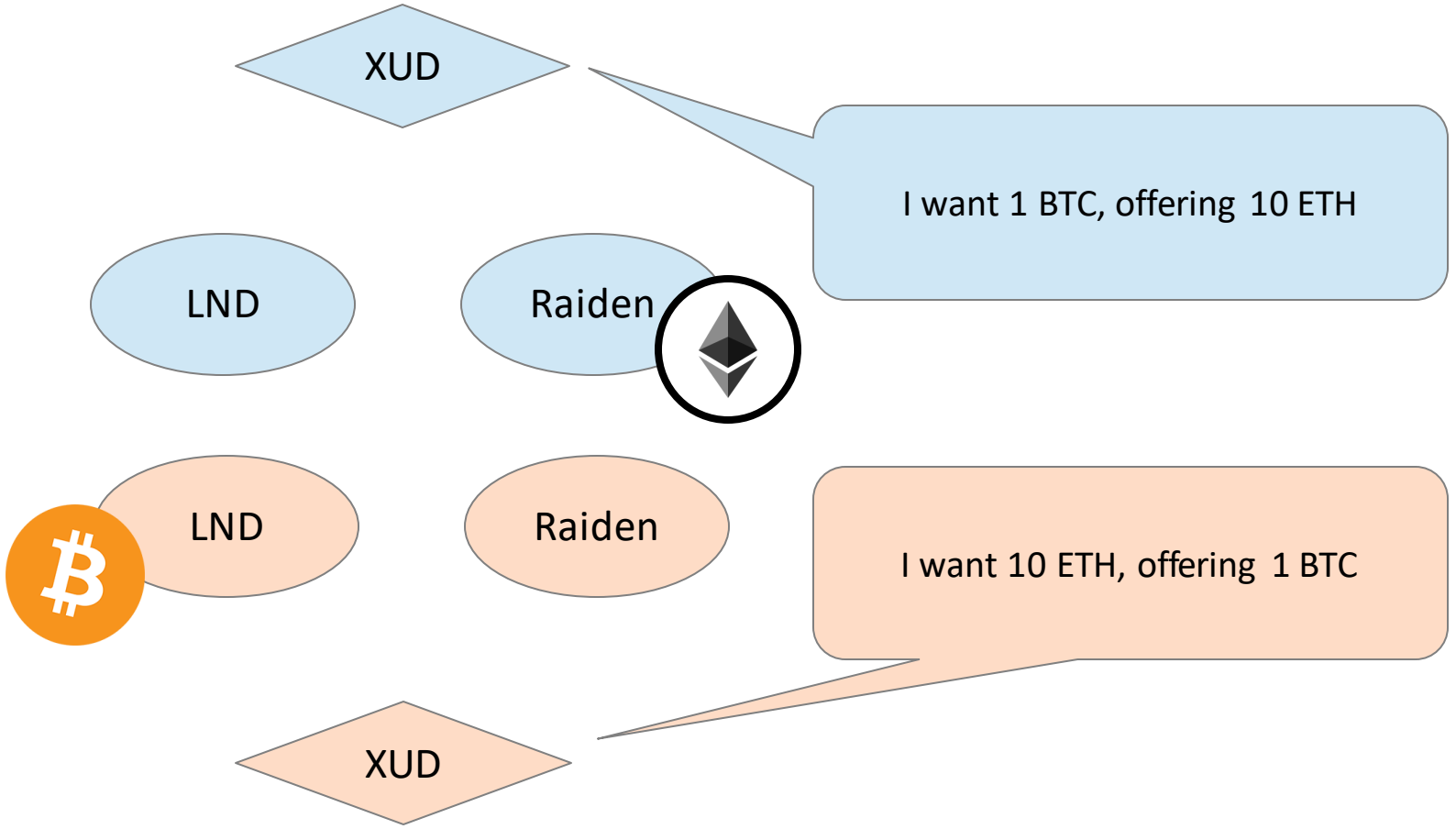
Lightning: HOLD invoices

Raiden: direct payments (+ resolver)

@kilrau

Atomic Swaps between ⚡ & Raiden

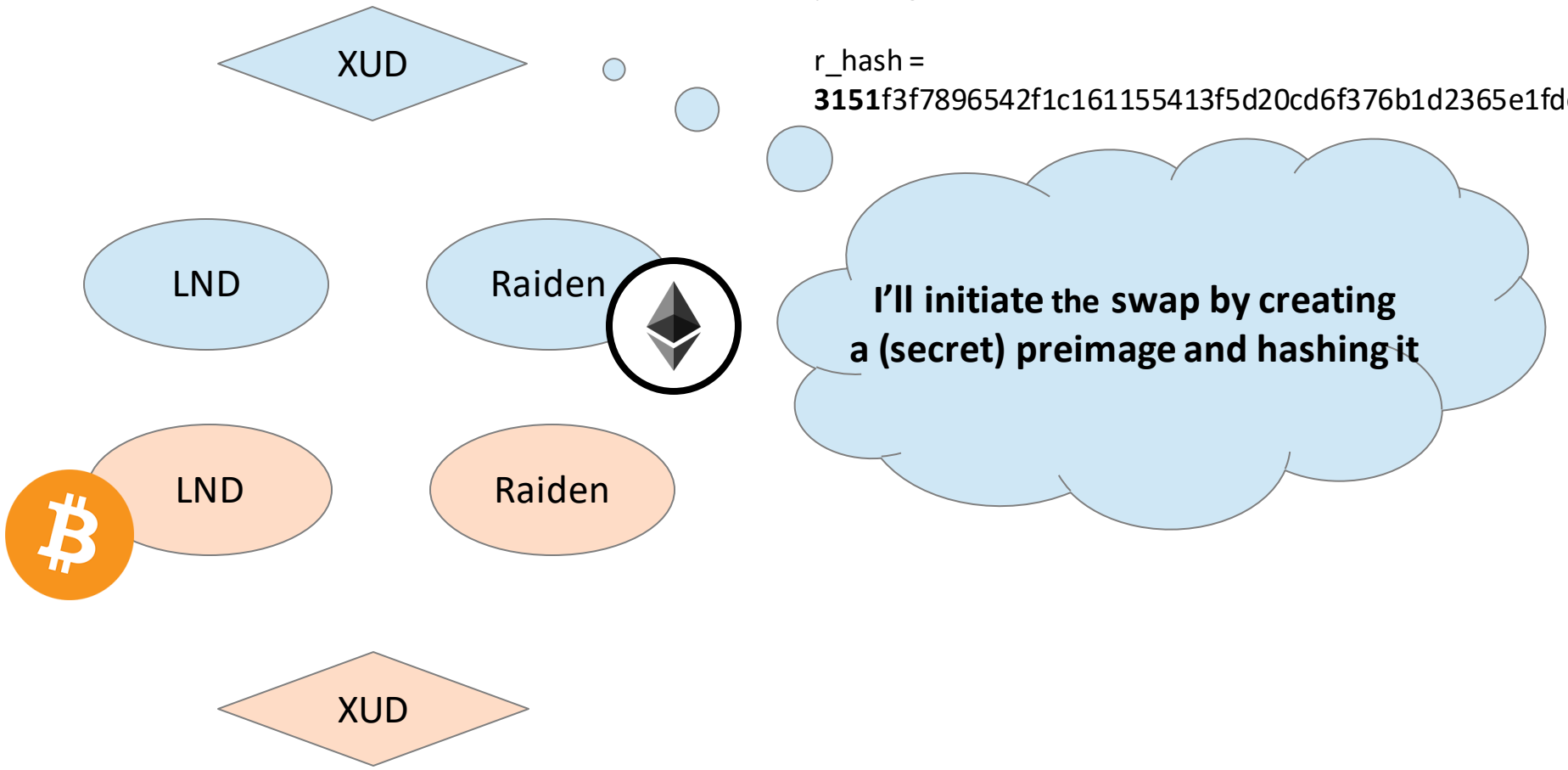
Step 0:



@kilrau

Atomic Swaps between ⚡ & Raiden

Step 0:



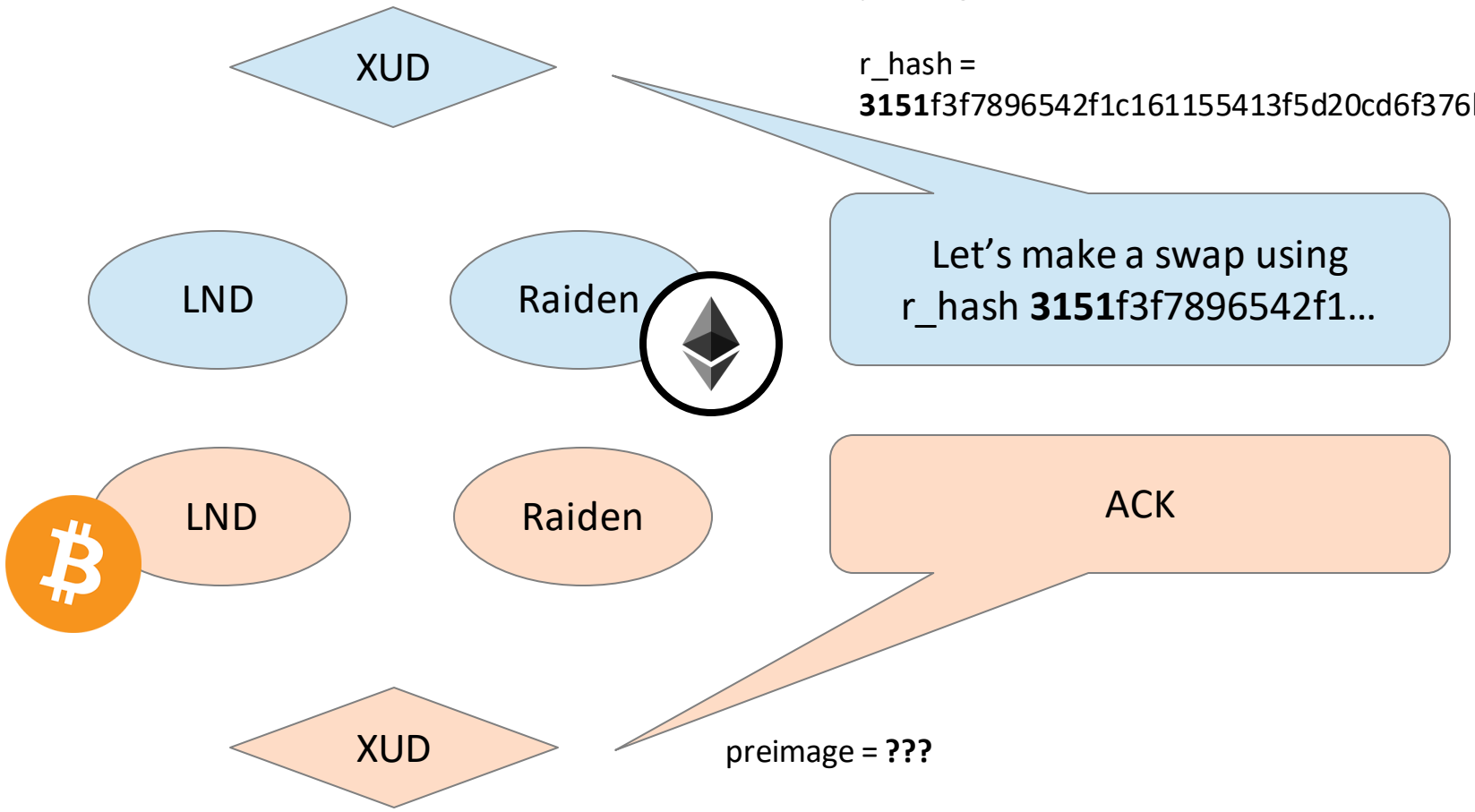
preimage = **a5b83**413f5d20cd6f376b1d2365e1fde900876542

r_hash = **3151**f3f7896542f1c161155413f5d20cd6f376b1d2365e1fde900d2218a7e990

@kilrau

Atomic Swaps between ⚡ & Raiden

Step 1:



preimage = **a5b83**413f5d20cd6f376b1d2365e1fde900876542

r_hash =
3151f3f7896542f1c161155413f5d20cd6f376b1d2365e1fde900d2218a7e990

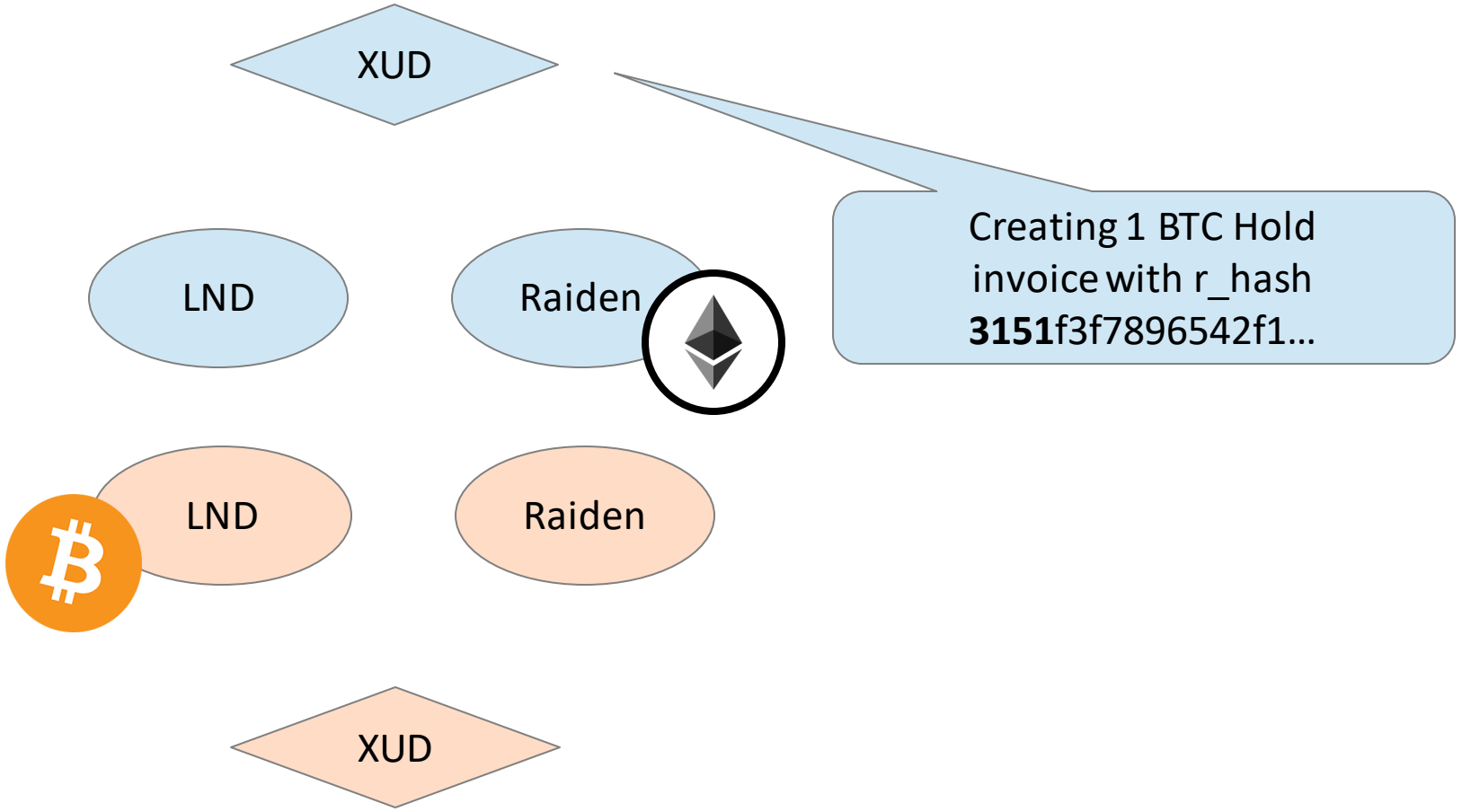
preimage = **???**

r_hash =
3151f3f7896542f1c161155413f5d20cd6f376b1d2365e1fde900d2218a7e990

@kilrau

Atomic Swaps between ⚡ & Raiden

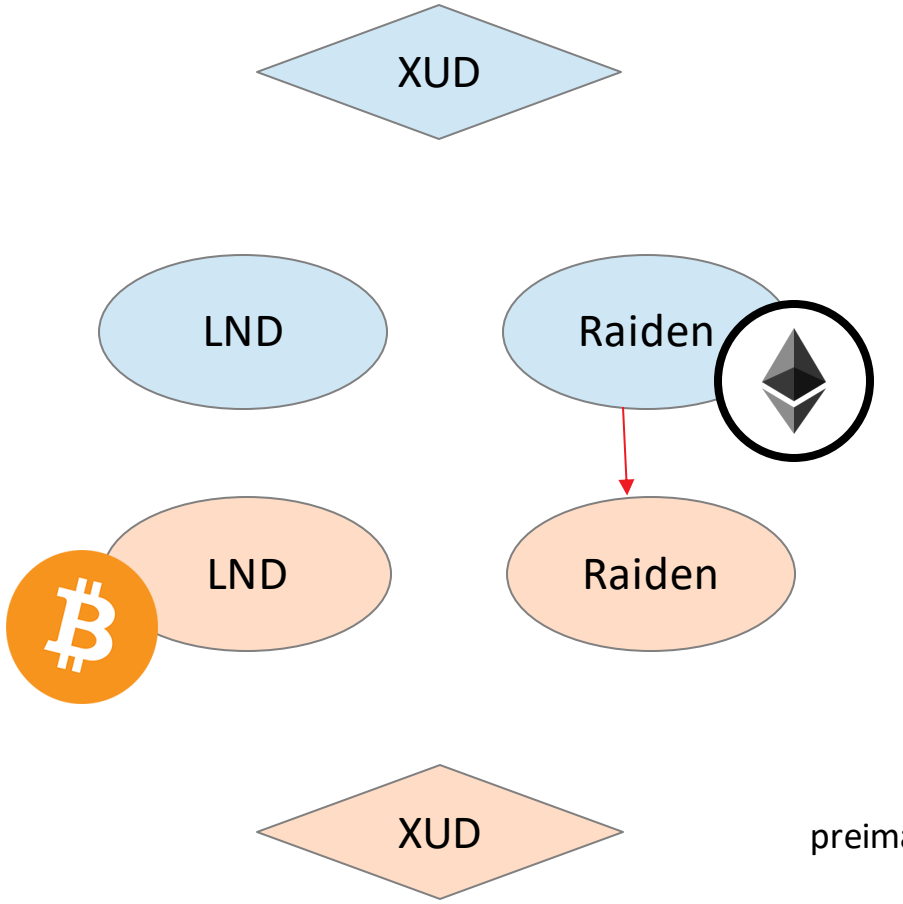
Step 2:



@kilrau

Atomic Swaps between ⚡ & Raiden

Step 2:



preimage = **a5b83**413f5d20cd6f376b1d2365e1fde900876542

r_hash = **3151**f3f7896542f1c161155413f5d20cd6f376b1d2365e1fde900d2218a7e990

Pending HTLC

amount = **10 ETH**
r_hash = **3151**f3f7896542f1…

Note: This payment can route through any number of intermediary nodes on the Raiden network

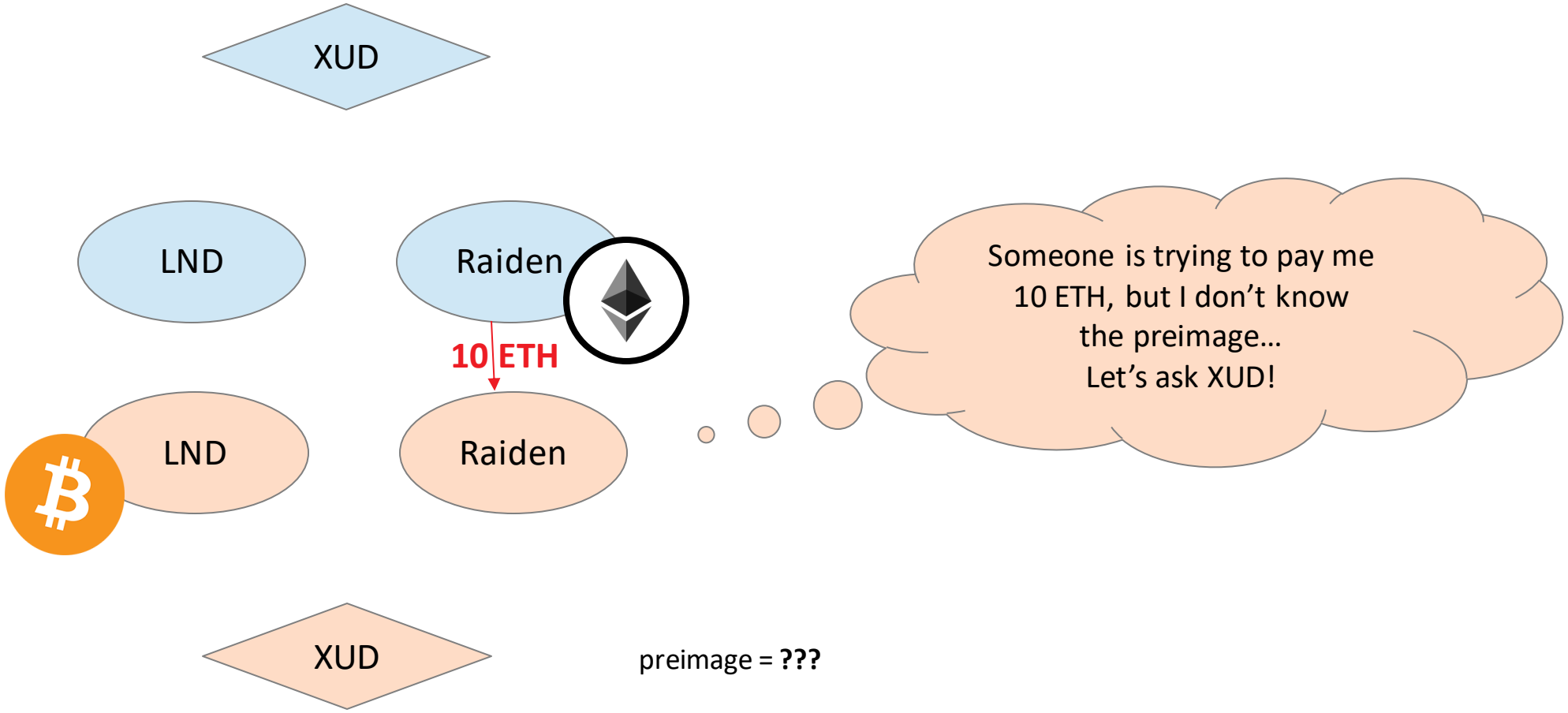
preimage = **???**

r_hash = **3151**f3f7896542f1c161155413f5d20cd6f376b1d2365e1fde900d2218a7e990

@kilrau

Atomic Swaps between ⚡ & Raiden

Step 3:



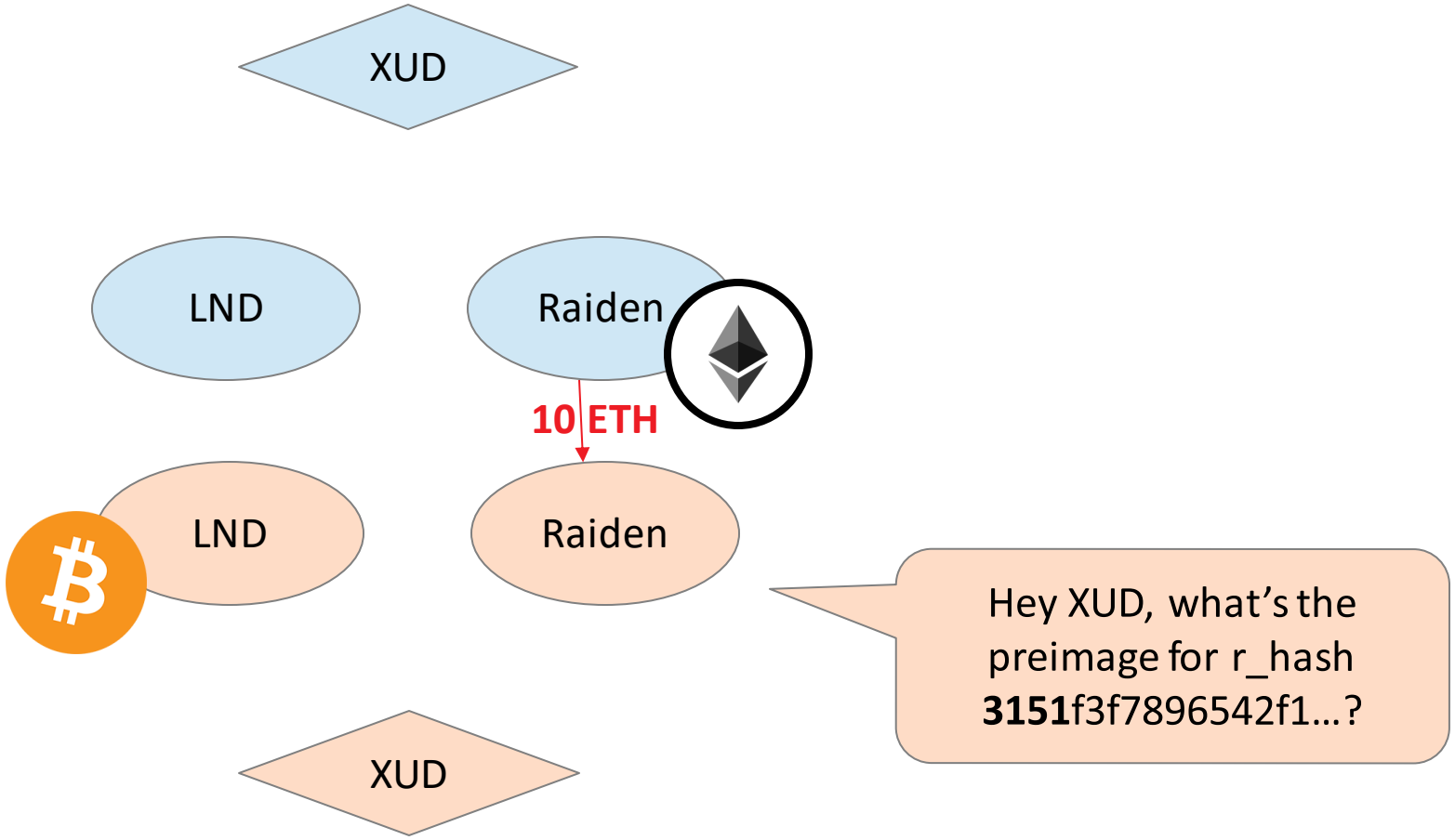
preimage = **???**

r_hash =
3151f3f7896542f1c161155413f5d20cd6f376b1d2365e1fde900d2218a7e990

@kilrau

Atomic Swaps between ⚡ & Raiden

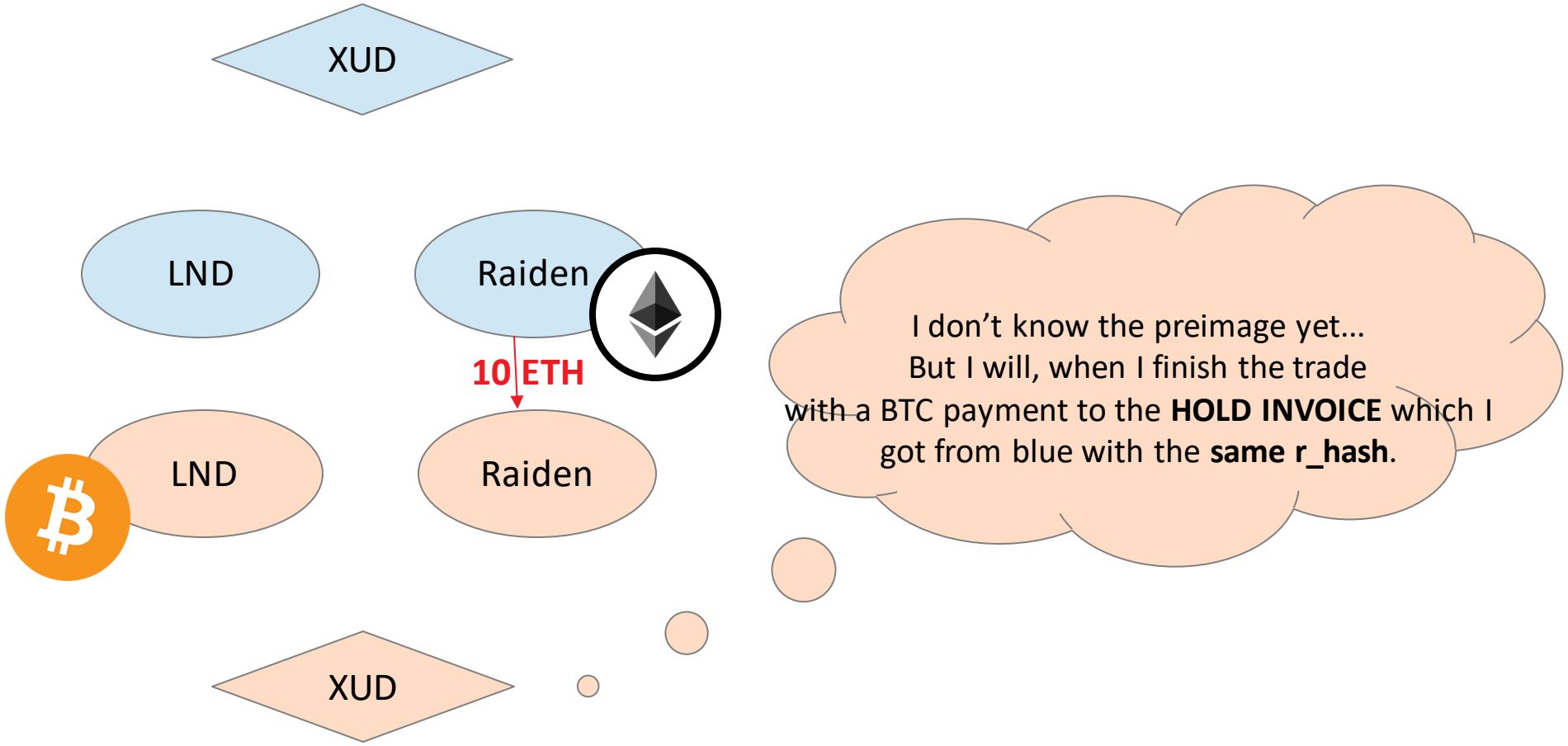
Step 3:



@kilrau

Atomic Swaps between ⚡ & Raiden

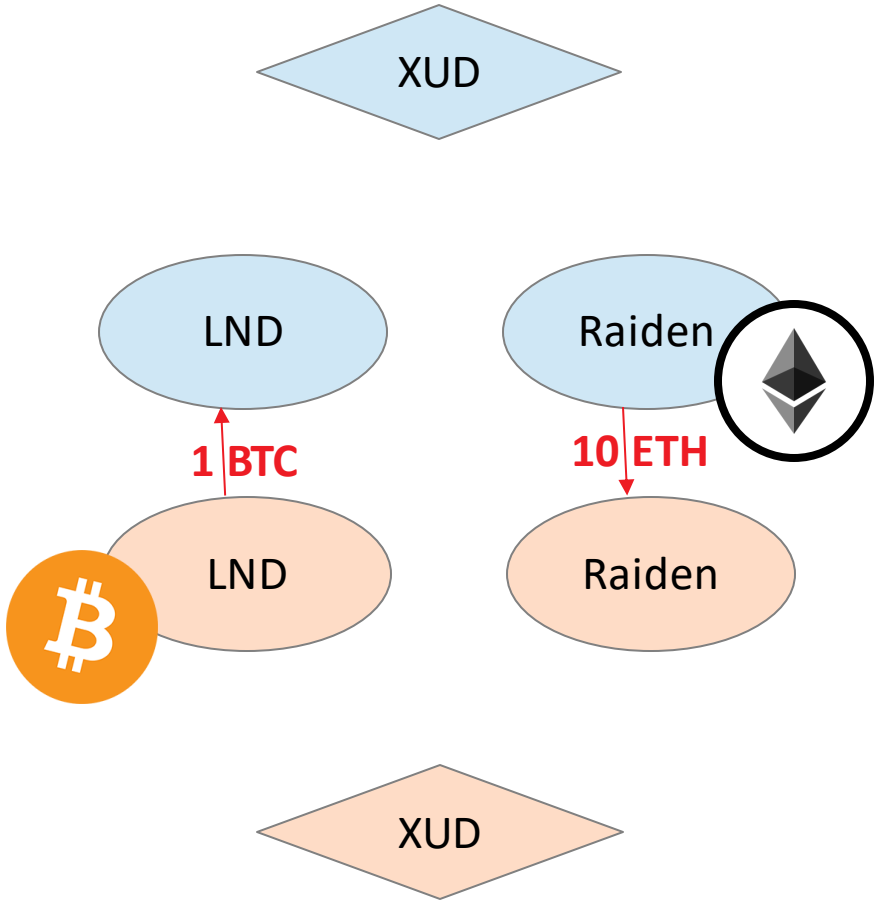
Step 4:



@kilrau

Atomic Swaps between ⚡ & Raiden

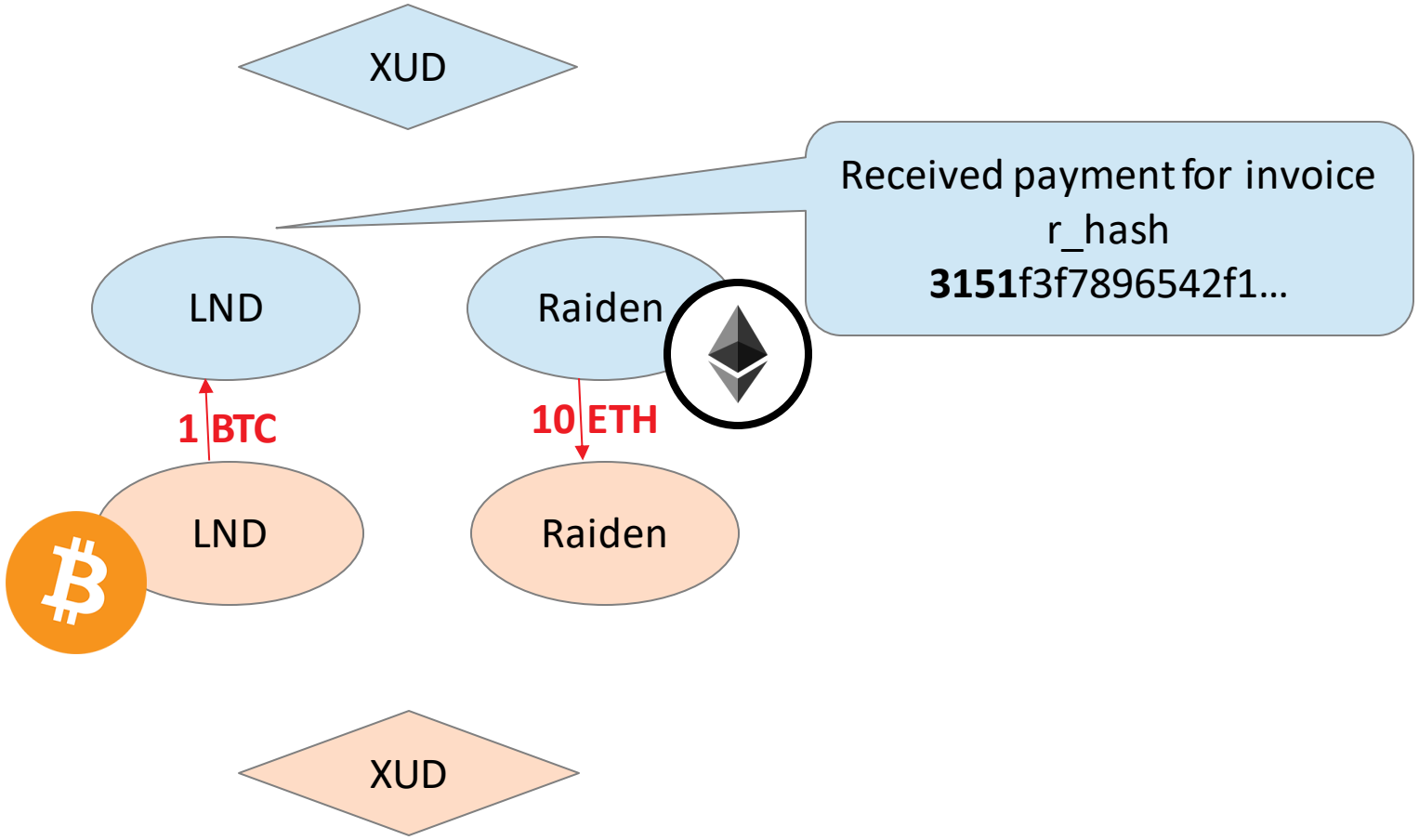
Step 4:



@kilrau

Atomic Swaps between ⚡ & Raiden

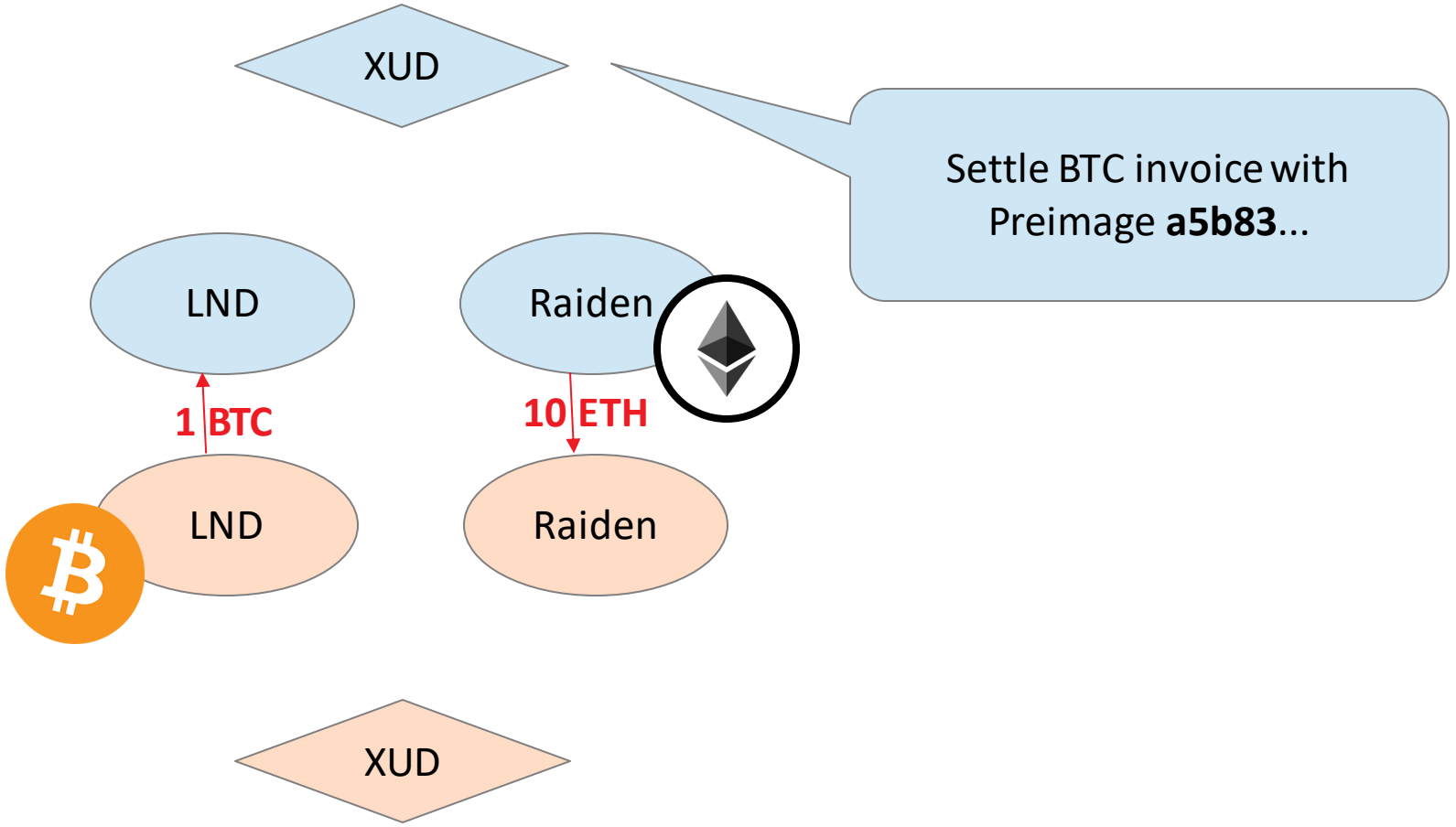
Step 5:



@kilrau

Atomic Swaps between ⚡ & Raiden

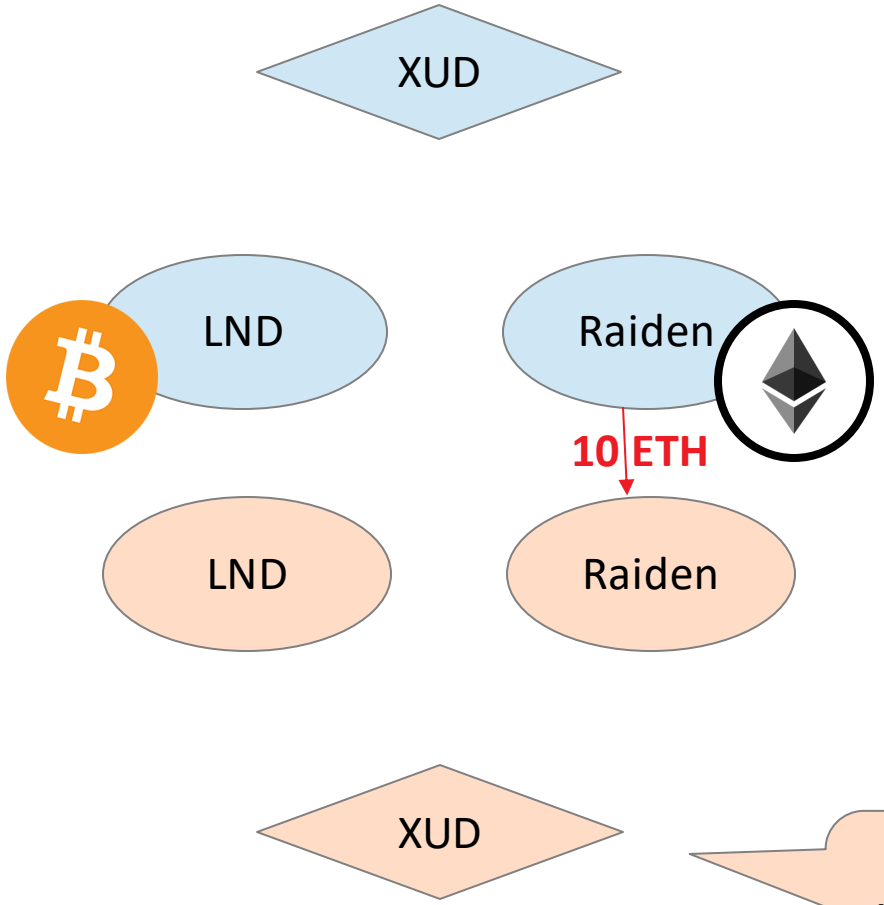
Step 6:



@kilrau

Atomic Swaps between ⚡ & Raiden

Step 6:

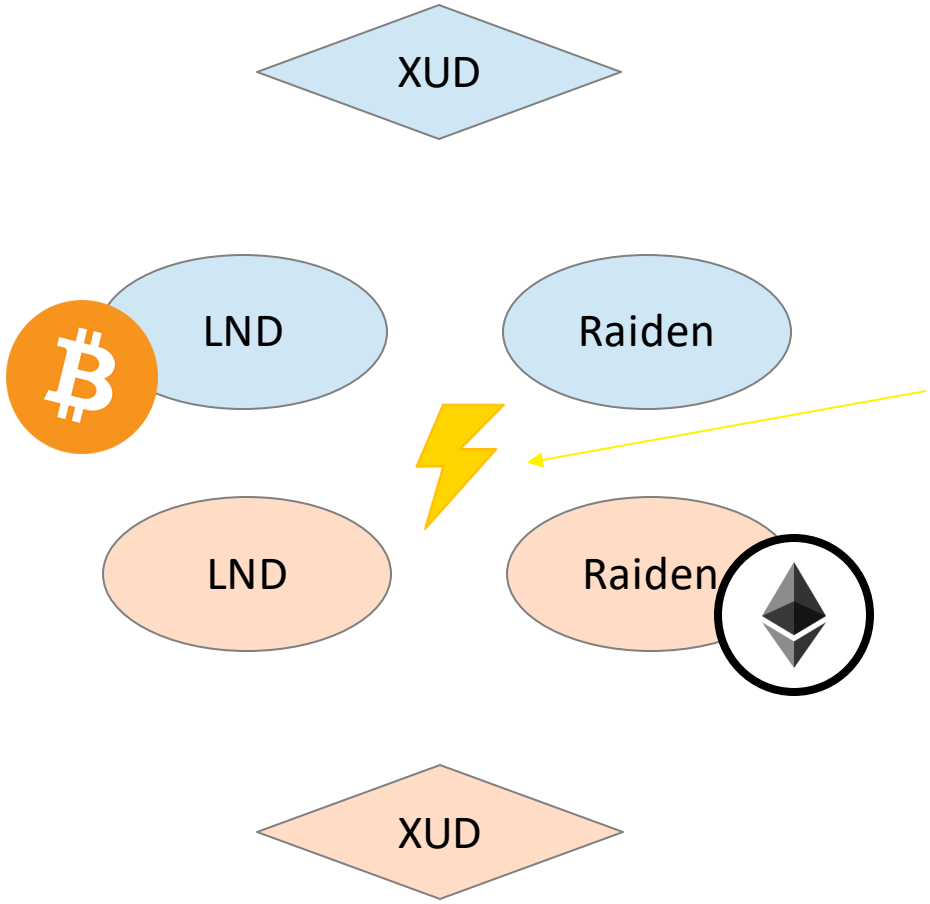


Preimage revealed

@kilrau

Atomic Swaps between ⚡ & Raiden

Step 6:



Preimage revealed

BTC Lightning & ETH Raiden payment settled!

Note: This payment can route through any number of intermediary nodes on the BTC & LTC lightning network

@kilrau

DEMO

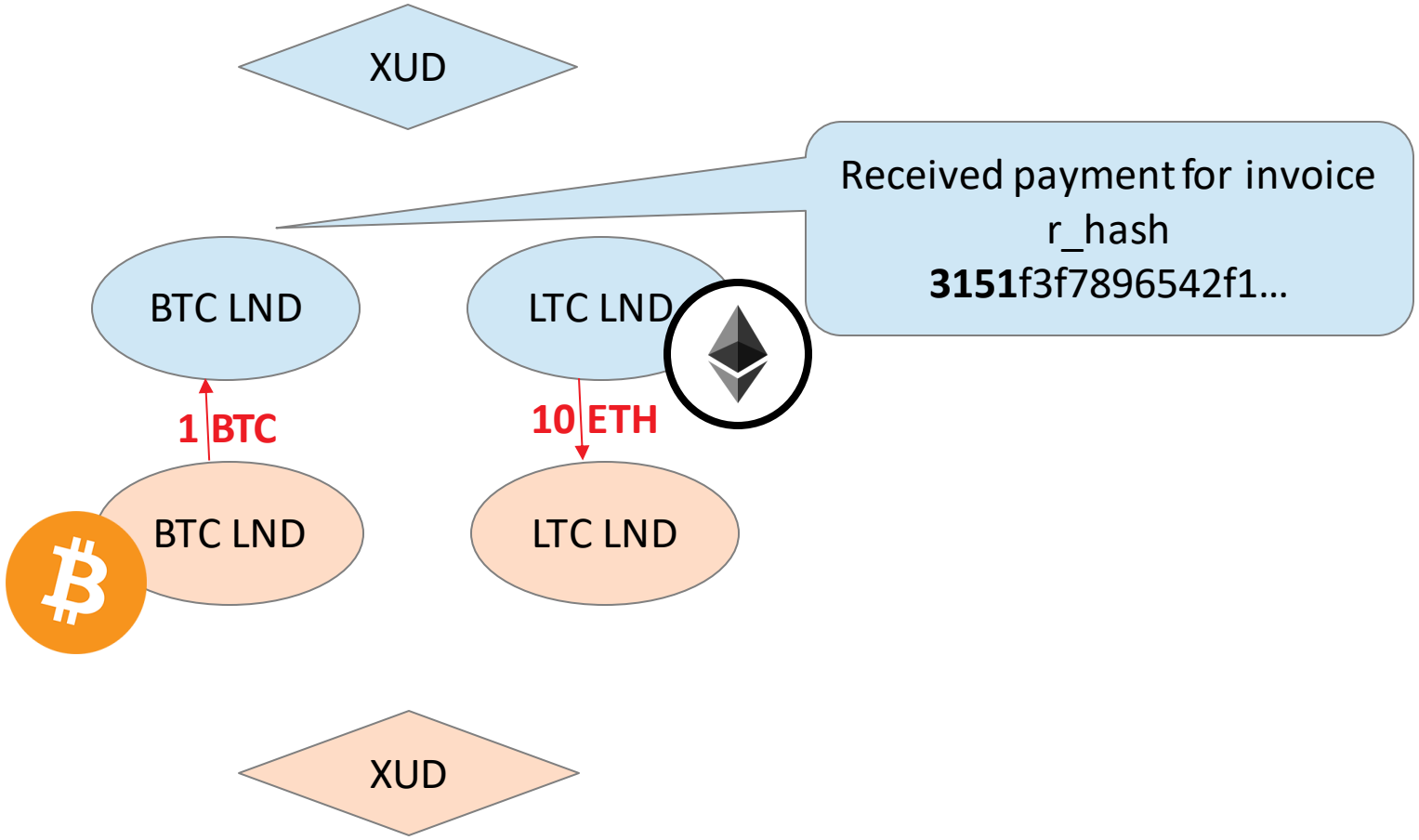
@kilrau

Sign up for early access to Bug Bounty

bit.do/xubounty

But there is a BUT

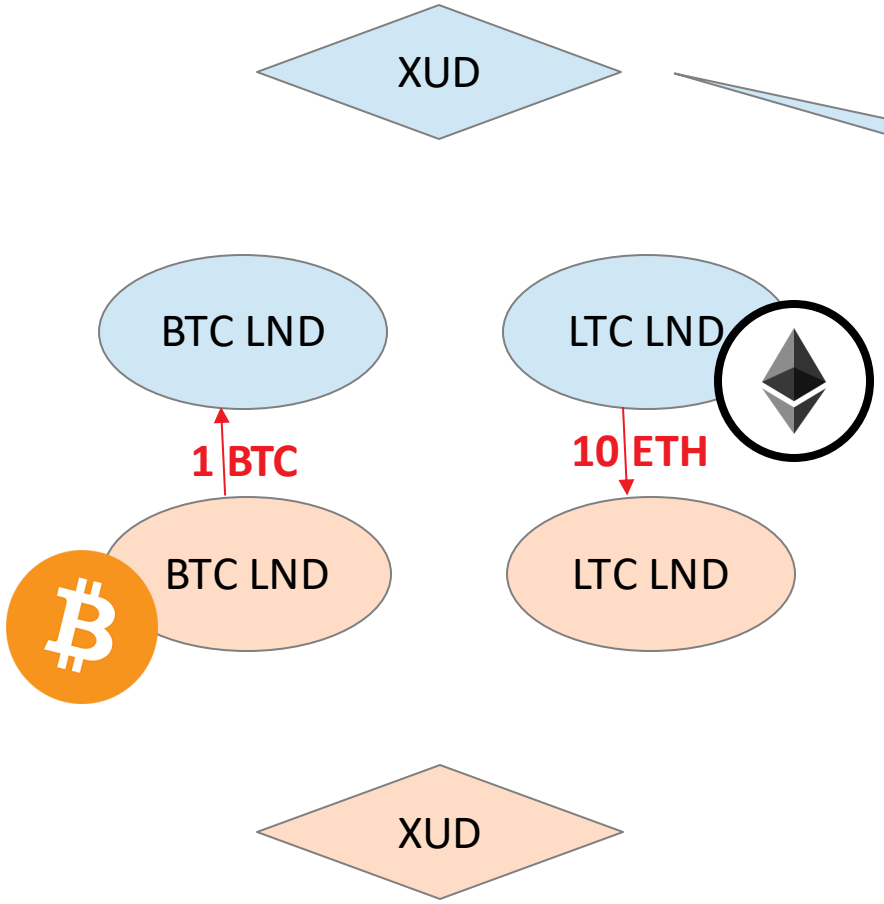
Step 5:



@kilrau

But there is a BUT

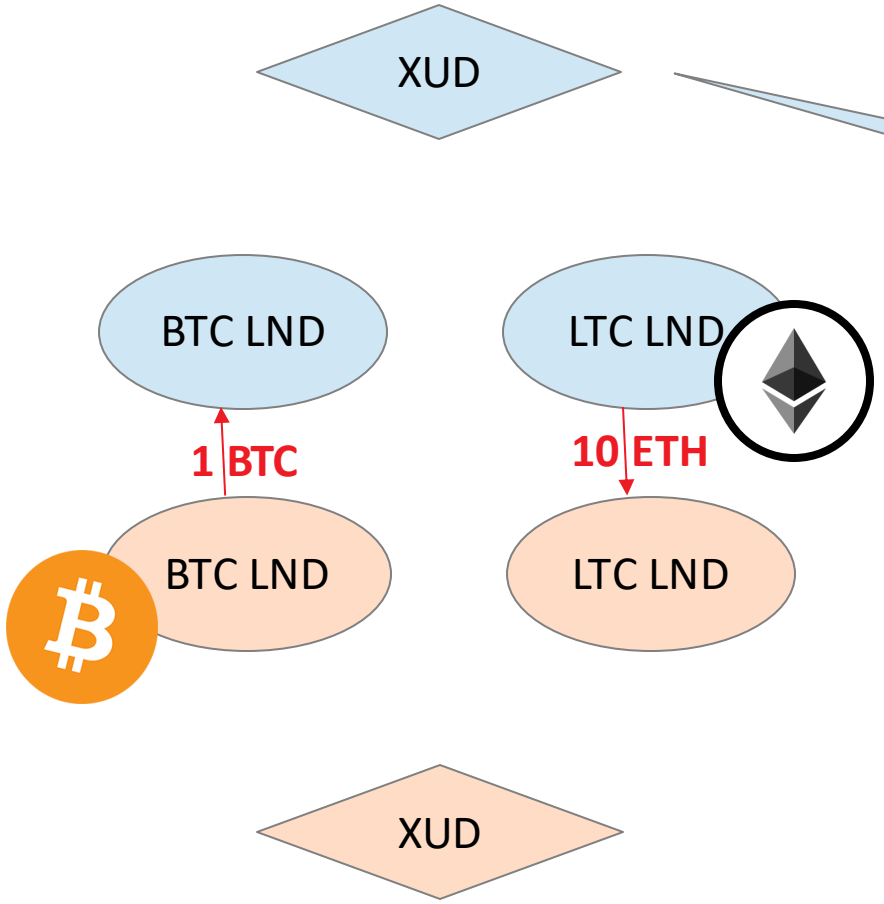
Step 5:



@kilrau

But there is a BUT

Step 5:

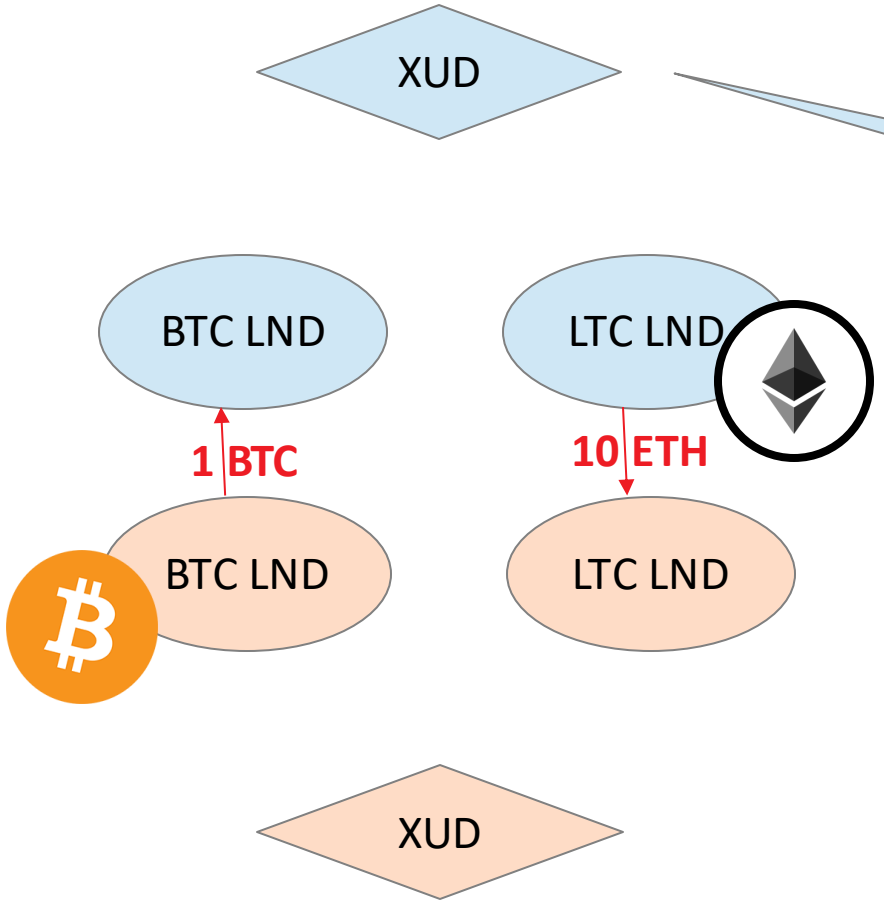


Up to timeout (hours, up to one week)

@kilrau

But there is a BUT

Step 5:



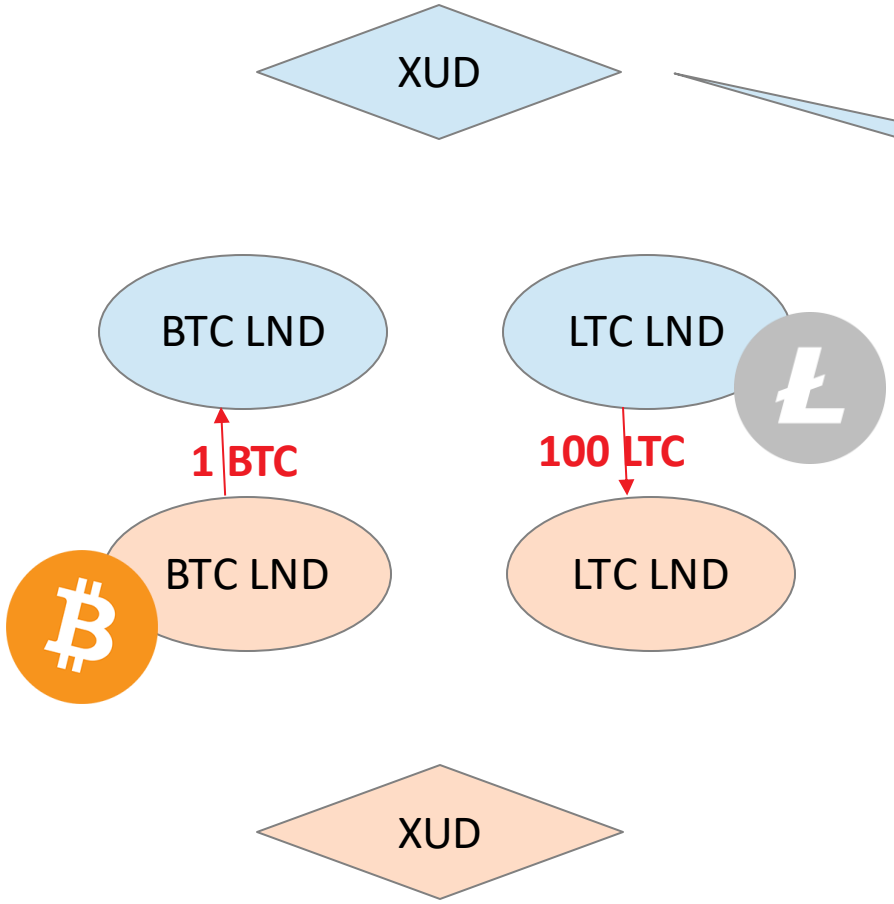
IF ETH vs. BTC went up

wait for timeout to get ETH back

@kilrau

But there is a BUT

Step 5:



IF ETH vs. BTC went up

wait for timeout to get ETH back

ELSE

release preimage

@kilrau

But there is a BUT

Option: right to execute a trade or not in the future

@kilrau

But there is a BUT

Option: right to execute a trade or not in the future

The **problem**: it's for **free**

But there is a BUT

Option: right to execute a trade or not in the future

The **problem**: it's for **free**

"Free Option Problem"

@kilrau

Free Option Problem

Good news: Conceptually solved for on-chain atomic swaps

Free Option Problem

Good news: Conceptually solved for on-chain atomic swaps

Bad News: Still unsolved for lightning-based atomic swaps

@kilrau

Free Option Problem On-Chain

Conceptually solved. How?

Free Option Problem On-Chain

Punishment.

@kilrau

Free Option Problem On-Chain

Punishment.

Both parties **A & B stake collateral** in target asset in additional HTLCs which use the **same preimage** as in atomic swap.

@kilrau

Free Option Problem On-Chain

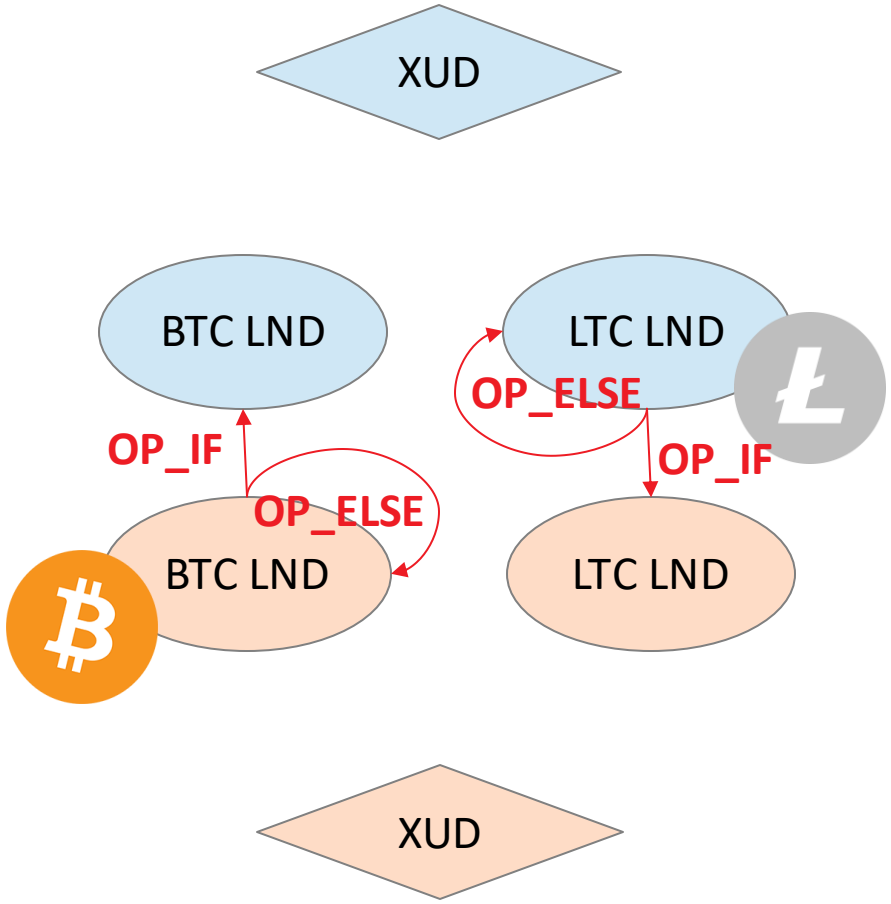
Punishment.

Both parties **A & B stake collateral** in target asset in additional HTLCs which use the **same preimage** as in atomic swap.

Both get collateral back, if preimage is released timely (measured in blocks).

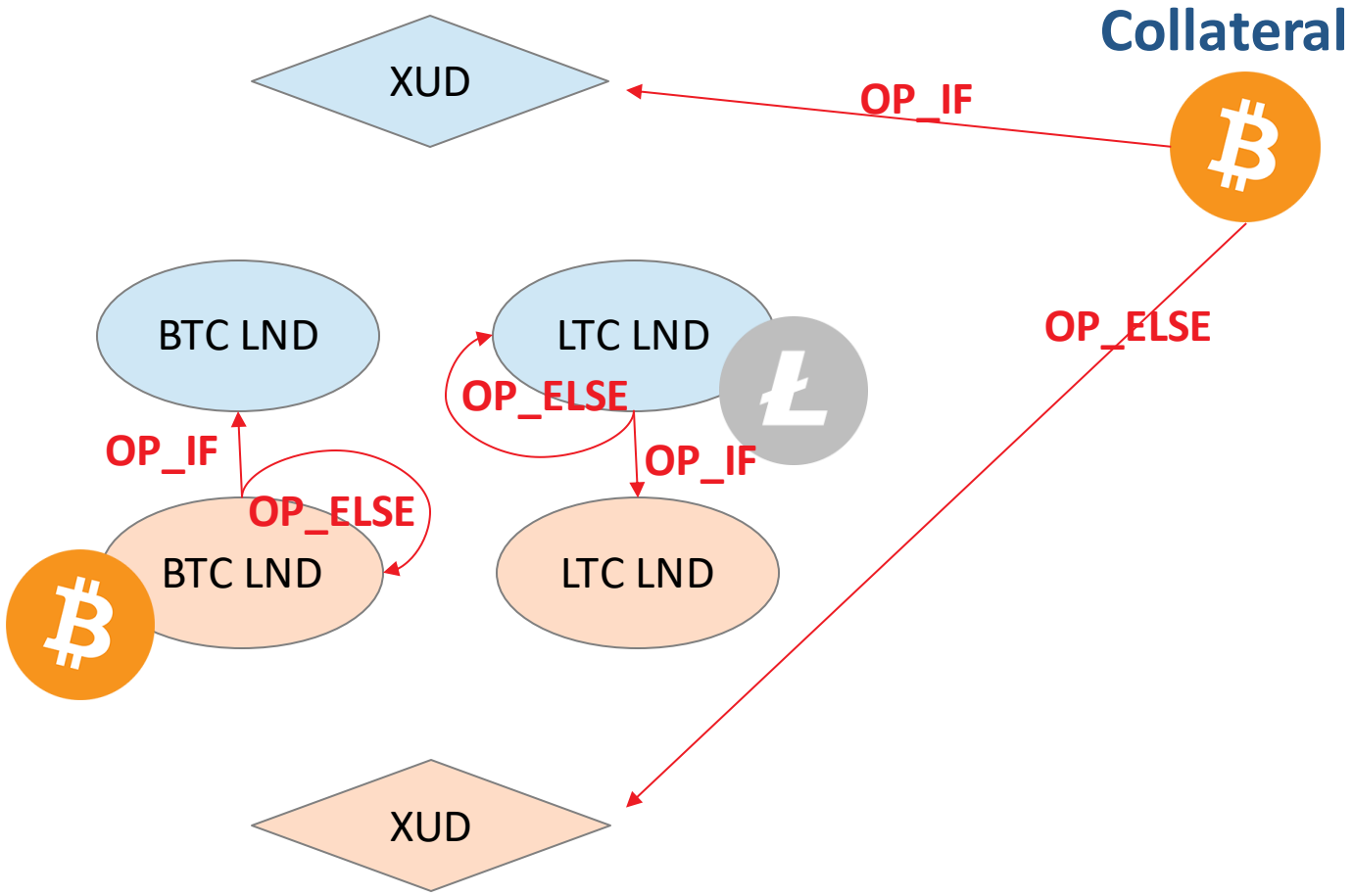
@kilrau

Free Option Problem On-Chain



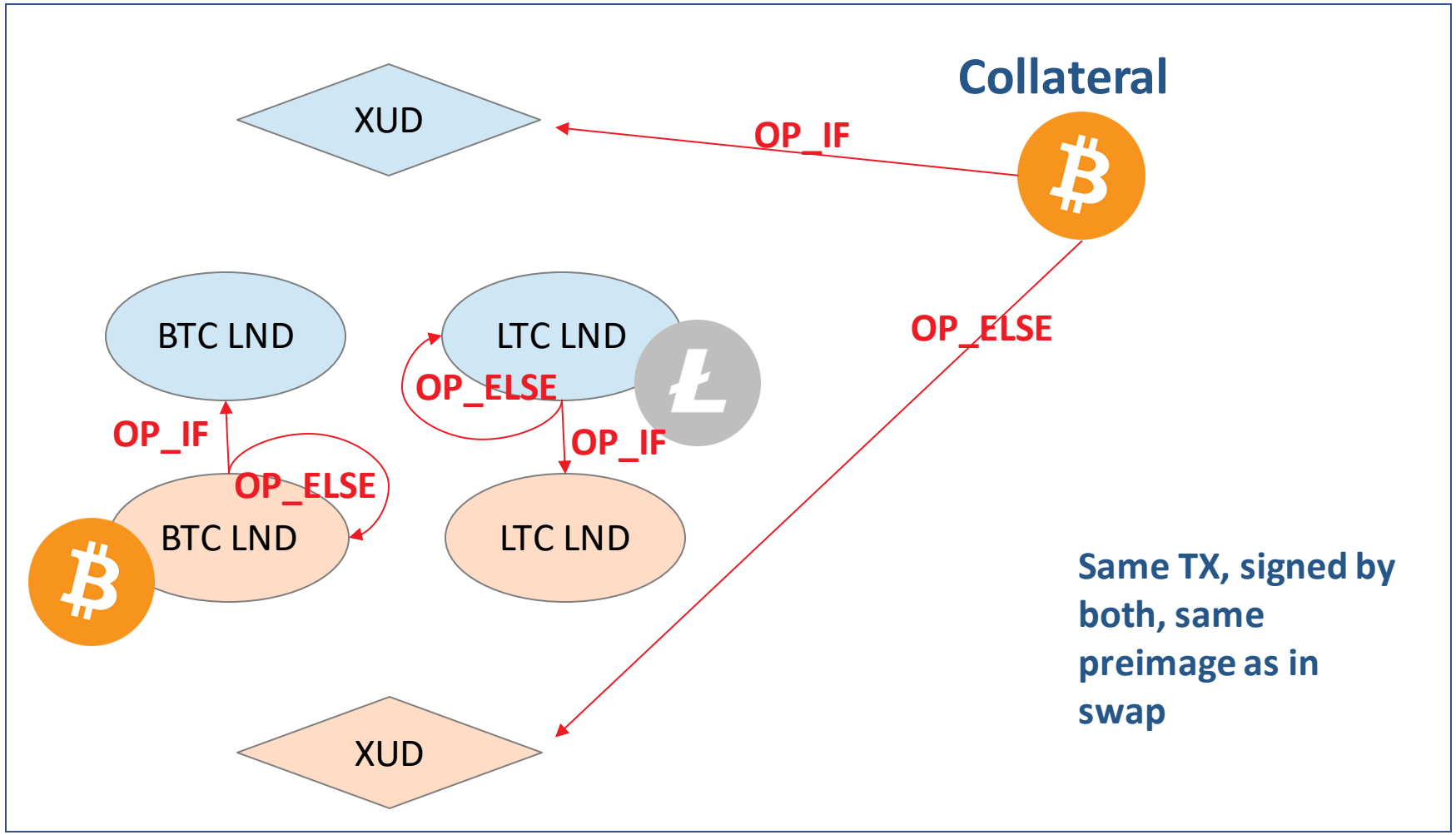
@kilrau

Free Option Problem On-Chain



@kilrau

Free Option Problem On-Chain



@kilrau

Free Option Problem on LN

Why not do the same on LN?

@kilrau

Free Option Problem on LN

Problem:

- Uniquely attribute fault

Free Option Problem on LN

Problem:

- Uniquely attribute fault
- Needed: effective timeout of e.g. 10 seconds. Not hours or days.

Free Option Problem on LN

Problem:

- Uniquely attribute fault
- Needed: effective timeout of e.g. 10 seconds. Not hours or days.
- https://lists.linuxfoundation.org/pipermail/lightning-dev/2018-December/001752.html

@kilrau

Open Research Question

@kilrau

Open Research Question

Join: gitter.im/exchangeunion/FOP

@kilrau

Open Research Question

Join: gitter.im/exchangeunion/FOP

Slides: github.com/exchangeunion/docs

@kilrau

Open Research Question

Join: gitter.im/exchangeunion/FOP

Slides: github.com/exchangeunion/docs

THX!

@kilrau