2023

Bitcoin Bridges: Cure or Curse?

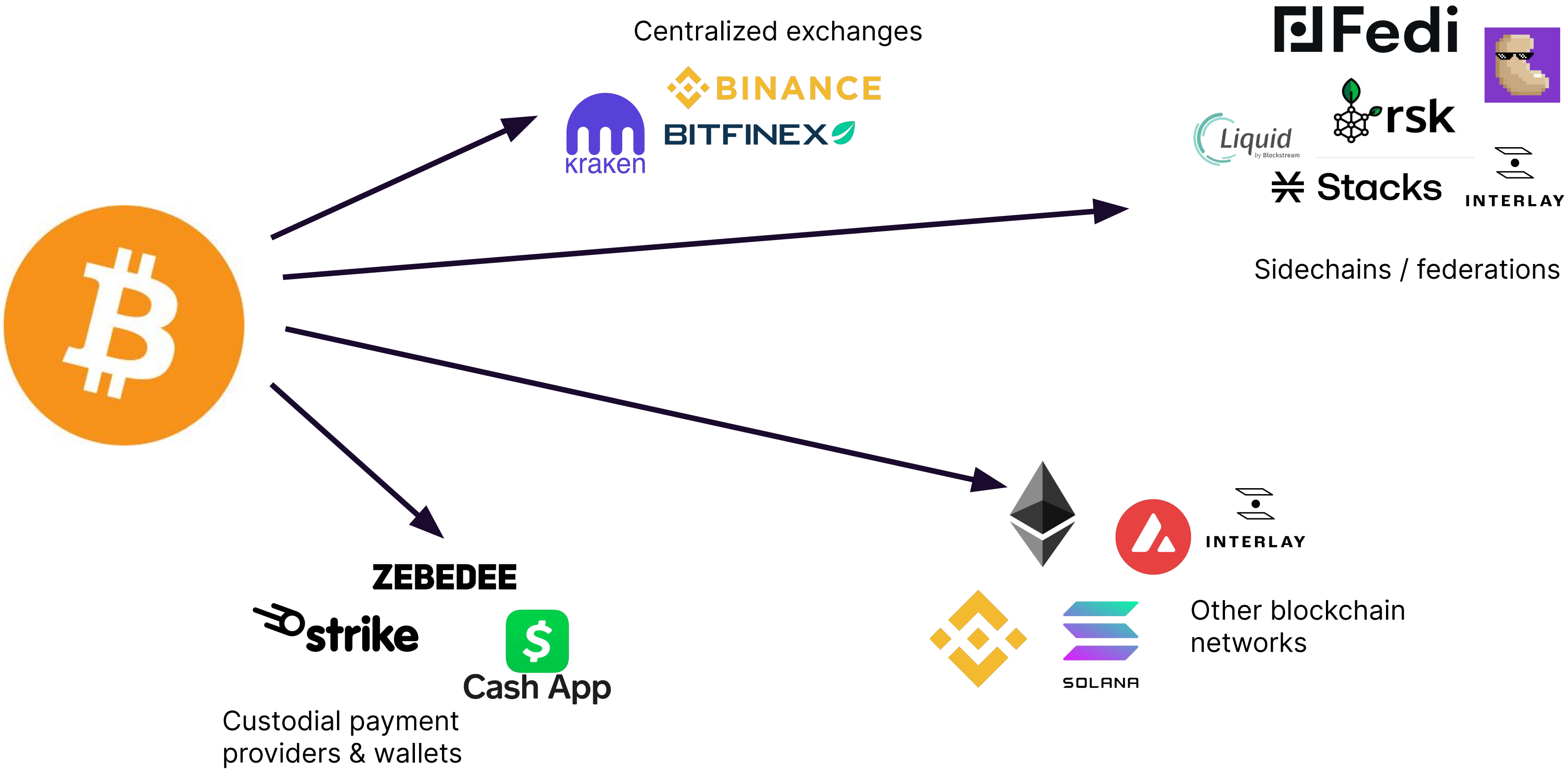
Alexei Zamyatin, Co-Founder @ Interlay
Pizza Day Prague

Agenda

- Wrapping: How to move BTC to other chains?
- Why is bridging so hard?
- How to build a decentralized BTC bridge?

Why should we care about Bitcoin bridges?

To bridge = to deposit



Centralized exchanges

Sidechains / federations

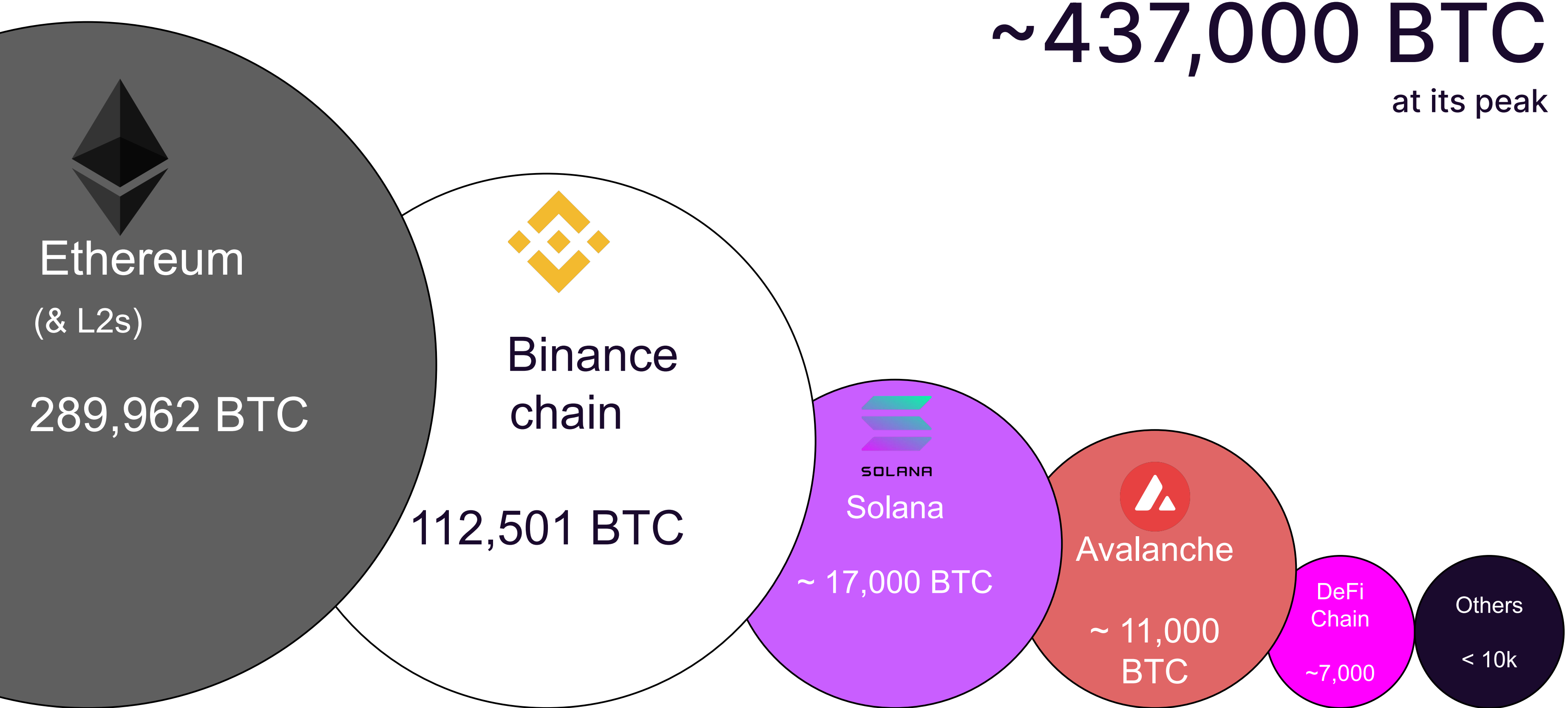
Other blockchain networks

Custodial payment providers & wallets

Bitcoin on other chains

~437,000 BTC

at its peak



Ethereum

(& L2s)

289,962 BTC

Binance chain

112,501 BTC

SOLANA

Solana

~ 17,000 BTC

Avalanche

~ 11,000 BTC

DeFi Chain

~7,000

Others

< 10k

How much is decentralized?

How much is decentralized?

< 0.3 %

Bitcoin Bridges 101

Goals

What?

1. **Deposit** BTC into an appchain (“application chain”)
2. **Use** BTC like a native asset on the appchain
3. **Withdraw** BTC back to Bitcoin

How?

UX → Same as using BTC on a centralized exchange

Security → Always be able to get my BTC back

Reminder: Trust Models

	On Bitcoin	**BTC on other chains**
What do I need?	Bitcoin wallet	Bitcoin wallet Wallet on other chain; A way to bridge BTC
What do I trust?	Bitcoin network is secure; Wallet not corrupted;	Bitcoin network is secure; Other network is secure; Wallets not corrupted; Bridge is not corrupted (might be centralized).
How can I check?	Open source code	Open source code; Reputation of bridge if centralized.

Wrapping

BTC only exists on Bitcoin.

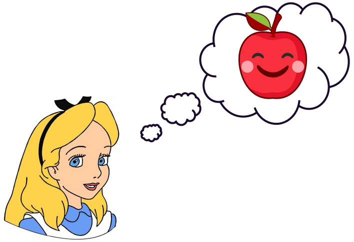
Wrapping = creating a 1:1 representation of BTC on another chain, i.e., as a native token.

In computer science terms:

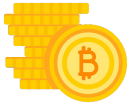
*“Obtain a **write lock** on the state of a UTXO and ensure **updates** made on the other chain are **applied** before the write lock is released”*

Why is bridging so difficult?

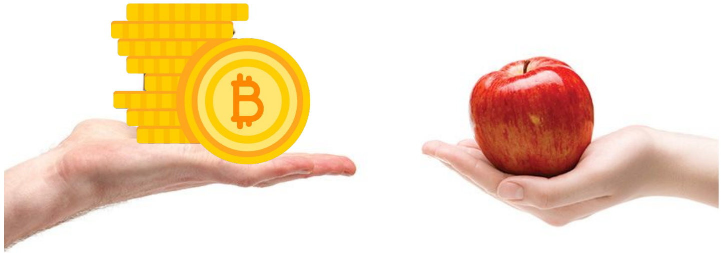
The good old Fair Exchange problem



Alice

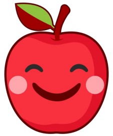


BTC



How to make sure the exchange is always fair?

Bob



Apple

Alice

(In the digital world) someone **must make the first move.**

To ensure fairness in 100% of cases:

Needs a Trusted Third Party

Bob

BTC

Apple

Formal proof (computer science), 1999

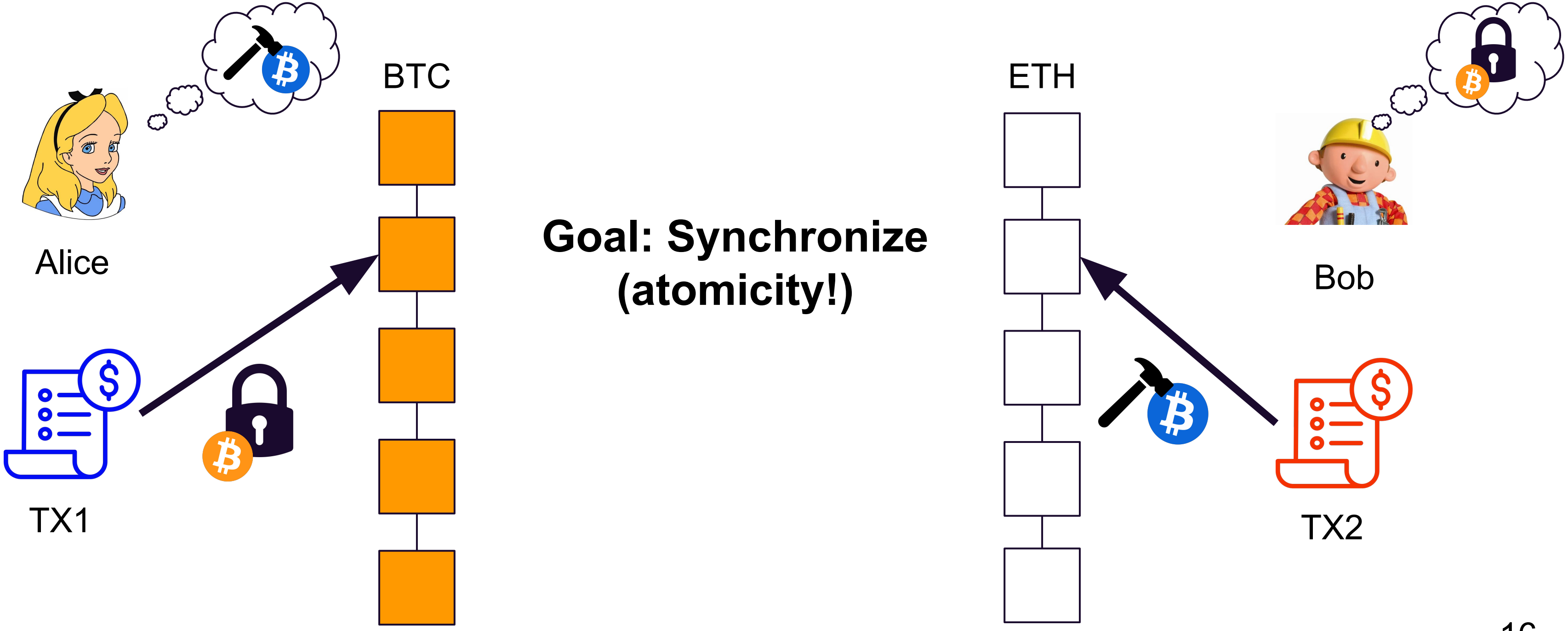
How does this relate to bridges??

Wrapping = swapping BTC for wrapped BTC

Unwrapping = swapping wrapped BTC for BTC

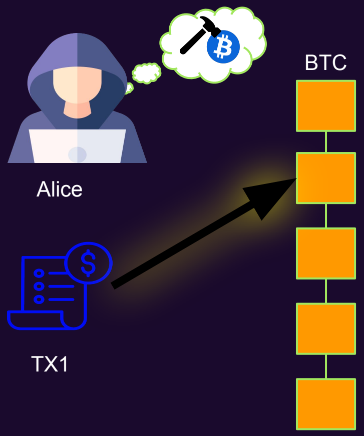
→ **Someone** needs to do the locking and unlocking of BTC on Bitcoin

The Bridge Problem



Goal: Synchronize (atomicity!)

Challenge of Bridging = Selecting a suitable custodian

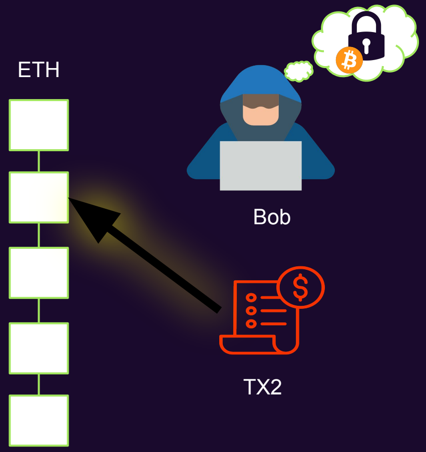


Centralized entity

Committee/Federation

Consensus of 3rd party network

Consensus of involved chains



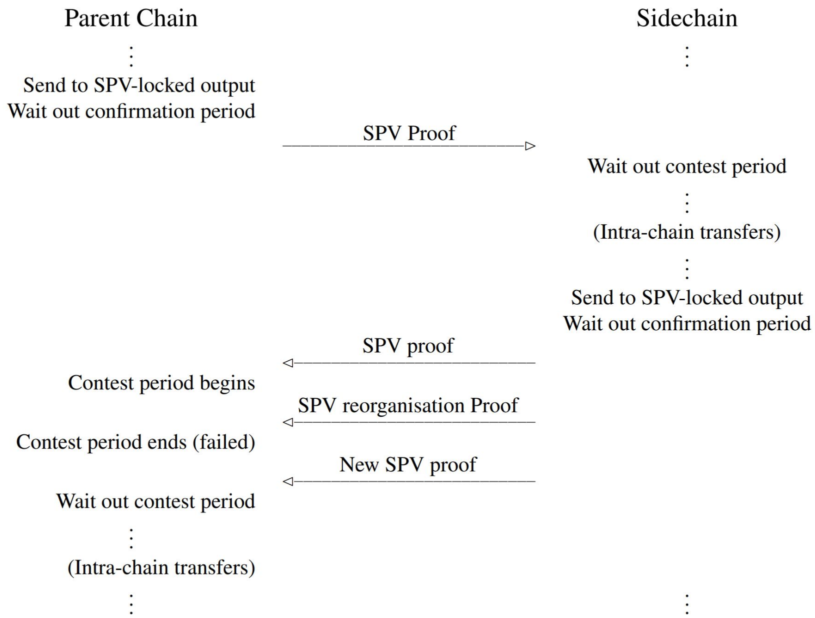
Best Case: Consensus of chains

Inherits the security / decentralization of the target network.

Example:

- Ethereum verifies Bitcoin SPV proofs
- Bitcoin verifies Ethereum SPV proofs
- 1 online party needed to relay proofs

Would need new Bitcoin op-code to verify lock/unlock on other networks/systems



Parent Chain

Sidechain

Send to SPV-locked output
Wait out confirmation period

SPV Proof

Wait out contest period

(Intra-chain transfers)

Send to SPV-locked output
Wait out confirmation period

SPV proof

Contest period begins

SPV reorganisation Proof

Contest period ends (failed)

New SPV proof

Wait out contest period

(Intra-chain transfers)

Figure 1: Example two-way peg protocol.

***2014** Sidechains paper by Back et. al.*

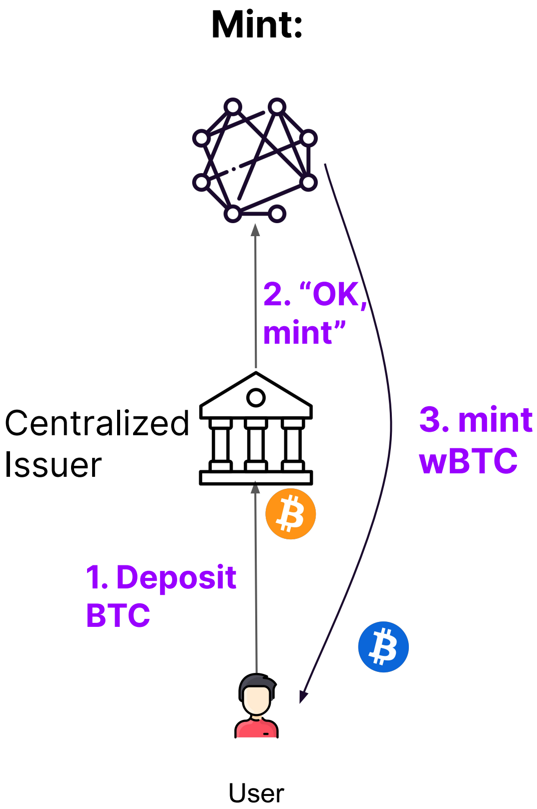
Bridging BTC Today = Hard Mode

Goal: Lock / unlock Bitcoin based on events on other chains.

Problem: **Bitcoin does not know about other chains**

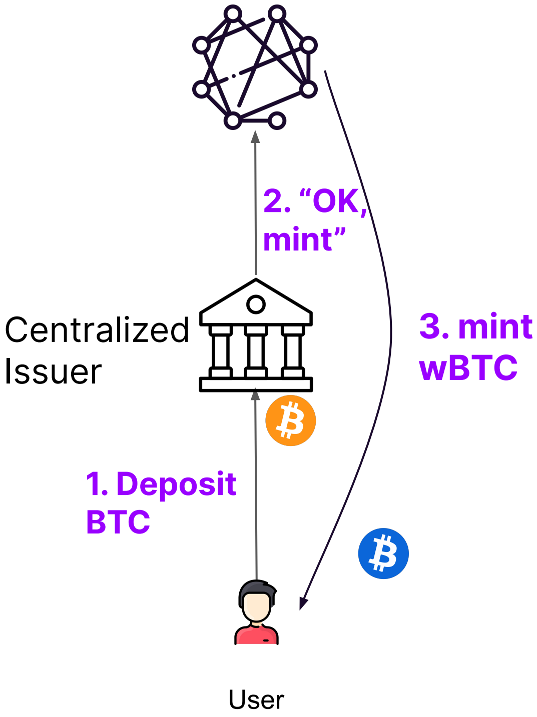
→ **Someone** needs to handle locking/unlocking of BTC

Most (centralized) bridges:



Most (centralized) bridges:

Mint:



2. "OK, mint"

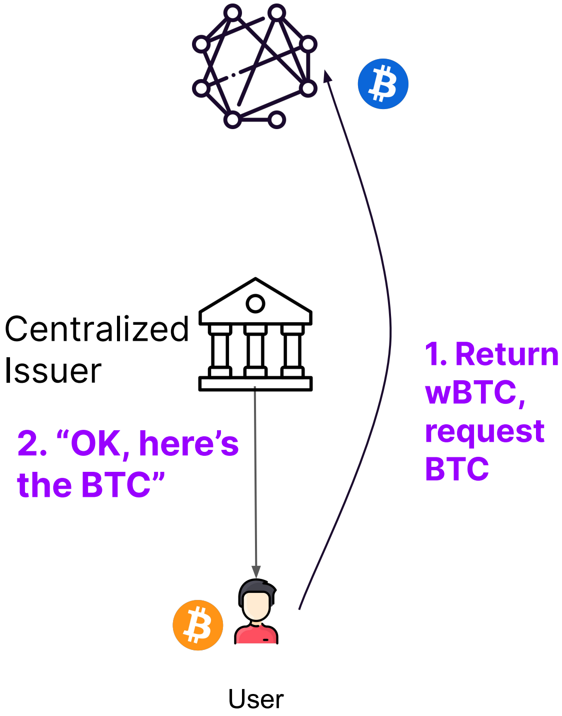
Centralized Issuer

3. mint wBTC

1. Deposit BTC

User

Redeem (success):



Centralized Issuer

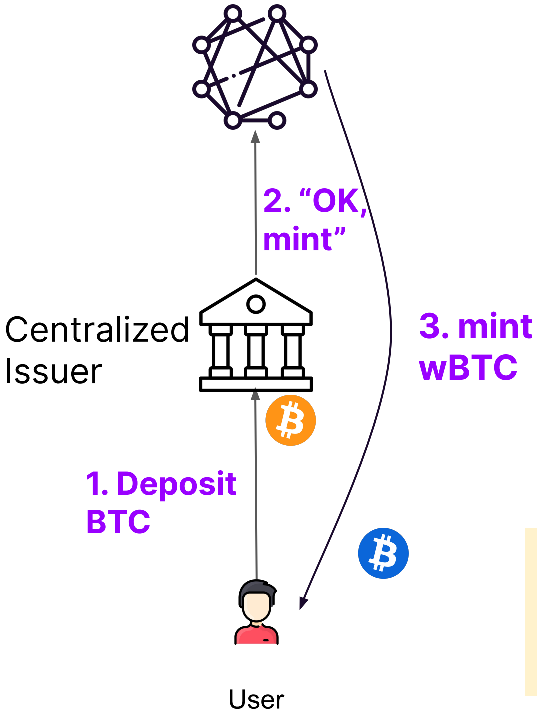
1. Return wBTC, request BTC

2. "OK, here's the BTC"

User

Most (centralized) bridges:

Mint:



Centralized Issuer

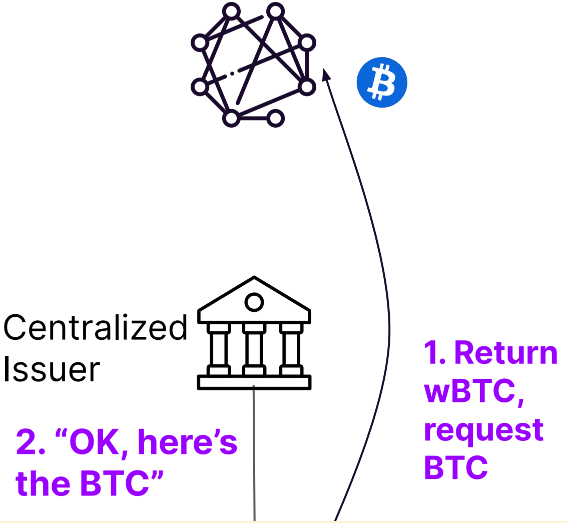
1. Deposit BTC

2. “OK, mint”

3. mint wBTC

User

Redeem (success):

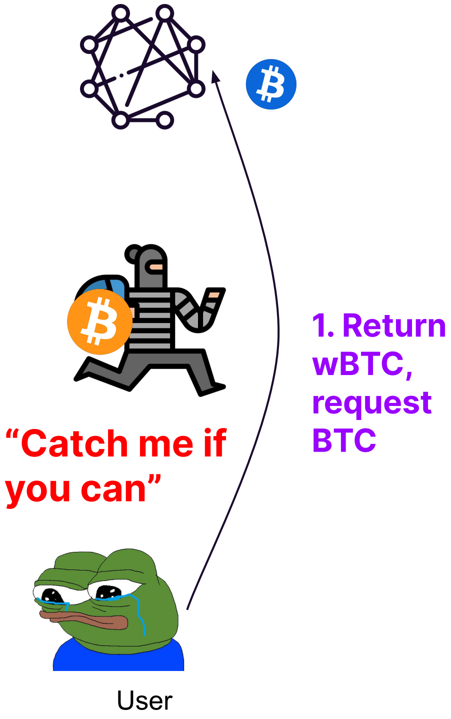


Centralized Issuer

1. Return wBTC, request BTC

2. “OK, here’s the BTC”

Redeem (fail):



1. Return wBTC, request BTC

“Catch me if you can”

User

No protection against theft/seizing/censorship/loss.

How to build a decentralized BTC bridge?

(without changing Bitcoin)

How to build a decentralized bridge?

1) Allow **anyone** to become an operator/custodian

How to build a decentralized bridge?

1) Allow **anyone** to become a operator/custodian

2) Realize this is even worse… now we’re **sending BTC to random people on the internet**

How to build a decentralized bridge?

1) Allow **anyone** to become a operator/custodian

2) Realize this is even worse… now we’re sending BTC to random people on the internet

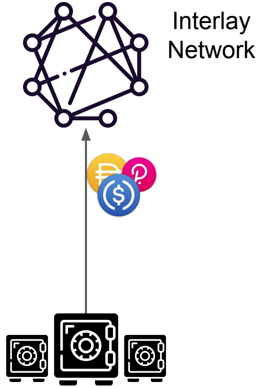
3) Use same tools as Bitcoin to fix:

- **Incentives:** operators lock collateral
- **Punishment:** if operator misbehaves, slash collateral (& reimburse victims)

Example: interBTC

0. Vaults Register

Vaults deposit collateral

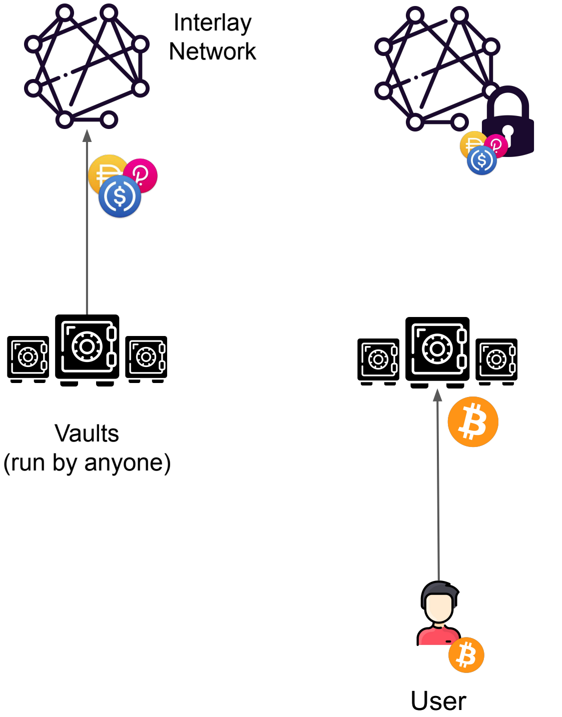


Vaults
(run by anyone)

Example: interBTC

0. Vaults Register
Vaults deposit collateral

1. Lock BTC
User: Lock BTC



Interlay Network

Vaults
(run by anyone)

User

$value(BTC) < value(collateral)$

e.g. 150% collateralization rate for USDT

Example: interBTC

0. Vaults Register

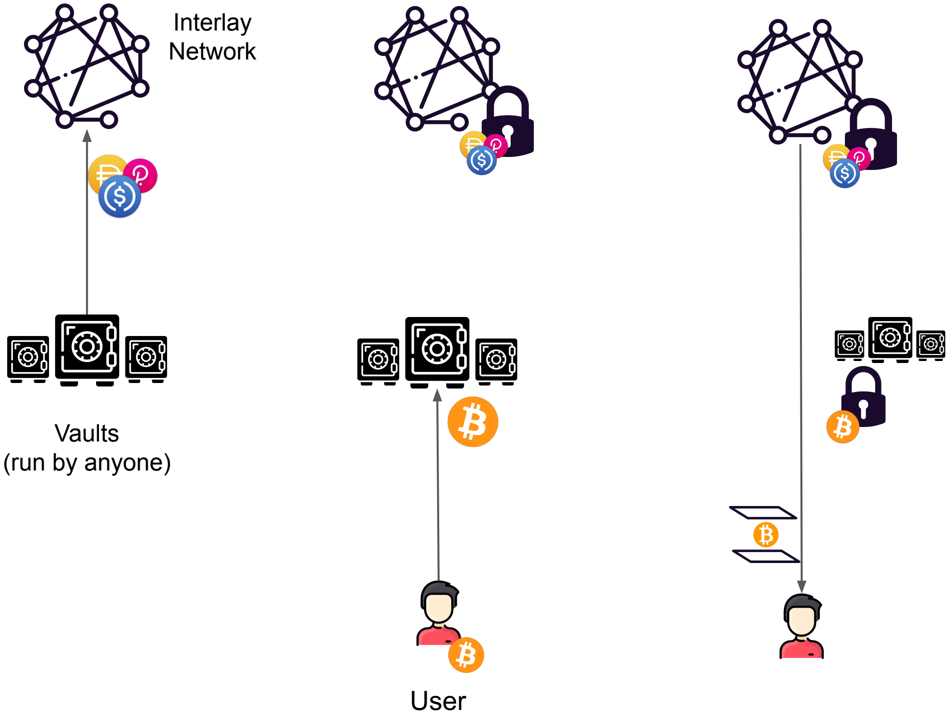
Vaults deposit collateral

1. Lock BTC

User: Lock BTC

2. Mint iBTC

Chain: Mint iBTC to User



Interlay Network

Vaults
(run by anyone)

User

Example: interBTC

0. Vaults Register

Vaults deposit collateral

1. Lock BTC

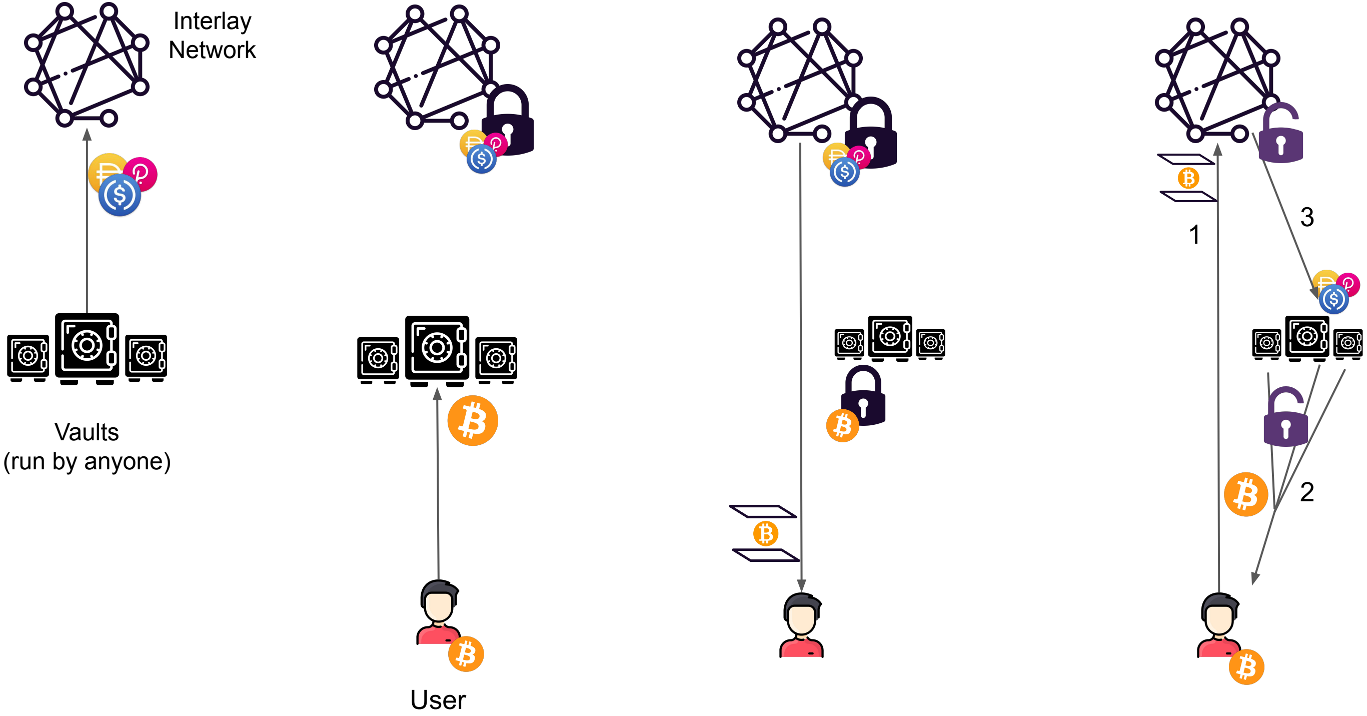
User: Lock BTC

2. Mint iBTC

Chain: Mint iBTC to User

3a. Redeem (Good Vault)

1. User returns iBTC,
2. Vault returns BTC to user,
3. Vault collateral unlocked



Interlay Network

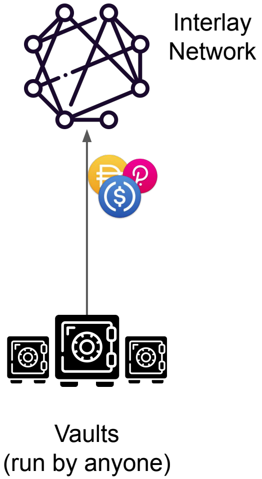
Vaults
(run by anyone)

User

Example: interBTC

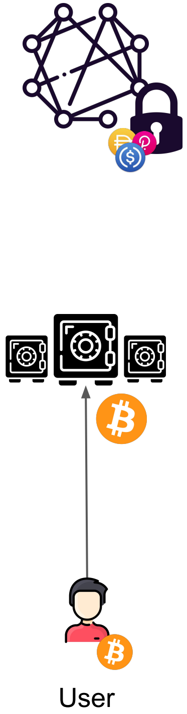
0. Vaults Register

Vaults deposit collateral



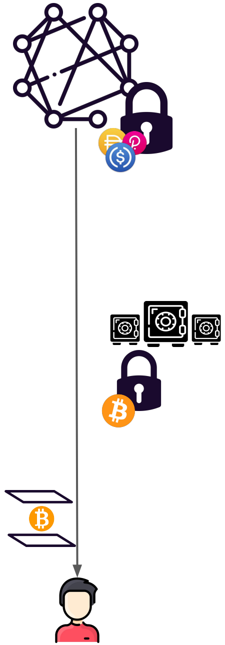
1. Lock BTC

User: Lock BTC



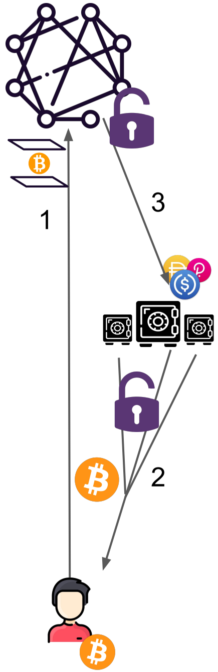
2. Mint iBTC

Chain: Mint iBTC to User



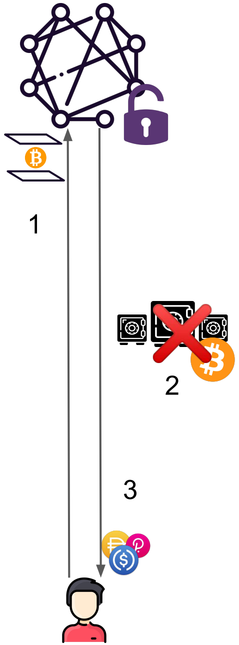
3a. Redeem (Good Vault)

1. User returns iBTC,
2. Vault returns BTC to user,
3. Vault collateral unlocked



3b. Reimburse (Bad Vault)

1. User returns iBTC,
2. Vault fails,
3. User is reimbursed (or tries different Vault)



How to verify BTC payments?

Bitcoin light client (SPV) deployed as a smart contract

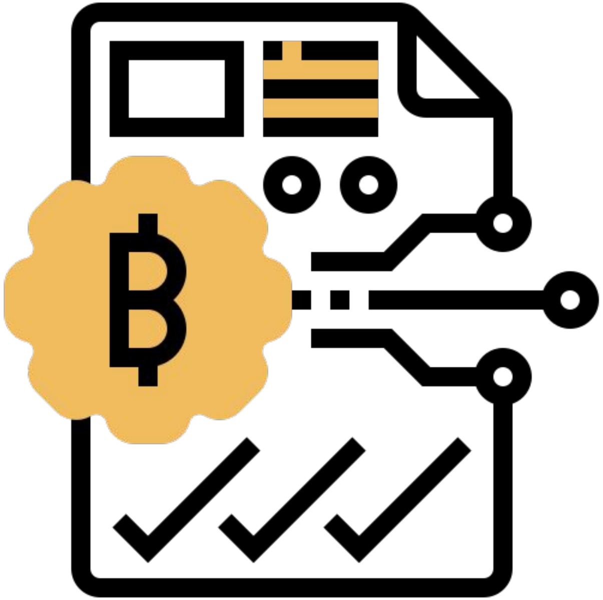
→ **Track** all Bitcoin block headers

→ **Verify** Bitcoin transactions

Security model: if in Bitcoin main chain → must be valid

(same as any mobile wallet)

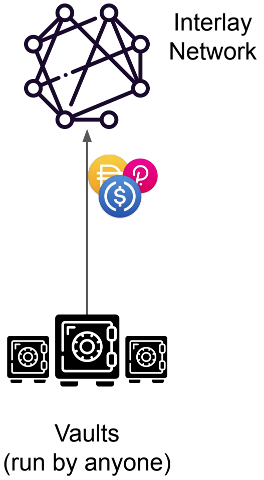
Someone needs to keep the light client up to date



Example: interBTC

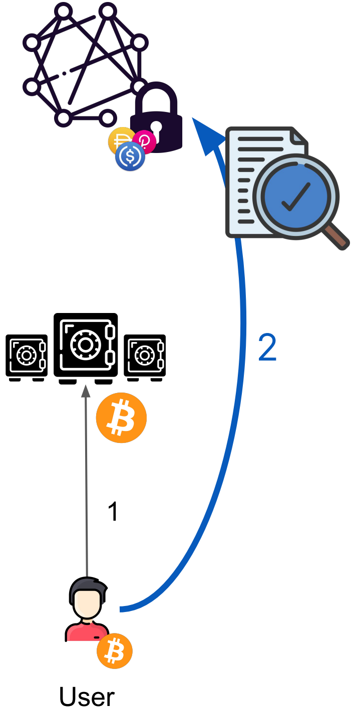
0. Vaults Register

Vaults deposit collateral



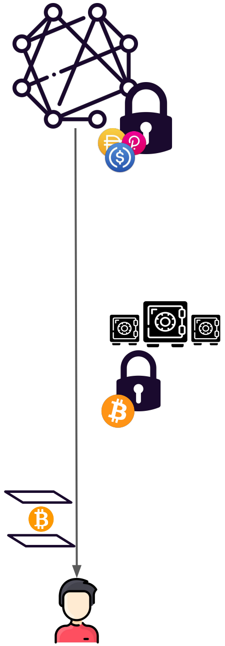
1. Lock BTC

User: Lock BTC



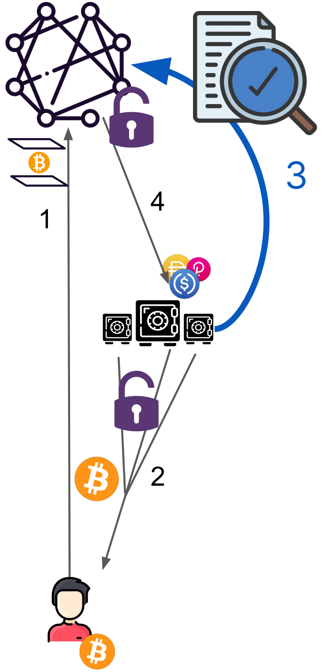
2. Mint iBTC

Chain: Mint iBTC to User



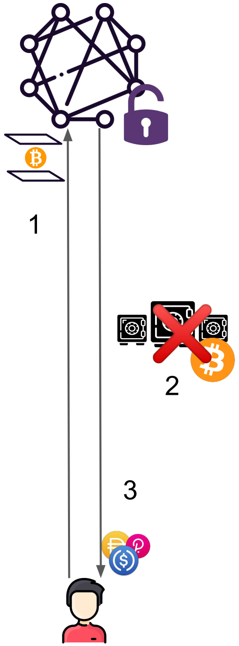
3a. Redeem (Good Vault)

1. User returns iBTC,
2. Vault returns BTC to user,
3. Vault collateral unlocked



3b. Reimburse (Bad Vault)

1. User returns iBTC,
2. Vault fails,
3. User is reimbursed (or tries different Vault)



Summary

1. **Issuer = smart contract**

2. **Permissionless network of Vaults (anyone can join)**

3. **Vaults are over-collateralized (insurance)**

4. **Verification: SPV light client**

Security assumption: I will get my BTC back or will be reimbursed

What we skipped

- **Collateral Management & Liquidations**
  - Vault collateral may decrease in value
  - Top up or or get liquidated → **same as lending protocols**

- **How do we know the price of BTC?**
  - Yes, needs oracles
  - Can be mix of centralized and decentralized exchanges

Extensions / Flavors

- **Vault Models**
  - Single key
  - Vault = multisig (plain, musig, MPC threshold sig,...)
  - Free for all vs pre-defined group vs one big Vault
- **Collateral type vs amount**
  - Full / partial
  - Diversified (USDC, ETH,...) vs native token (risky)
- **Security assumption**
  - Pessimistic / optimistic
- **Verification type**
  - Light client / 3rd party oracle / coinvote

Comparison of some bridges

	BTC custody / Security Model	**Collateral?**
RSK	Multisig by group of 3rd parties	No
Stacks xBTC	Centralized 3rd party custodian	No
Stacks sBTC	Multisigs of STX stakers, rotating	Yes, but STX token
tBTC v2	Big (50/100), rotating multisig of 3rd parties	No
Liquid	Multisig by sidechain operators (federated system)	No
Fedimint	Multisig of operators of the mint (federated system)	No
Cashu	Single key, operator of the mint (custodial system)	No
Interlay iBTC	Decentralized network of collateralized 3rd party custodians	Yes (multi-collateral)

Other Bridge Models

Miner-enforced bridges

What?
Bitcoin miners verify bridges, ensuring lock/unlock handled correctly.

How?
For example: BIP300

- BIP300: miners vote on peg-in & peg-out transactions over (long) periods of time

But?
→ Needs a fork.

Check out BIPs or layertwolabs.com for more details

ZK Roll-ups

What?

Bitcoin verifies if lock/unlock was correct on the other side by checking a cryptographic proof.

How?

Encode verification of state of another chain as an op-code. Verification of ZK proofs is very efficient (creating them is expensive).

But?

→Needs a fork

→Needs zk technology to mature

Read more on https://bitcoinrollups.org/

Conclusion

Conclusion

Bridge problem = Problem of secure custody

Cure:

- More use cases & adoption of BTC without security risk to Bitcoin
- Objectively more secure than centralized exchanges (if using secure bridge)

Curse:

- 99% of BTC bridges are centralized & wrongly marketed
- Decentralization is hard and comes at a cost (capital efficiency)
- Fees accrued on other chains, not Bitcoin

Thanks!

Feel to reach out at:

Twitter: @alexeiZamyatin

Nostr (new):

Check out what we are doing at Interlay:

Twitter: @interlayHQ
Website: interlay.io
Community: linktr.ee/interlay
More research:

Side note:
There are **no**
non-custodial bridges…

… yet?

Towards Non-Custodial Bridges

Fully non-custodial bridges are not possible.

Possible: application specific setups

Example: lending

- My BTC in multisig with 3rd party
- 3rd party can only get BTC if I default on the loan

How?

- DLCs (discrete log contracts): encode different outcomes based on exchange rate
- Another 3rd party = oracle signs transactions based on outcome (ideally, “blind”)

But?

Trust oracle → “but” that’s the case with most decentralized financial applications