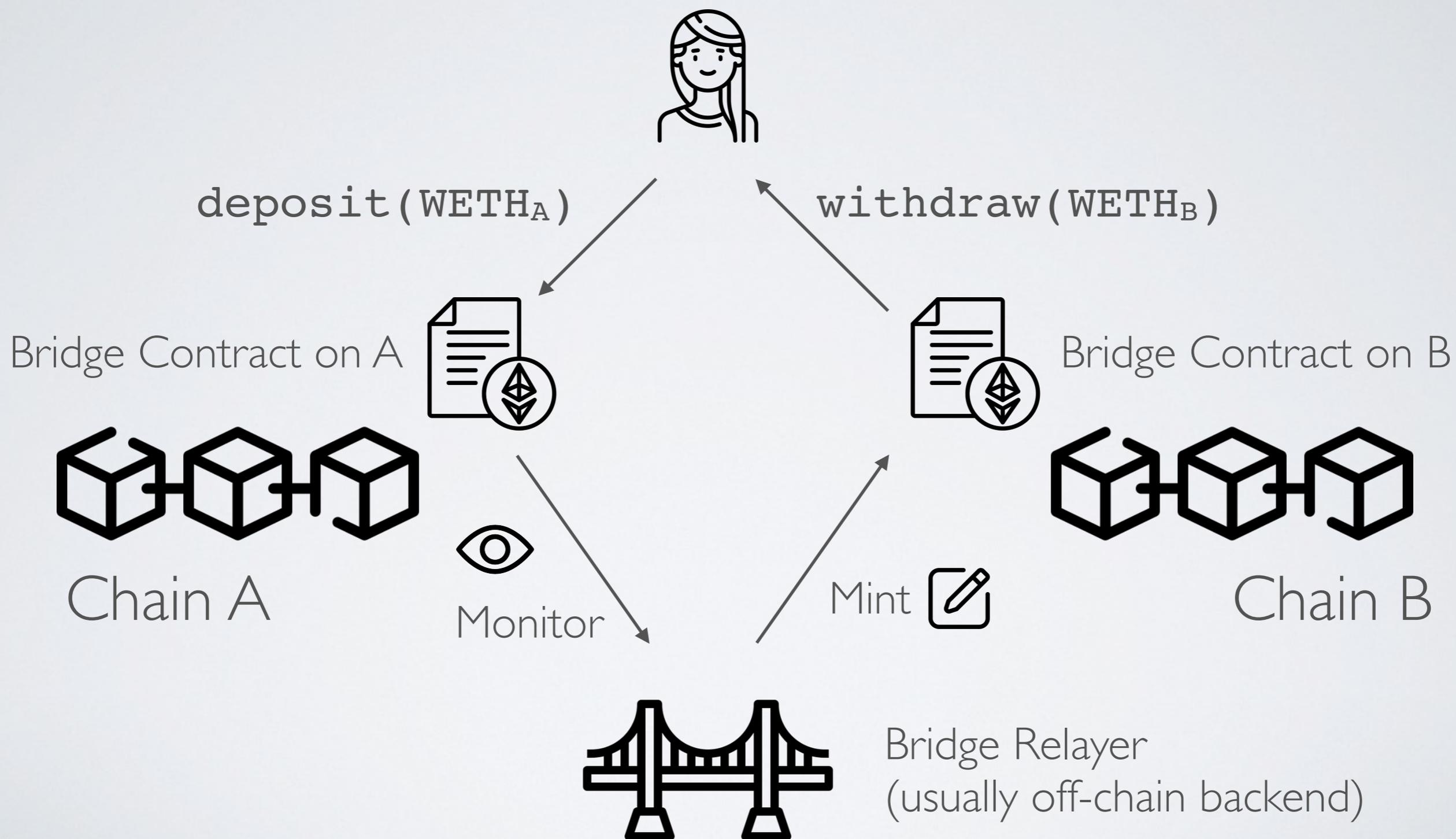


# Bridging and Scaling

Thierry Sans

Bridging

# Transferring assets from one chain to another (and vice versa)



# Scaling

# The problem with blockchain mainnets

## Transaction Speed

- Bitcoin: 7 tx/s
- Ethereum: 15 tx/s

## Transaction Fees

- Bitcoin: 0.000012 BTC (~\$1 USD)
- Ethereum: 0.0002 ETH (~ \$2 USD)

◎ Transaction speed and transaction fees are **not arbitrary**

➔ They are a direct result of deliberate design choices that **prioritize decentralization** and **security**

i.e the cost to ensure that no single entity can dominate the network

# Solutions

- Use a faster consensus  
(hard without compromising with security)
- Split the chains into multiple ones called "shards"  
(work in progress for Ethereum)
- **Rollups** for off-chain or L2-chains

# The concept of Rollup

The idea is to process transactions outside of the main chain

- either off-chain
  - or onto another L2-chain (bridging)  
with a faster/cheaper validation
- ◎ Both are a necessary trade of with security and decentralization

# Basic Blockchain

The blockchain update its state (balances, storage, ...) after each transaction



$TX_{Alice}$

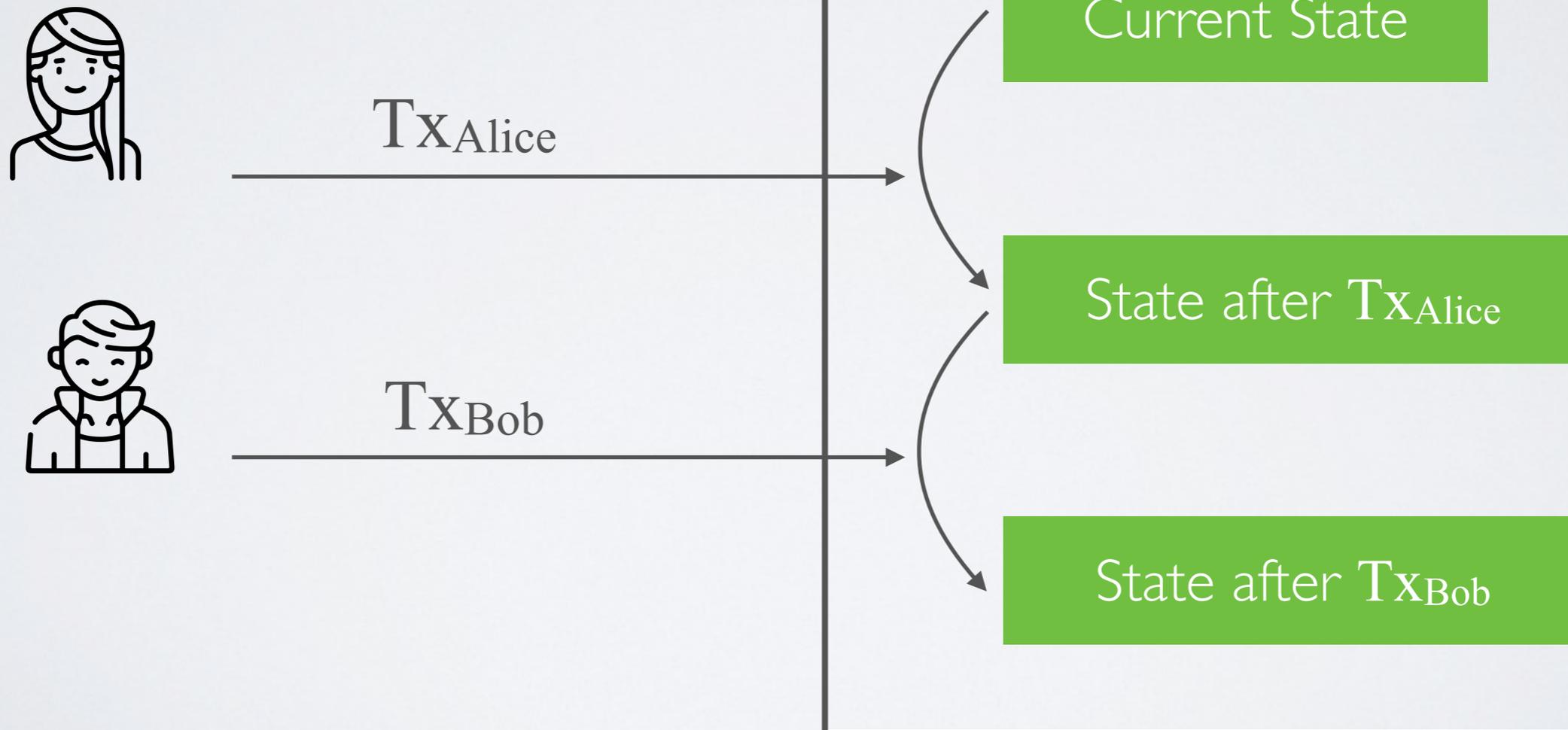


$TX_{Bob}$

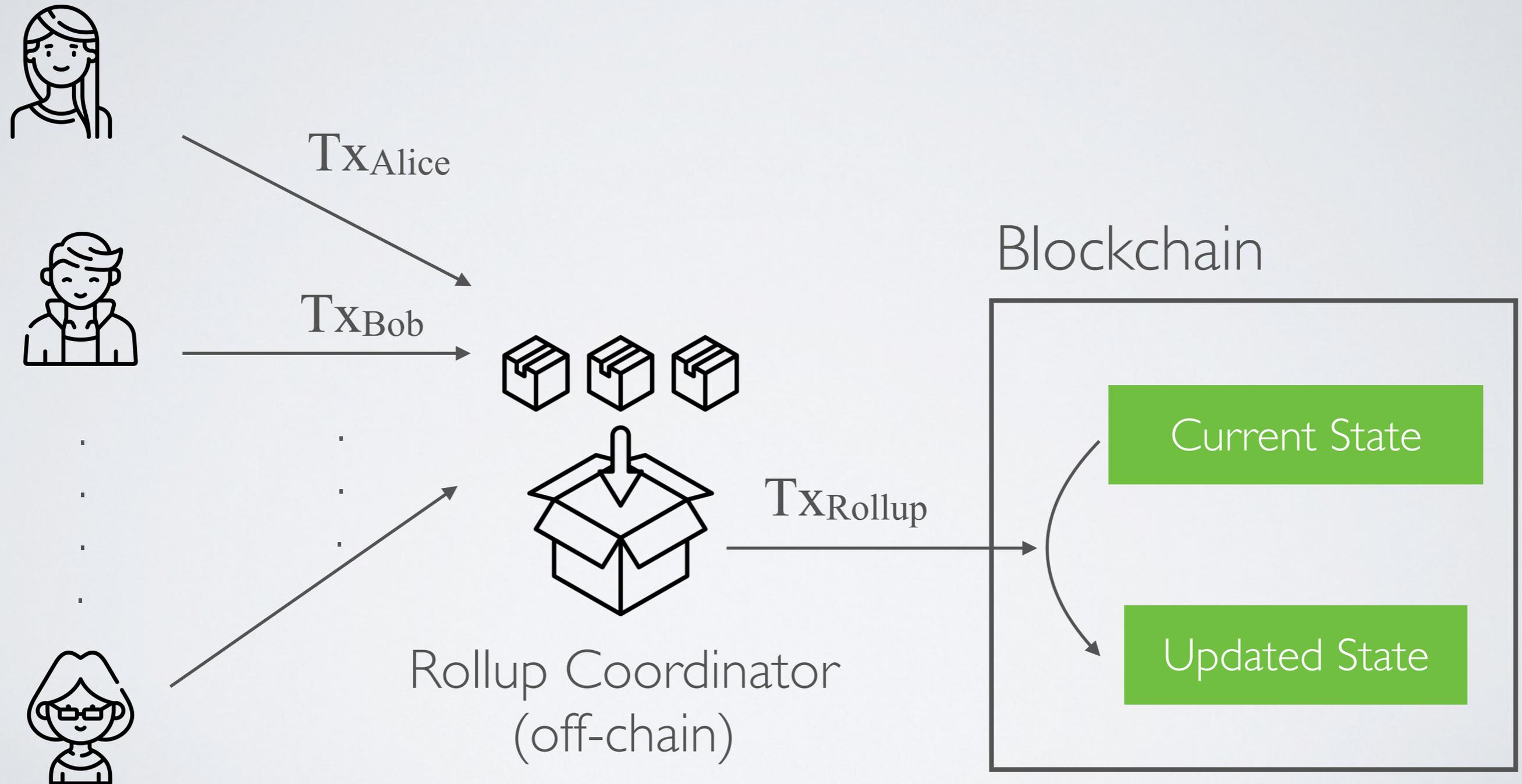
Current State

State after  $TX_{Alice}$

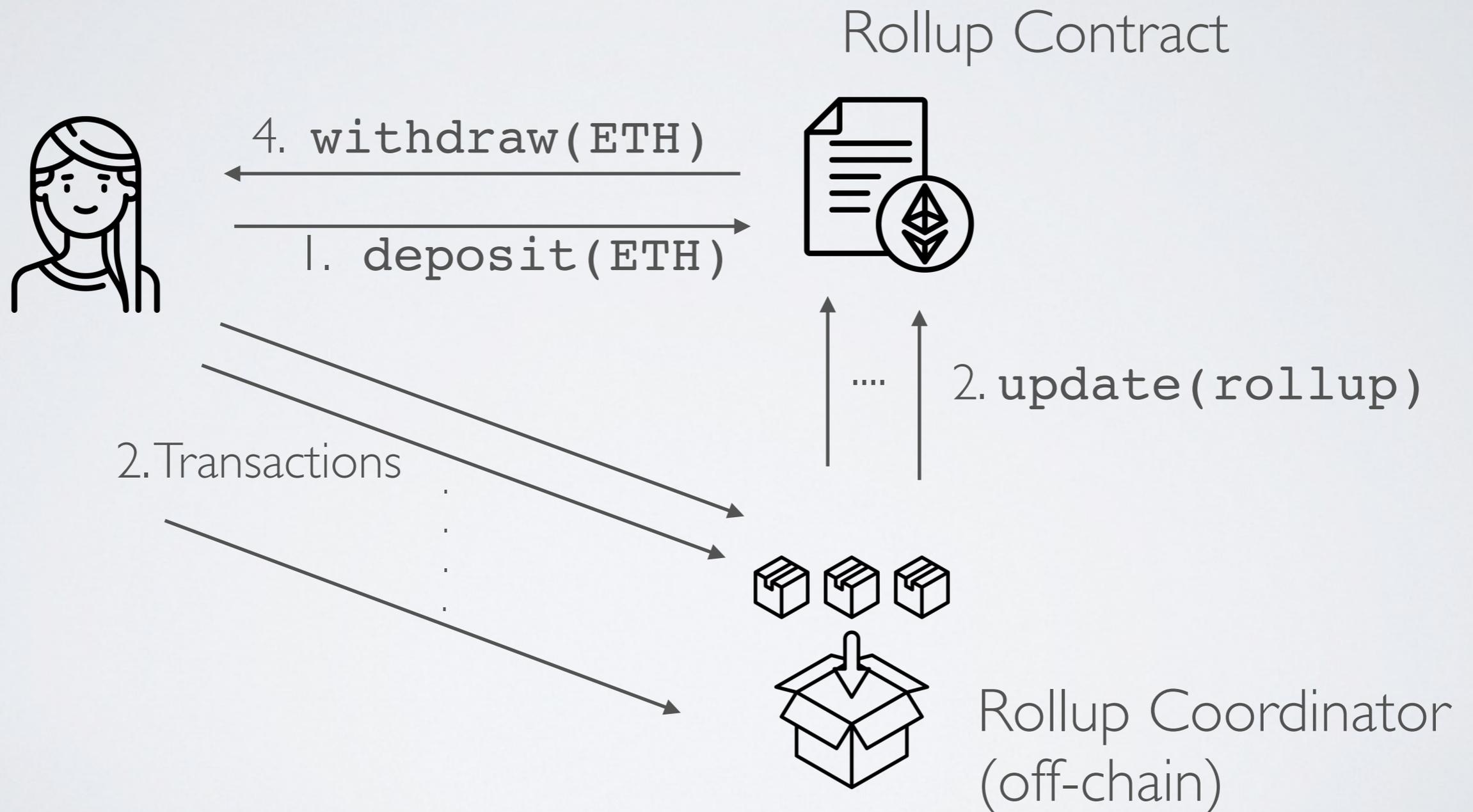
State after  $TX_{Bob}$



# Using a rollup to batch many transactions into one



# Using a rollup contract



# Three types of Rollup

## **Naive Rollup** (not gas efficient)

The rollup is entirely verified on-chain  
(not use in practice because but good to understand the concept of rollup)

## **Optimistic Rollup** (very gas efficient)

The rollup is verified off-chain (after deployment)

## **Zk-Rollup** (pretty gas efficient)

The rollup comes with a ZK-proof that is verified on-chain

# Naive Rollup

➔ The rollup contract keeps track of all users' balances

A rollup is the list of transactions verified on-chain by

- verifying each transaction signature
- checking and updating each user's balances accordingly

⦿ Not use in practice because gas inefficient

# Optimistic Rollup

The rollup contract only stores the root of a Merkle tree with the leafs being the different users balances

- ➔ When users want to withdraw funds, they need to provide a Merkle proof showing that the pair **(address, balance)** is in the corresponding tree

The rollup is

1. the list of transactions verified off-chained
2. and the new tree root computed off-chain

- ➔ The rollup coordinator is trusted to compute the right balances and build the right tree

Once the rollup submitted on-chain, verifiers can check it and challenge it during a dispute period (usually 7 days)

- If the rollup is proved fraudulent, it is rolled back and the stake deposited by the rollup coordinator is slashed

# Zk-Rollup

The rollup contract stores a Merkle root (similar to Optimistic)

The rollup is

1. the list of transactions verified off-chained,
  2. the new tree root computed off-chain
  3. and a ZK-proof proving that given the old tree and the list of transactions, the new tree is correct
- ✓ The Zk-proof is verified on-chain

# Summary on Rollups

## Rollups

- Executes transactions off-chain
  - Posts transaction data on Ethereum
  - Ensures data availability for fraud/validity proofs
- ⦿ Uses **calldata** bytes which are expensive
  - ⦿ Bloats the chain state with data that does not need to be permanent

# EIP-4844 Proto-Danksharding

## Ethereum blob

- ~128 KB binary data chunk
  - Attached to a special transaction type
  - Not accessible to the EVM
  - Stored temporarily (~18 days) and pruned afterward
  - Has separate (and cheaper) blob gas fees
- ✓ Cheaper rollups
  - ✓ Avoid state bloats
  - ✓ Can be used for other applications