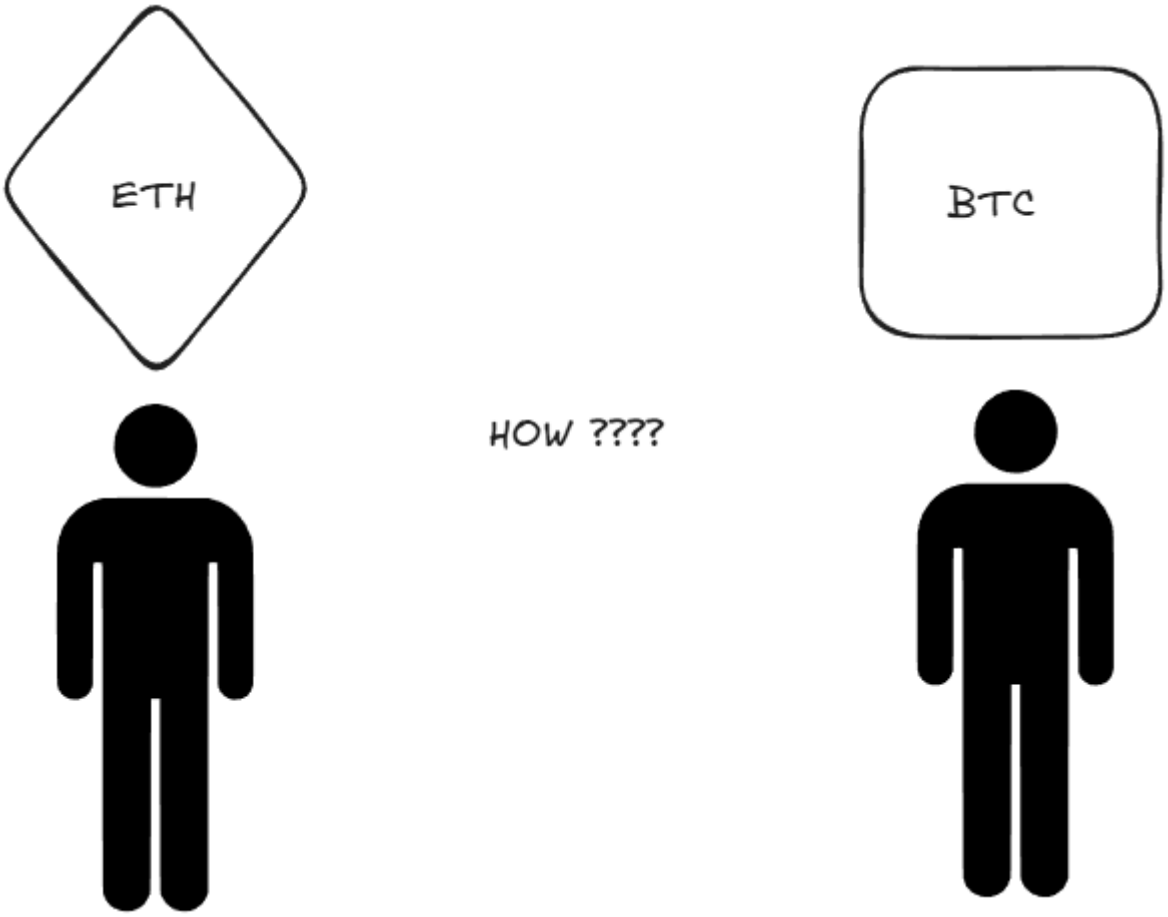
How Atomic Swap Work ?

TOKI

Summary

Introduction

Imagine this: you have Bitcoin (BTC), and I have Ethereum (ETH). You want my ETH, and I want your BTC. How do we make the trade without trusting each other or relying on an exchange?



A centralized exchange? That means giving up control of your funds and trusting a third party. Not ideal.

A direct trade? But what if one of us sends first and the other decides to disappear?

Enter **Atomic Swaps**—a cryptographic way to **exchange assets securely, without trust, and without intermediaries**. The trade either happens **completely** or **not at all**. No middle ground, no risk of getting scammed.

How does it work? Let's break it down.

What is an Atomic Swap?

The Problem with Traditional Swaps

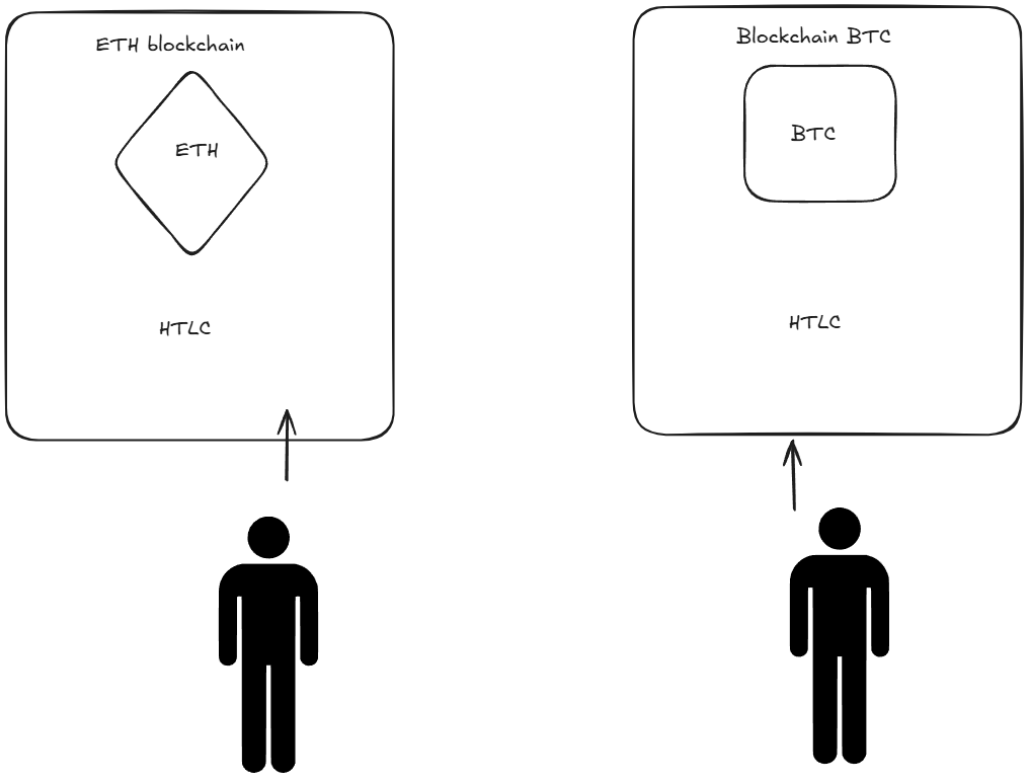
Trading crypto between two different blockchains is tricky. On centralized exchanges, you must trust a third party to hold your funds. With peer-to-peer trades, someone has to send first, which opens the door for scams.

Atomic Swaps solve this by ensuring that either both parties get what they agreed to—or no one gets anything.

What Makes Atomic Swaps Different?

Atomic Swaps use smart contracts to lock the funds until both parties fulfill their part of the deal. These contracts ensure that the trade is all-or-nothing—if one side doesn't hold up their end, the swap is canceled, and both users keep their original funds.

This is possible thanks to a special type of smart contract called a Hash Time-Locked Contract (HTLC).

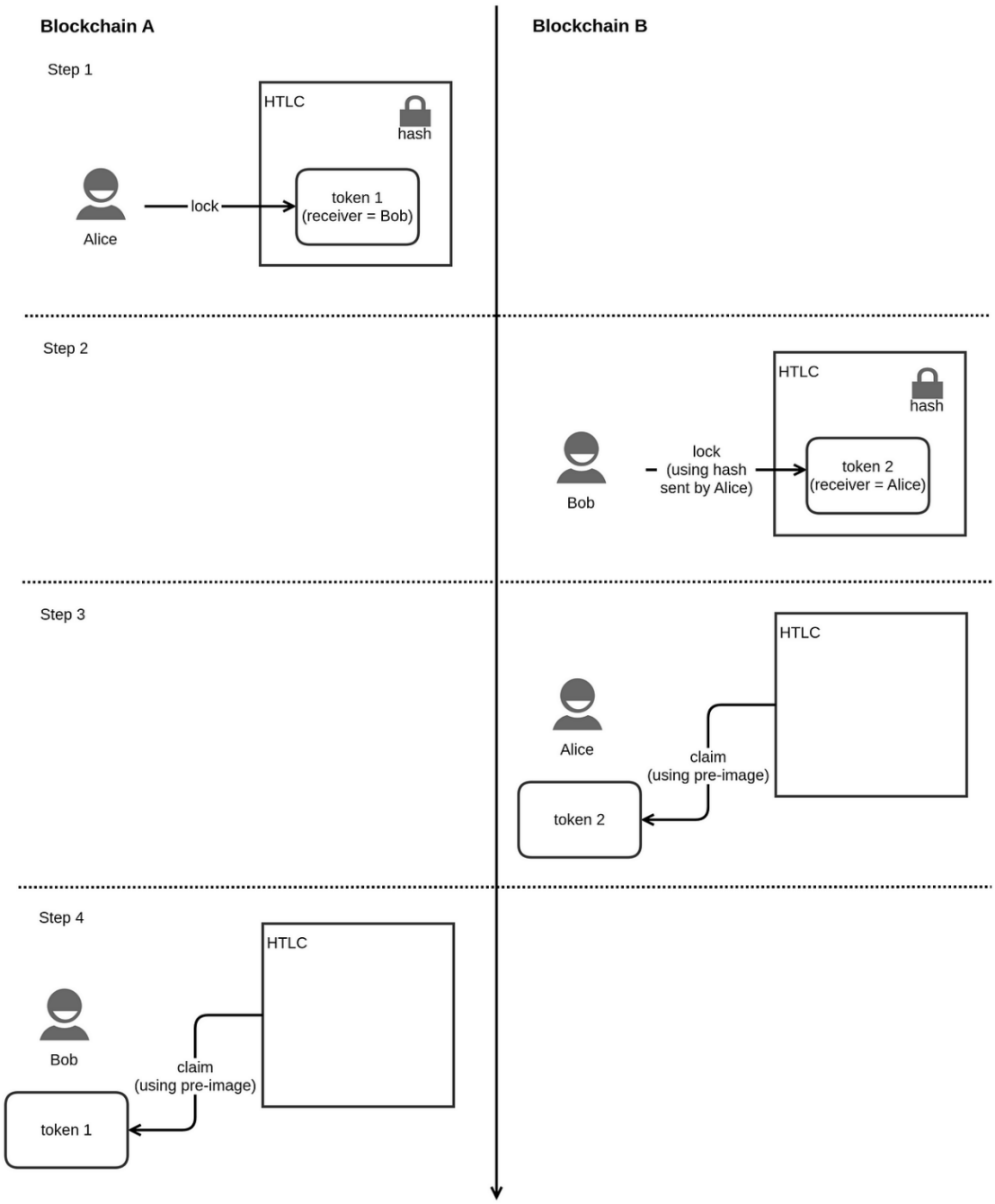


Once both parties have locked their assets in their respective HTLCs, the swap is completed when one party reveals the secret key used to unlock the funds. For example, if the Ethereum trader generates a secret and shares its hash, the Bitcoin trader uses that hash to lock their BTC. When the Ethereum trader claims the BTC, they must reveal the secret in the process. Since the Ethereum contract was locked using the same hash, the Bitcoin trader can now use the revealed secret to unlock their ETH. This ensures that both parties receive their funds securely without the need for trust.

Essential Components of Atomic Swaps

Hash Time-Locked Contracts (HTLCs)

HTLCs are the backbone of Atomic Swaps. They ensure that the trade can only happen if a cryptographic secret is revealed. Otherwise, the funds are refunded after a set time.



Hash and Secret

One party generates a secret, hashes it, and shares only the hash with the other party. This hash acts as a kind of "lock" on the funds. The only way to claim the funds is to reveal the original secret.

On Ethereum, a simple way to generate this hash is by using Keccak-256, the hashing function behind Ethereum's keccak256() function. A user can do this in Solidity with:

```
bytes32 secret = keccak256(abi.encodePacked(VALUE));
```

On Bitcoin, hashing is typically done using SHA-256 instead of Keccak-256. The equivalent process on Bitcoin involves generating the secret off-chain and hashing it with SHA-256 before locking funds in an HTLC script. This can be done with:

```
echo -n "my-value" | sha256sum
```

The resulting hash is then embedded in a Bitcoin Script HTLC, ensuring that the funds can only be claimed by revealing the original secret.

Time Lock

A time lock ensures that if something goes wrong—like one party not claiming their funds—the swap gets automatically canceled, and the funds return to their owners.

This guarantees that no one can get stuck in a half-completed trade.

The Atomic Swap Process

Setting the Terms

Before anything happens, both traders agree on:

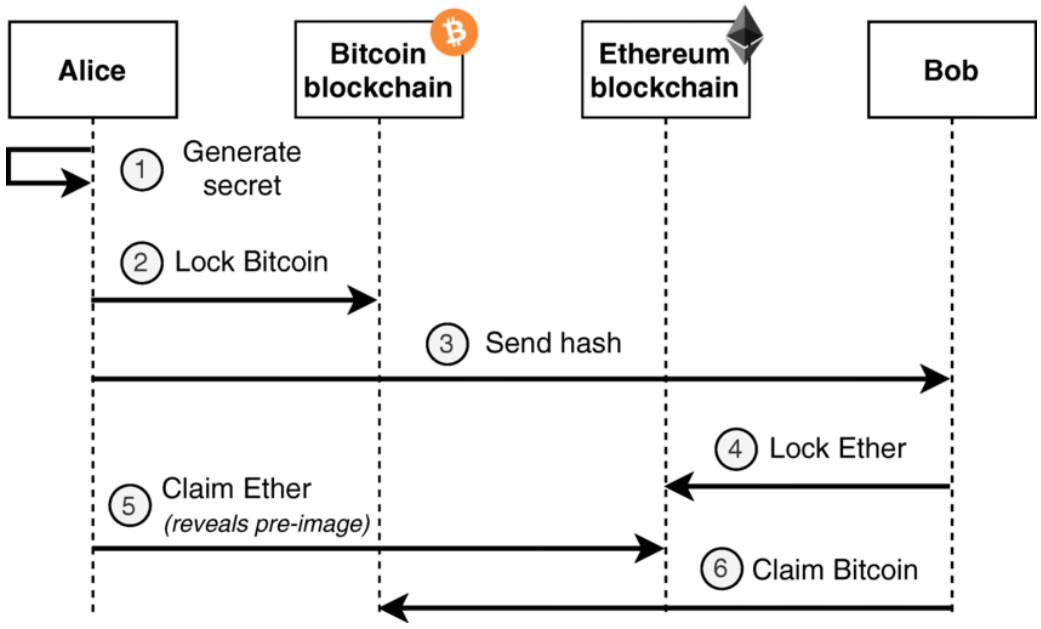
- The amount of BTC and ETH being exchanged
- The time limit for the swap
- The hash used to lock the funds

Deploying Smart Contracts on Ethereum and Bitcoin

1. The BTC owner creates an HTLC on the Bitcoin blockchain, locking their BTC using the hash.
2. The ETH owner sees the BTC is locked and creates an HTLC on the Ethereum blockchain using the same hash.
3. Now both funds are locked, and neither party can access them yet.

Unlocking Funds and Completing the Swap

1. The ETH owner **reveals the secret** to claim the BTC.
2. The BTC owner sees the secret and uses it to unlock the ETH.
3. The swap is complete—both parties get their funds.



Security of Atomic Swaps

Conditions for Cancellation and Refund

Atomic Swaps are designed to ensure that if one party fails to complete their part of the trade, neither side loses their assets. This is achieved through time-lock conditions, which specify that if the swap is not finalized within a certain timeframe, both parties can reclaim their original funds.

Protection Against Attacks

Atomic Swaps are fundamentally designed to be secure and trustless, ensuring that both parties either complete the trade or get their funds back. By using Hash Time-Locked Contracts (HTLCs), the protocol guarantees that no one can unilaterally take the other party's funds without revealing the necessary cryptographic proof. Since swaps occur on-chain, they inherit the security of their respective blockchains, making them resistant to centralized exchange hacks and counterparty fraud.

However, one potential risk is the timing attack, where a malicious actor waits until the last possible moment to reveal the secret. This could create issues if the blockchain is congested, potentially leading to higher transaction fees or even forcing the counterparty to miss their time window to claim their funds. To mitigate this, users should set sufficient time-lock durations and monitor network conditions to ensure they can finalize transactions before expiration.

Despite such edge cases, Atomic Swaps remain one of the most secure ways to perform cross-chain trades

Conclusion

You might wonder, how is an HTLC actually coded? What specific conditions ensure that it works securely across different blockchains? How does it enforce the "all-or-nothing" nature of Atomic Swaps, making sure that no one can cheat the system?

I won't spoil the answer for you—go explore and see for yourself 😊

Github: gitToki