Cross-chain & Layer 2 solutions

Prepared by Kirill Sizov

Bridges

Use-cases

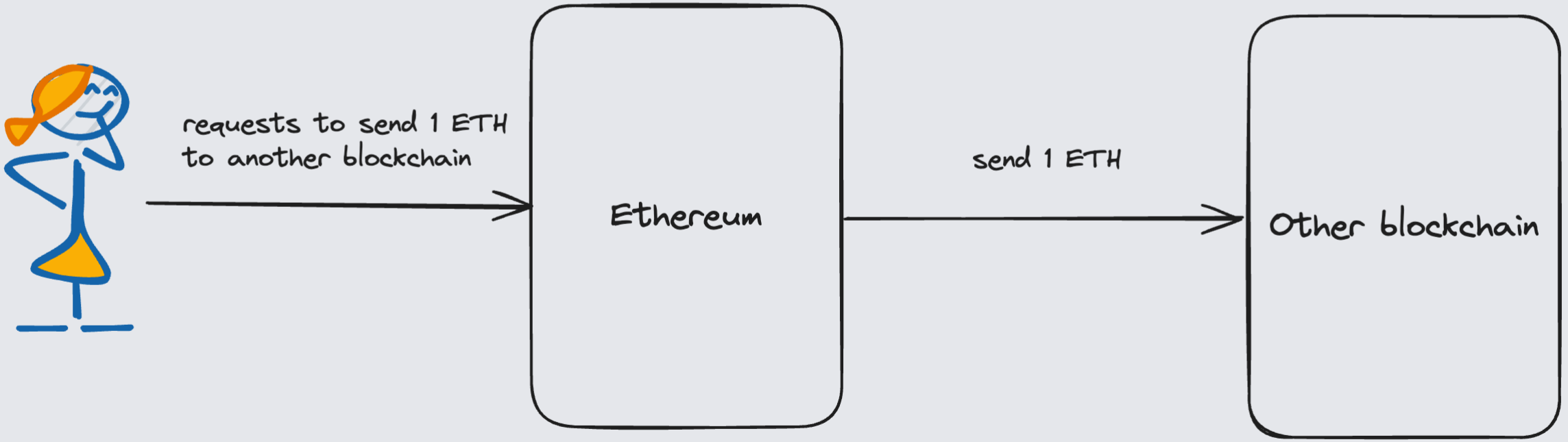
- Swap assets between blockchains
- Transfer assets to another blockchain.
- Communicate data between blockchains.
- Contracts on one blockchain call functions in contracts on another blockchain.

Motivation

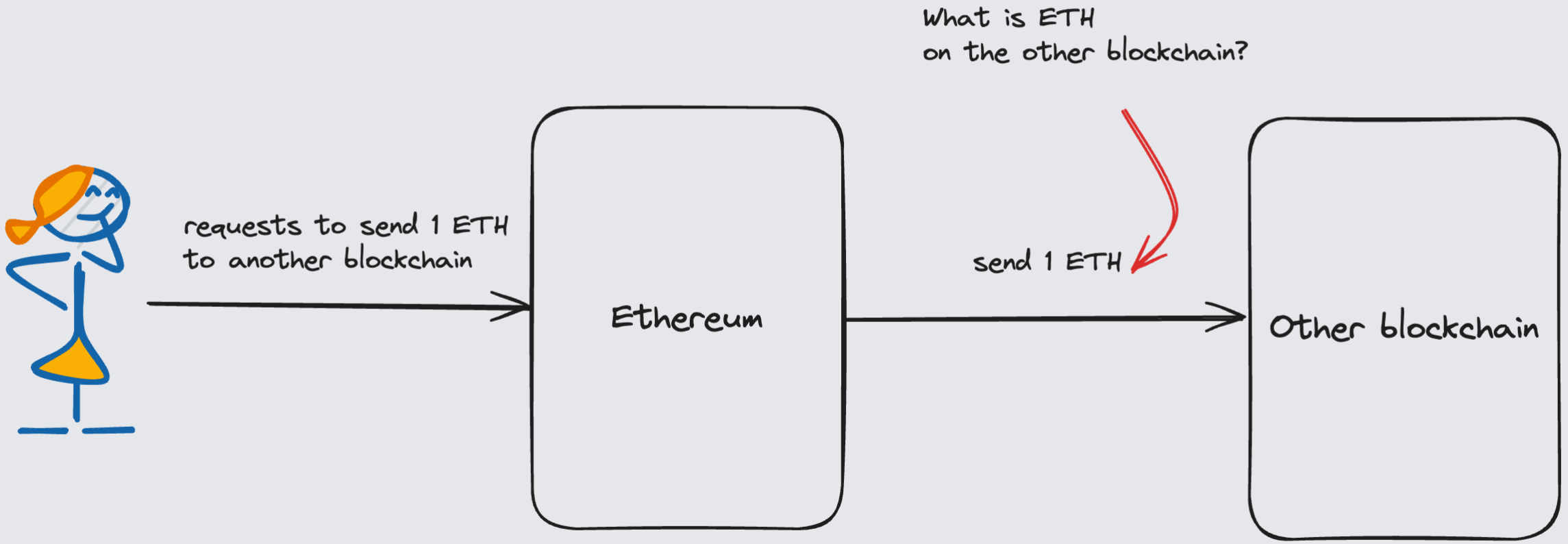
Why would you want to move value from one blockchain to another?

- Transaction fees.
- Block confirmation times.
- Functionality.
- Capital utility.

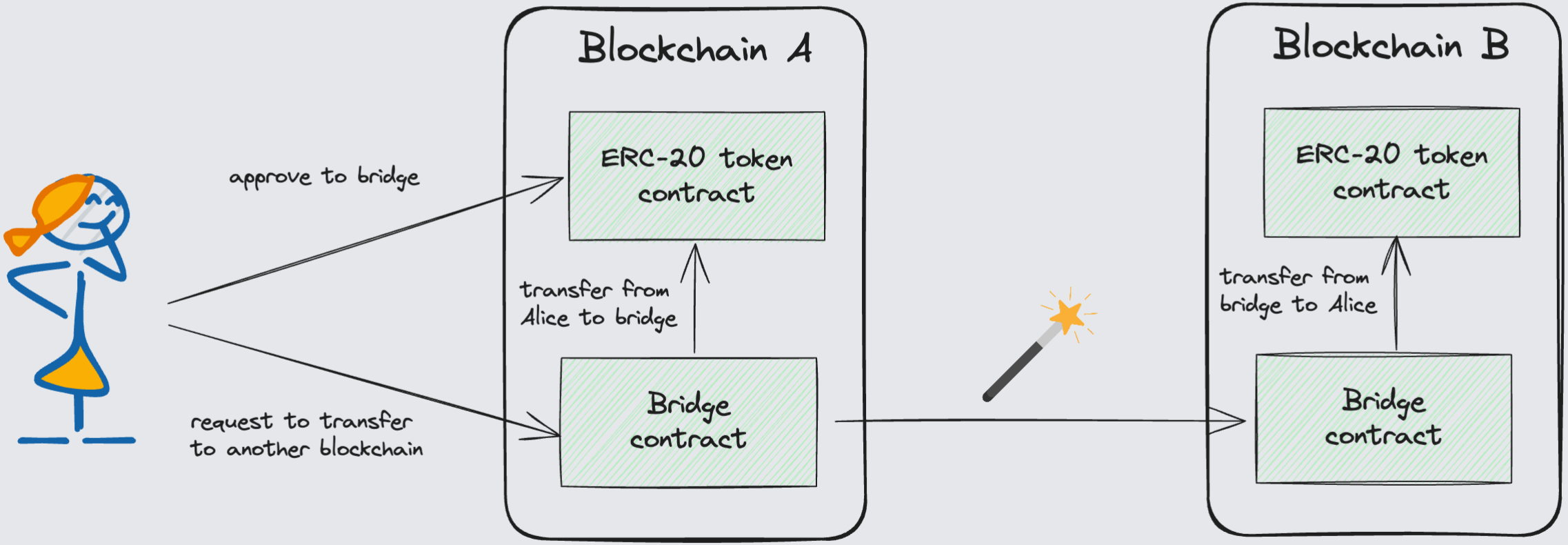
Value transfer



Value transfer



Value transfer



Cross-chain messaging

We need confirmation that some data came from a particular blockchain.

Events

EVM-compatible blockchains can emit `events`:

```
event Deposit(address _from, address _to, uint256 _amount);

function transfer(address _recipient, uint256 _amount) external {

    ...

    emit Deposit(msg.sender, _recipient, _amount);
}
```

Events

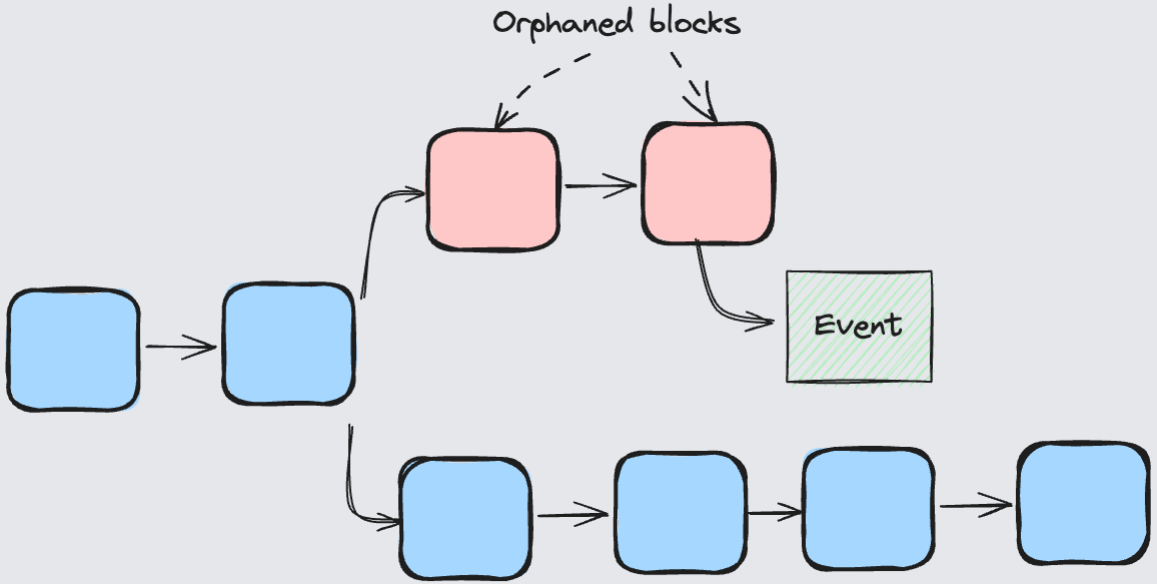
- Events are stored in transaction receipts.
- The merkle tree root of a transaction receipts is included in the block header.

Non-EVM

- However, there are other non-EVM compatible blockchains.
- Most blockchains have some sort of event or message that can be emitted based on programmable logic.

Finality

Cross-chain messaging system should only use events that belong to transactions that have been included in blocks that are (probably) final.



Types

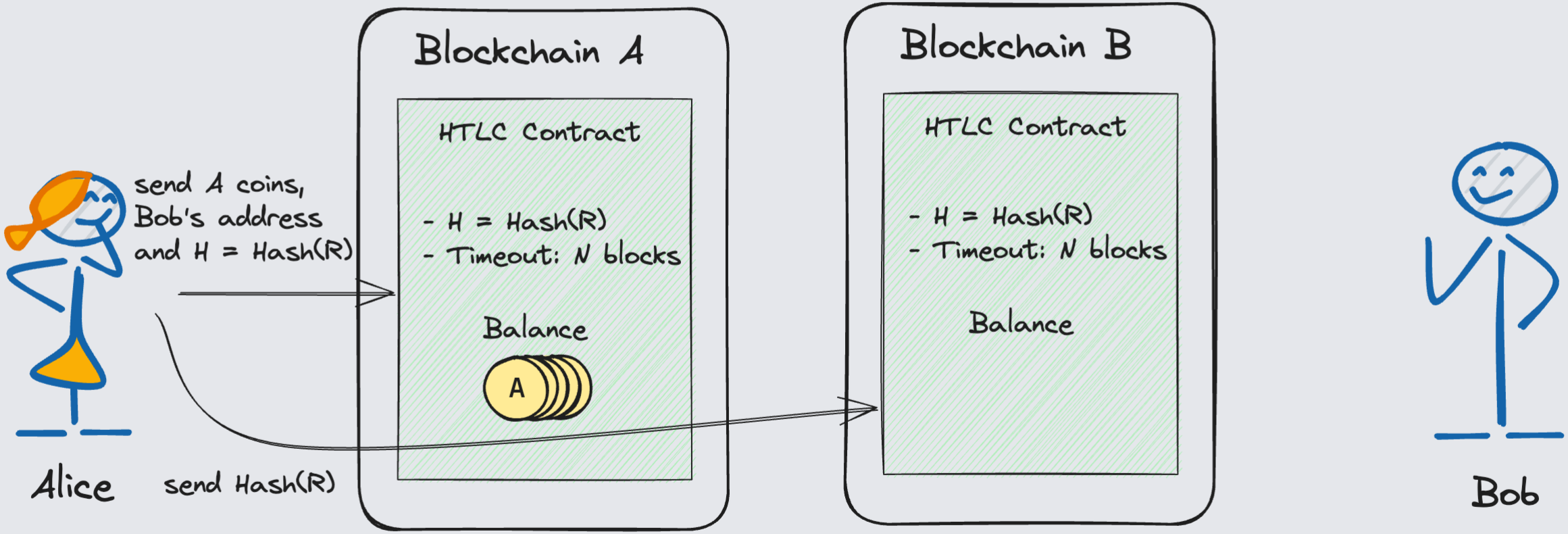
- Trustless: only to blockchain.
- Almost trustless: trust that at least one intermediate party is honest.
- Semi Trusted: trust that at least a threshold number of intermediate party are honest.
- Trusted: trust that a single party is honest.

Trustless / Almost trustless

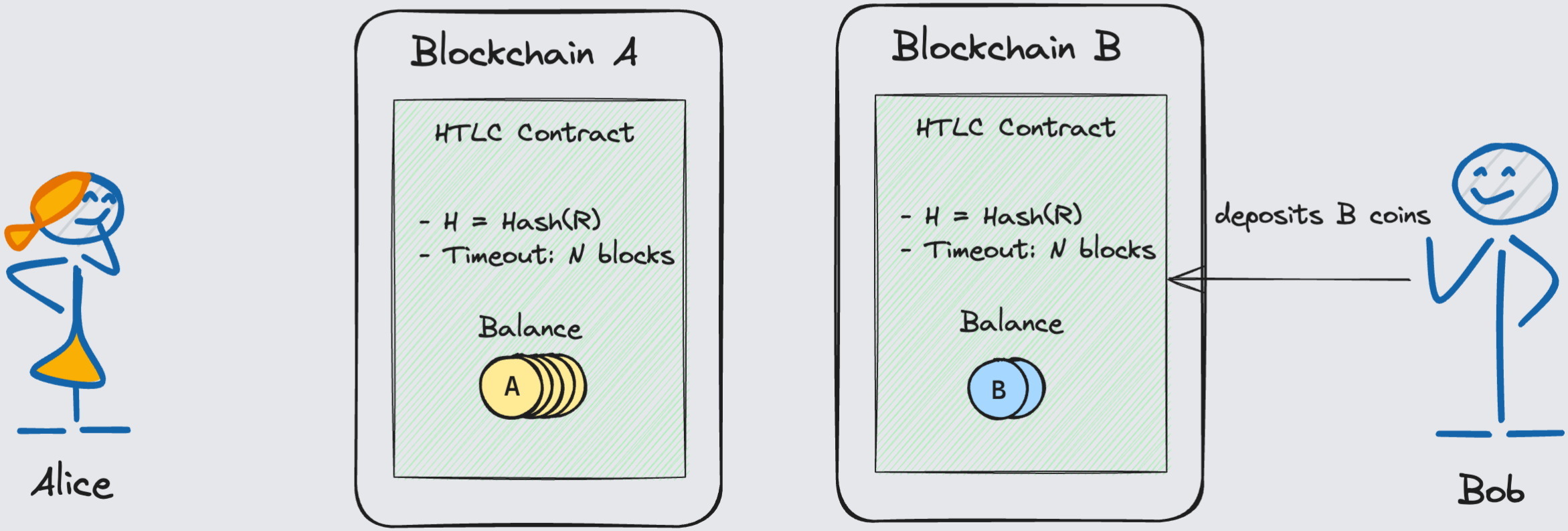
Hash TimeLock Contract (HTLC)

- HTLC rely on cryptographic time-lock functions to ensure that transactions are completed within a specified period or otherwise reverted.

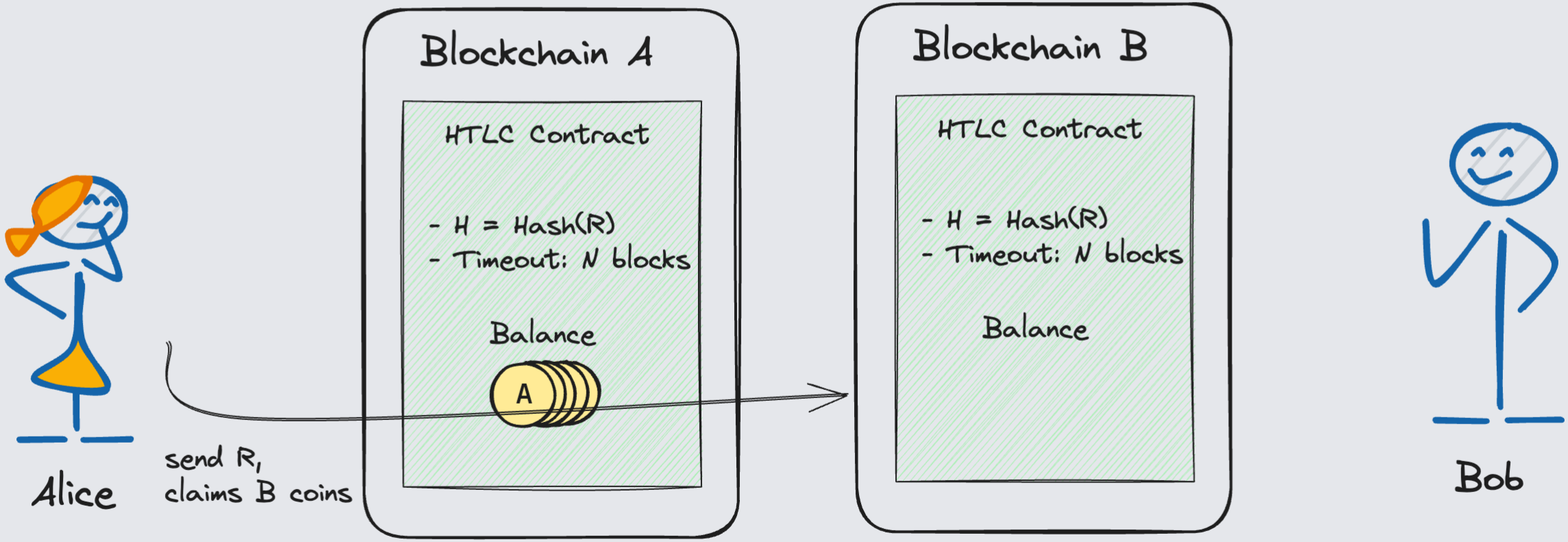
HTLC


Blockchain A
HTLC Contract
- H = Hash(R)
- Timeout: N blocks
Balance
A
Blockchain B
HTLC Contract
- H = Hash(R)
- Timeout: N blocks
Balance
send A coins,
Bob's address
and H = Hash(R)
send Hash(R)
Alice
Bob

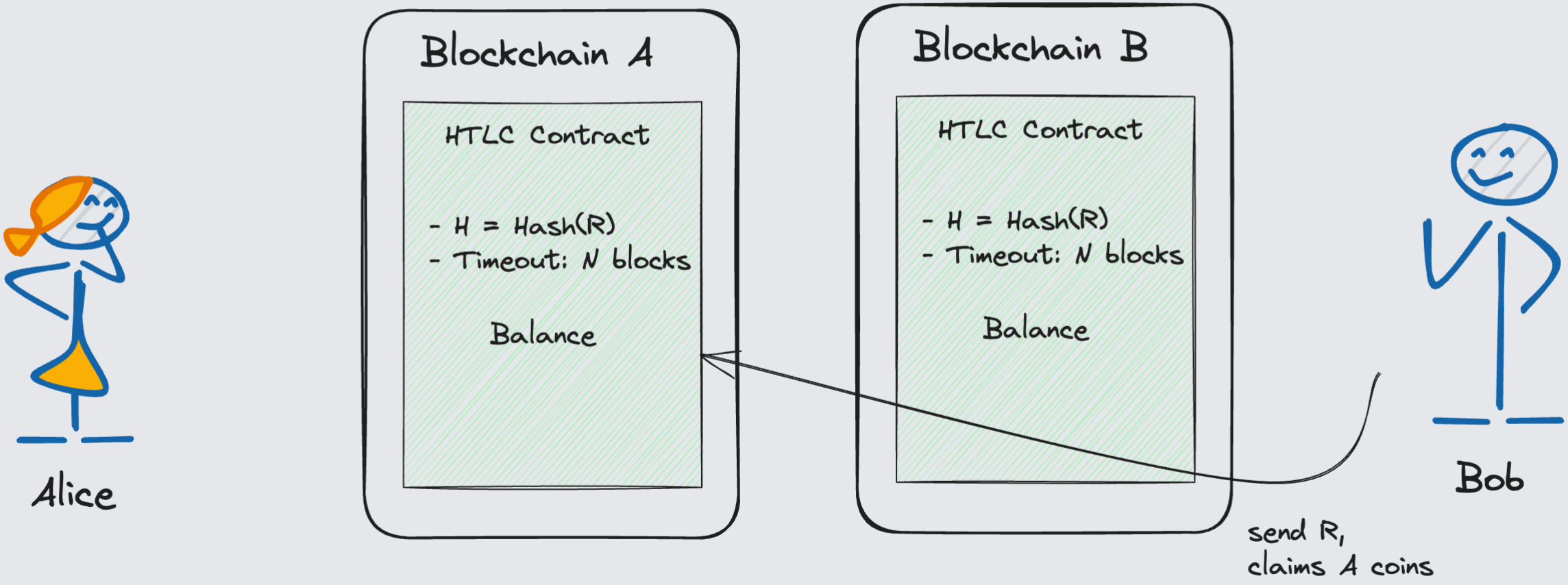
HTLC


Blockchain A
HTLC Contract
- H = Hash(R)
- Timeout: N blocks
Balance
A
Blockchain B
HTLC Contract
- H = Hash(R)
- Timeout: N blocks
Balance
B
deposits B coins
Alice
Bob

HTLC


Blockchain A
HTLC Contract
- H = Hash(R)
- Timeout: N blocks
Balance
A
Blockchain B
HTLC Contract
- H = Hash(R)
- Timeout: N blocks
Balance
Alice
send R,
claims B coins
Bob

HTLC


Blockchain A
HTLC Contract
- H = Hash(R)
- Timeout: N blocks
Balance
Blockchain B
HTLC Contract
- H = Hash(R)
- Timeout: N blocks
Balance
Alice
Bob
send R,
claims A coins

HTLC

Pros

- Trustless.
- Atomic.

Cons

- Limited to value transfer.
- Subject to griefing attacks.
- Possible stealing funds.

HTLC as bridge

- Two transactions on both blockchains.
- Alice has to submit transaction on blockchain B, Bob has to submit transaction on blockchain A.
- Blockchain validators could be bribed to refuse to include transactions in blocks before timeout.

Modified HTLC

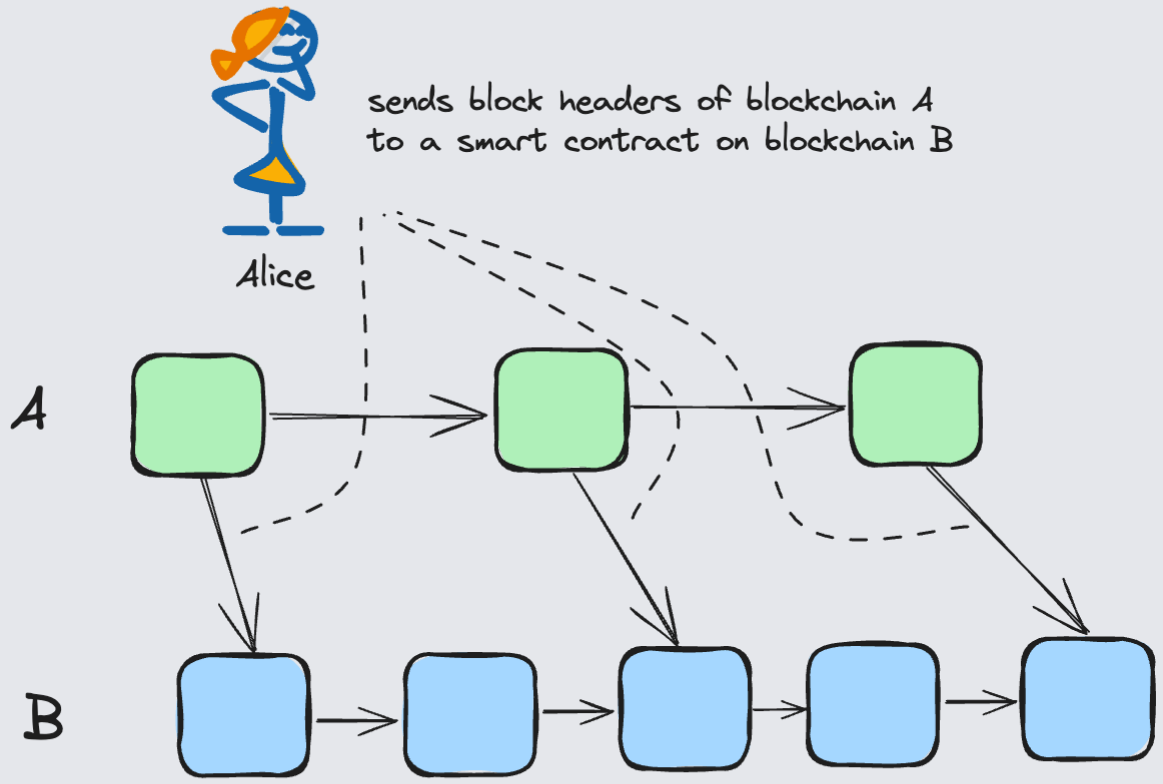
- **Connext** have modified the HTLC protocol such that Alice is a component called a Router.
- The Router does transfer on blockchain A for Bob.

BTC relay

- BTC relay is a bridge between the Bitcoin & Ethereum, though it can be used as general concept how to operate a light client of a PoW blockchain on another blockchain.
- Use the PoW hashing power of the source blockchain to be sure that information can be trusted.

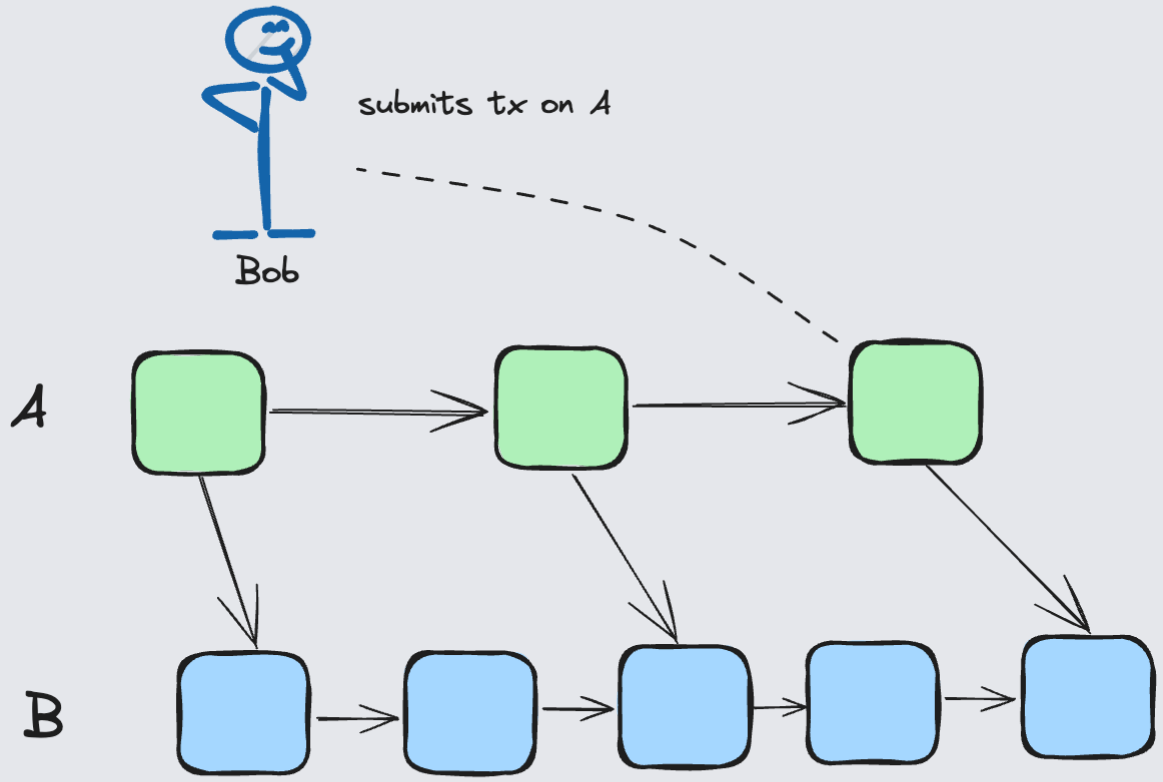
BTC relay

- Relayers compete to transfer block headers to the destination blockchain.
- Block headers are accepted if they match the PoW algorithm.



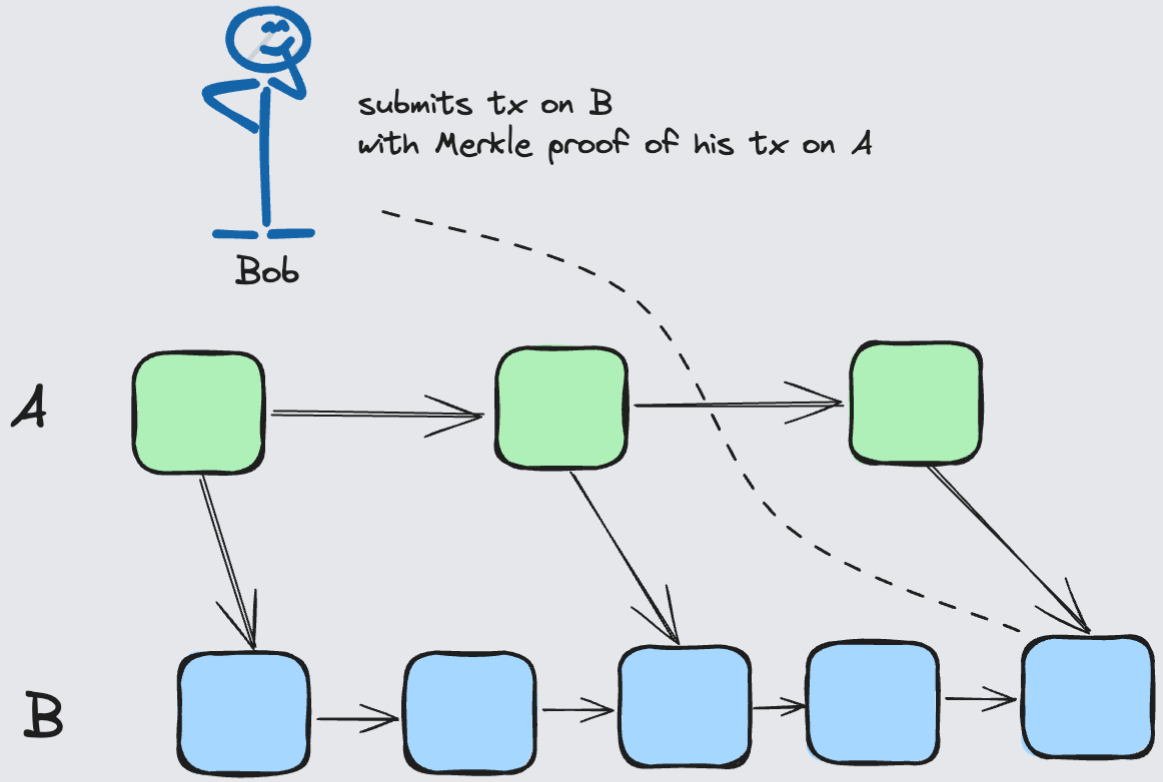
BTC relay

Bob submits a transaction on the source blockchain (Bitcoin).



BTC relay

Bob submits a transaction on the destination blockchain including the transaction from the source chain and a Merkle proof showing the transaction was included in a block on the source blockchain (Bitcoin).



BTC relay

Pros

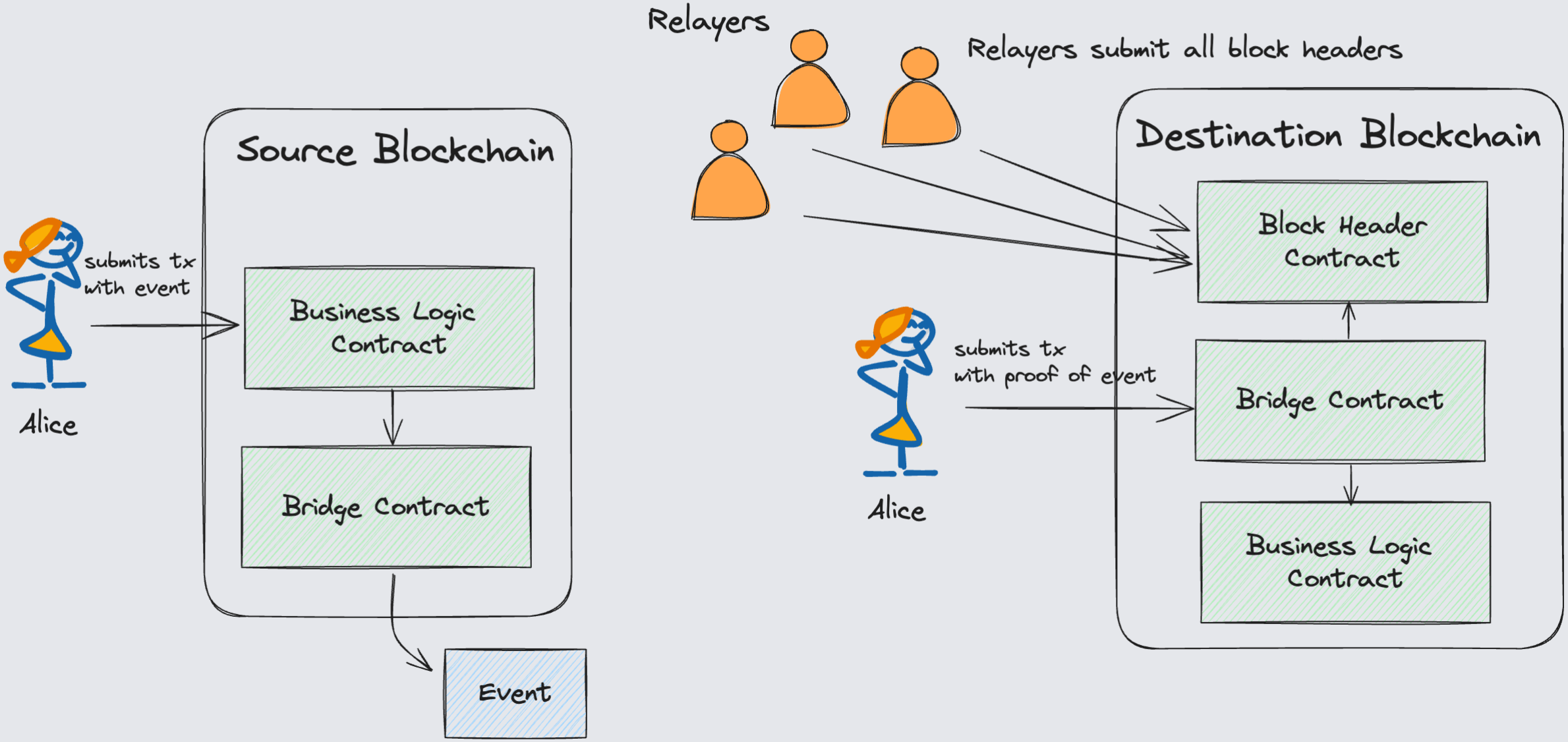
- Only need one honest relayer.

Cons

- All block headers must be transferred.
- Uneconomical if not enough transactions.

Semi Trusted

Block header transfer: separately



Block header transfer: separately

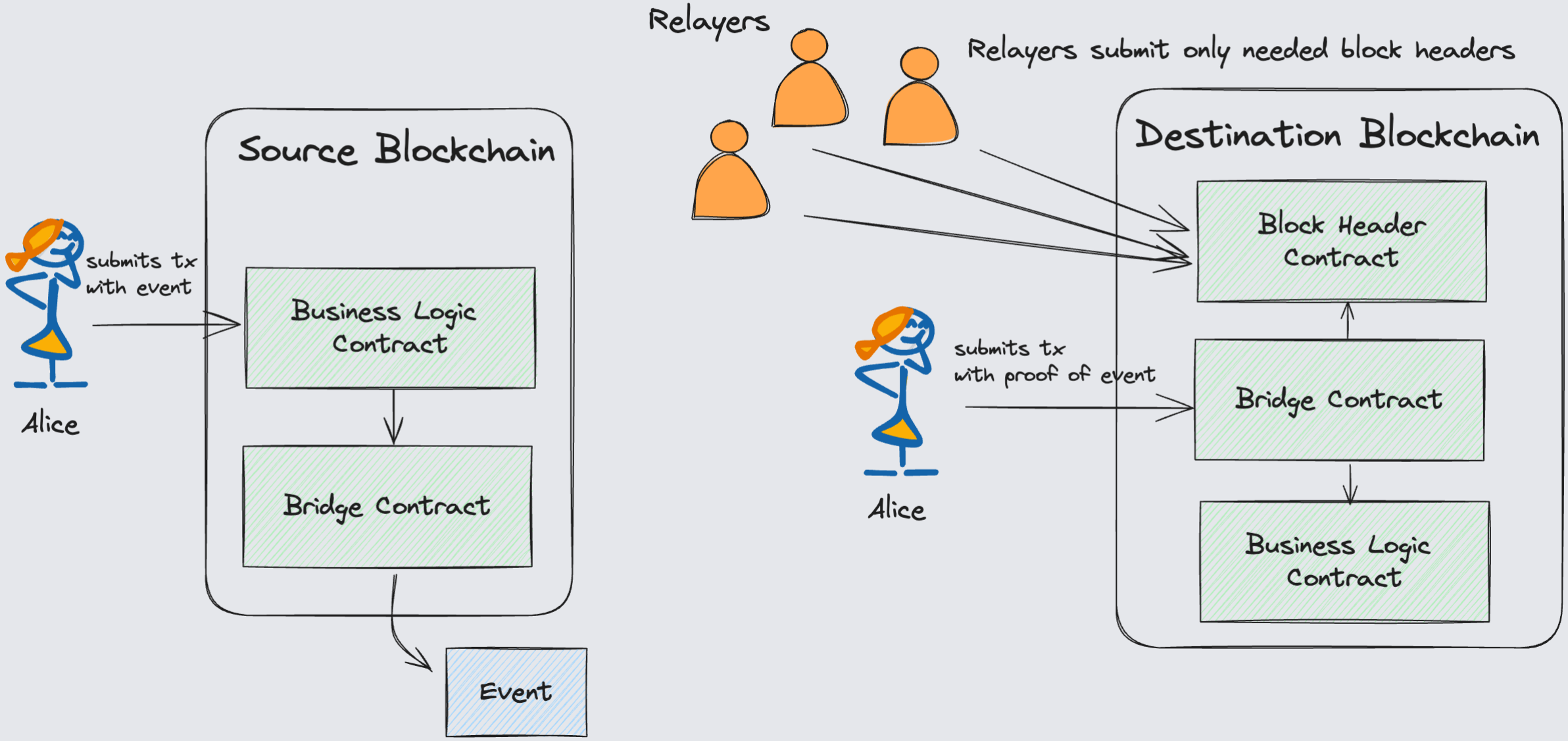
Pros

- Simple.
- Relayers do not need to cooperate.
- Users do not interact with Relayers.

Cons

- One transaction for each relayer, for each block to submit signed block headers.
- Relayers are paying to submit transactions on the destination blockchain.
- User has to be able to submit a transaction on the destination blockchain.

Block header transfer: submit only needed blocks



Block header transfer: submit only needed blocks

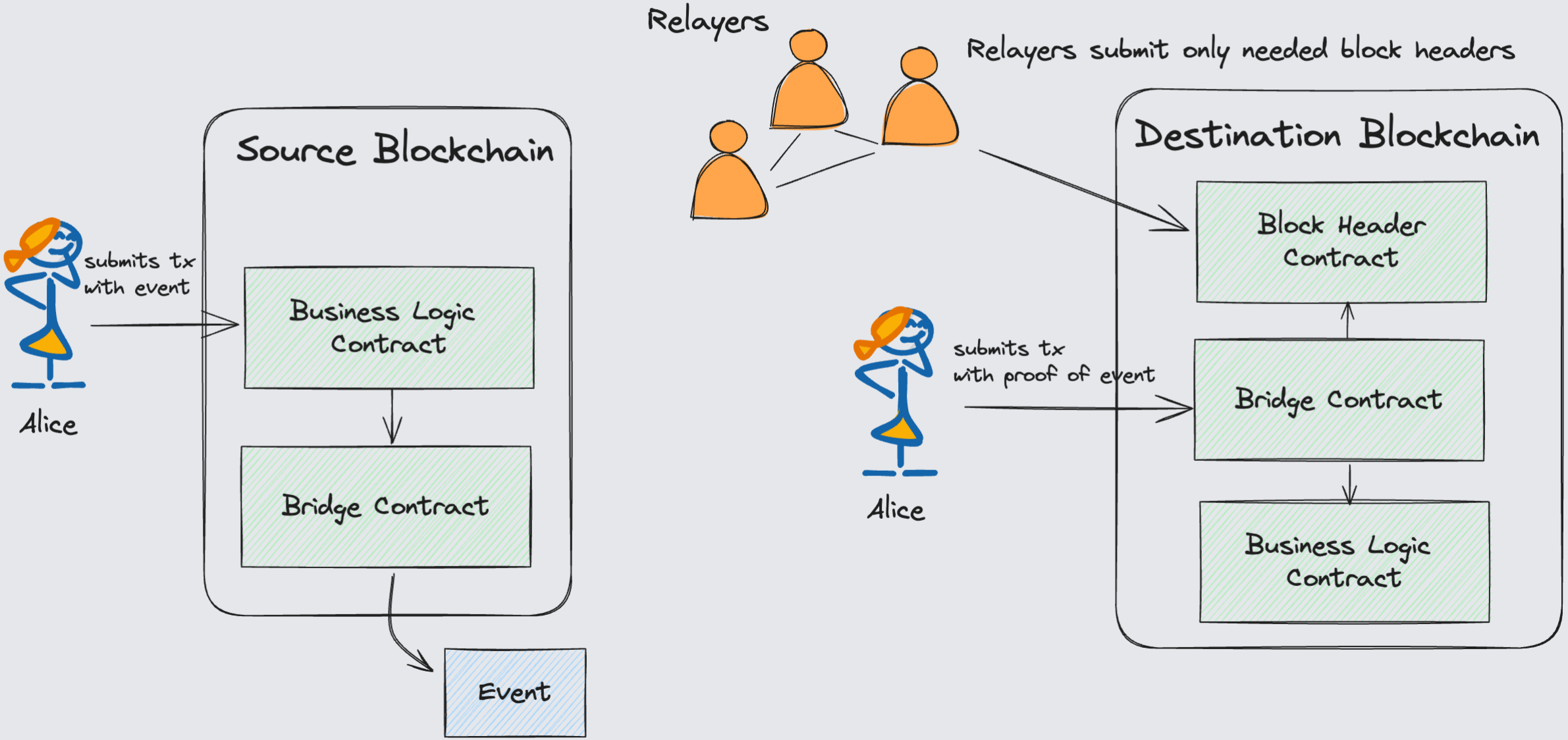
Pros

- One transaction per relayer per block header that is needed.
- Relayers do not need to cooperate.
- Users do not interact with Relayers.

Cons

- Relayers need to analyse transactions in blocks to determine which block headers need to be transferred.
- Relayers are paying to submit transactions on the destination chain.
- User has to be able to submit a transaction on the destination chain.

Block header transfer: relayers cooperate



Block header transfer: relayers cooperate

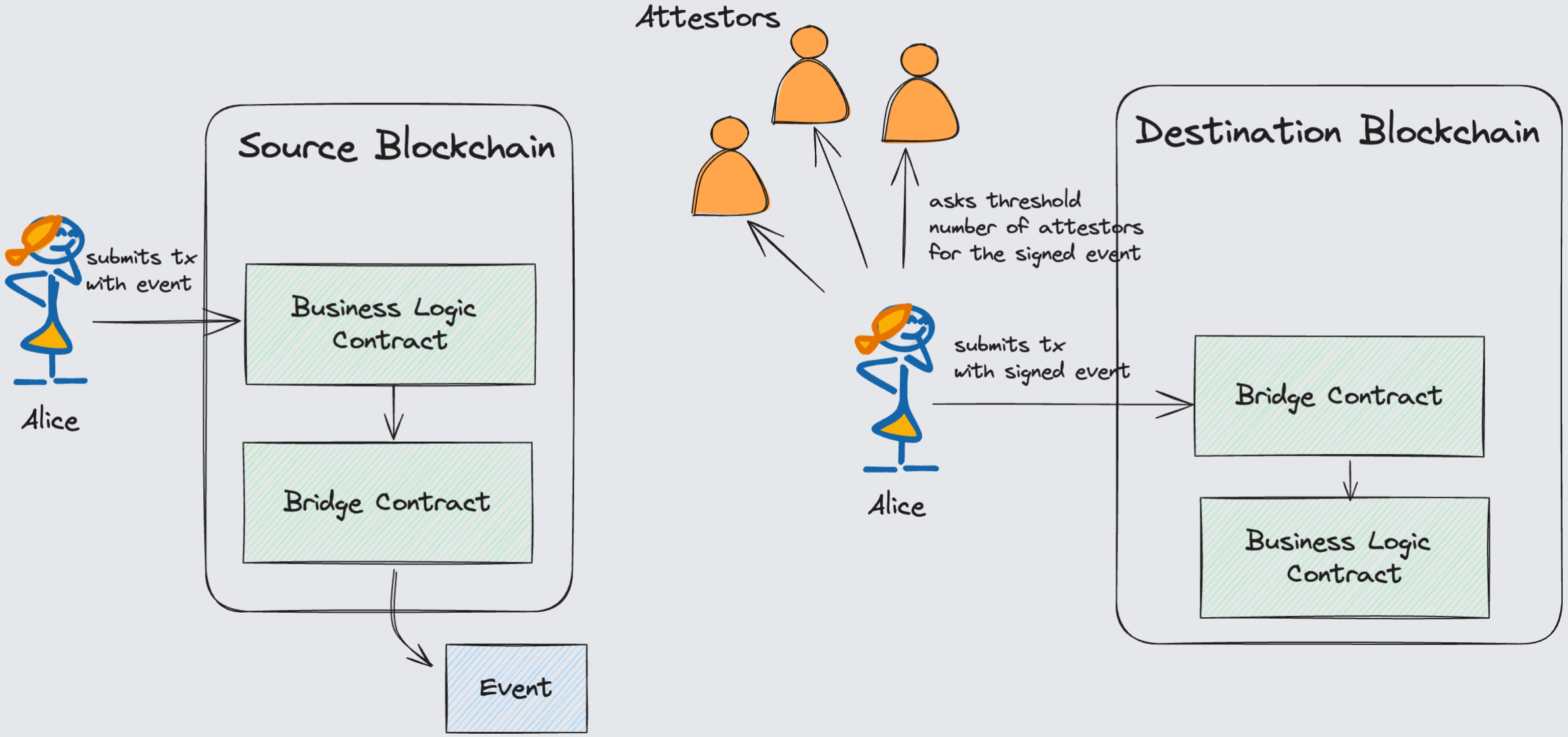
Pros

- One transaction per block header that is needed.
- Users do not interact with Relayers.

Cons

- Relayers need to cooperate.
- Relayers need to analyse transactions in blocks to determine which block headers need to be transferred.
- Relayers are paying to submit transactions on the destination chain.
- User has to be able to submit a transaction on the destination chain.

Event sign: separate attestor sign



Event sign: separate attestor sign

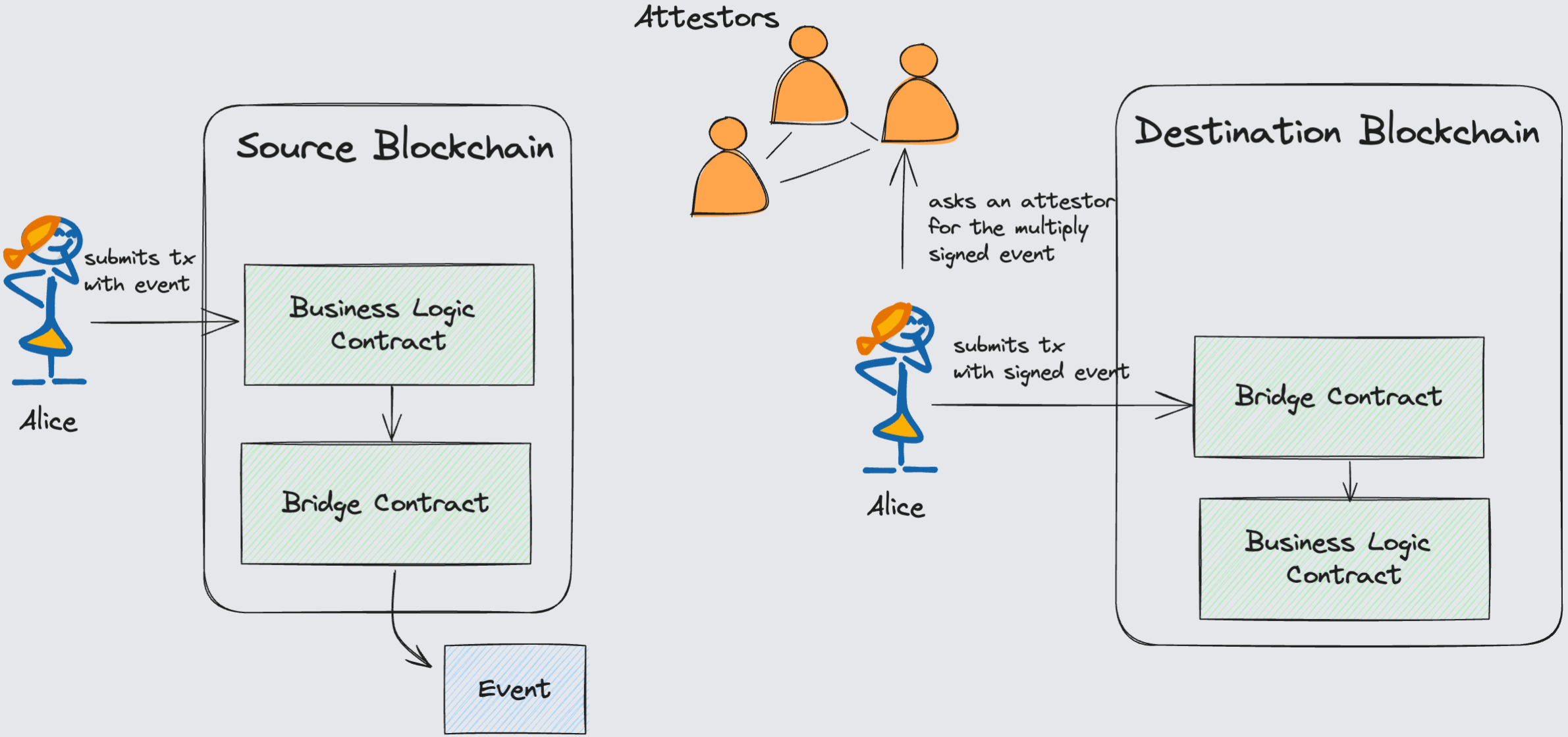
Pros

- Attestors do not submit transactions on the destination blockchain.
- Attestors sign all transactions from a certain contract. They don’t need to know understand the target blockchain.
- Attestors do not need to cooperate.

Cons

- Users interact with Attestors.
- User has to be able to submit a transaction on the destination blockchain.

Event sign: attestors cooperate



Event sign: attestors cooperate

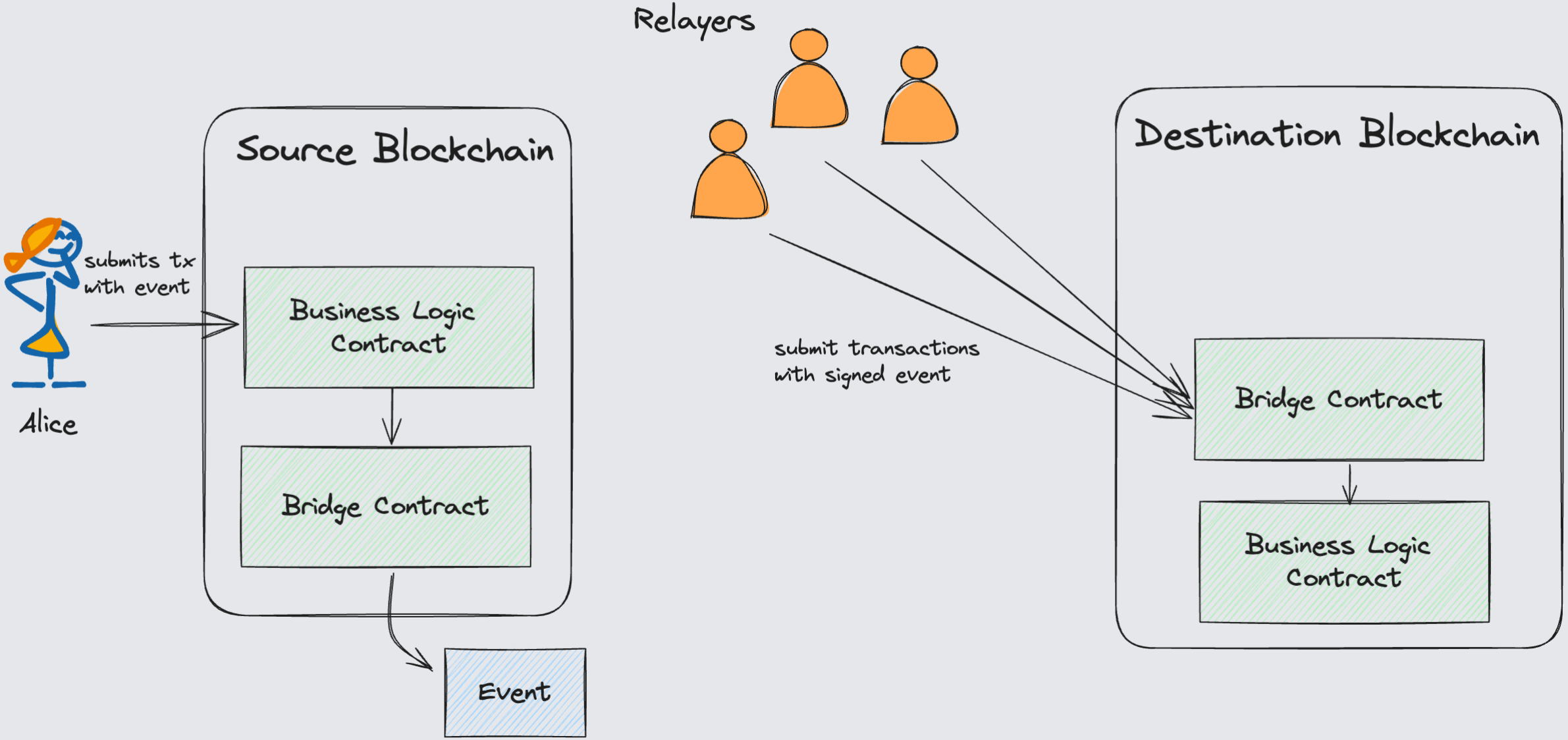
Pros

- Attestors do not submit transactions on the destination blockchain.
- Attestors sign all transactions from a certain contract. They don’t need to know understand the target blockchain.
- Users interact with one attestor.

Cons

- Attestors need to cooperate.
- User has to be able to submit a transaction on the destination blockchain.

Event sign: relayers submit separately



Event sign: relayers submit separately

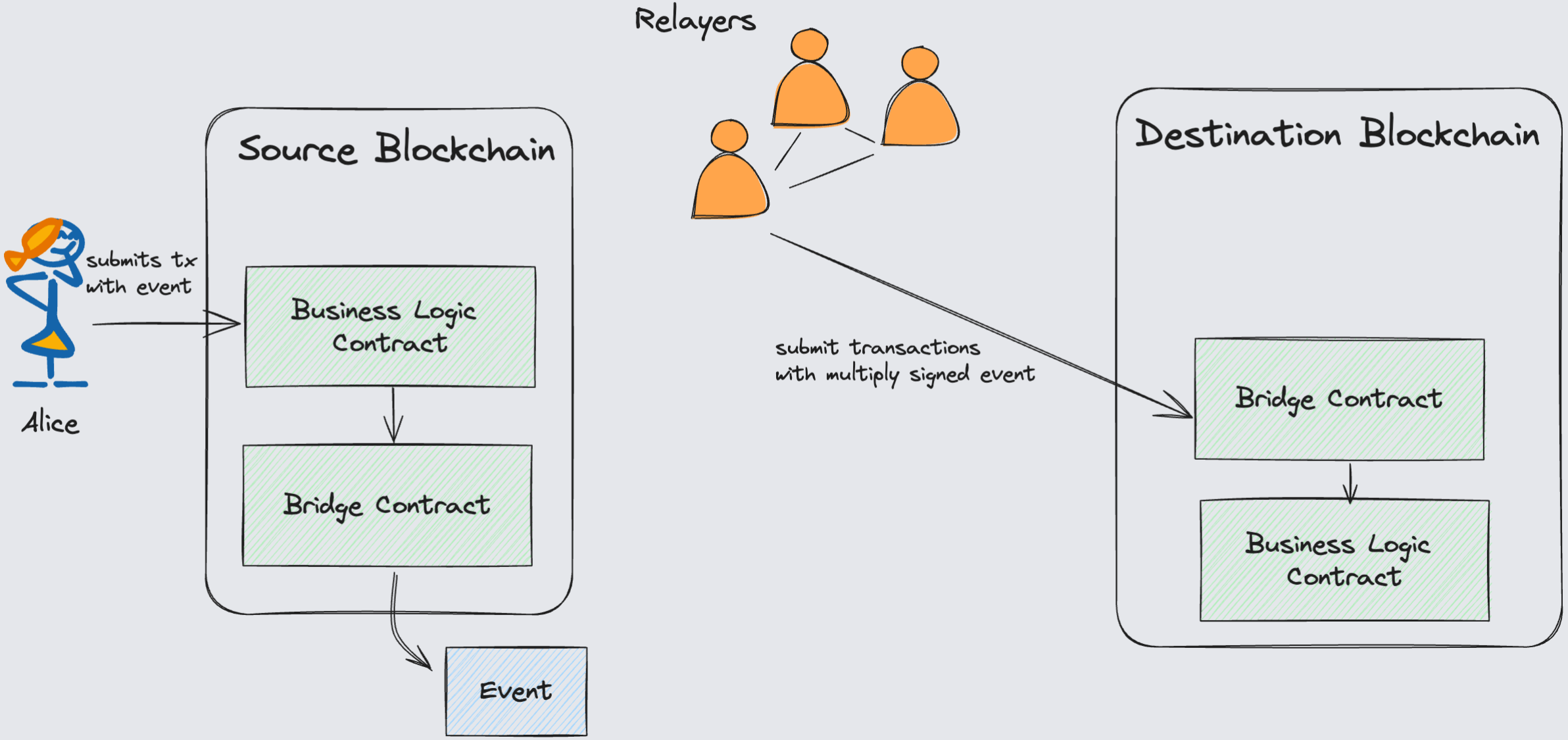
Pros

- Users do not need to be able to submit transactions on the destination blockchain.
- Users do not interact with Relayers.
- Relayers do not need to cooperate.

Cons

- One transaction per Relayer per event.
- Relayers submit transactions on the destination blockchain.

Event sign: relayers cooperate



Event sign: relayers cooperate

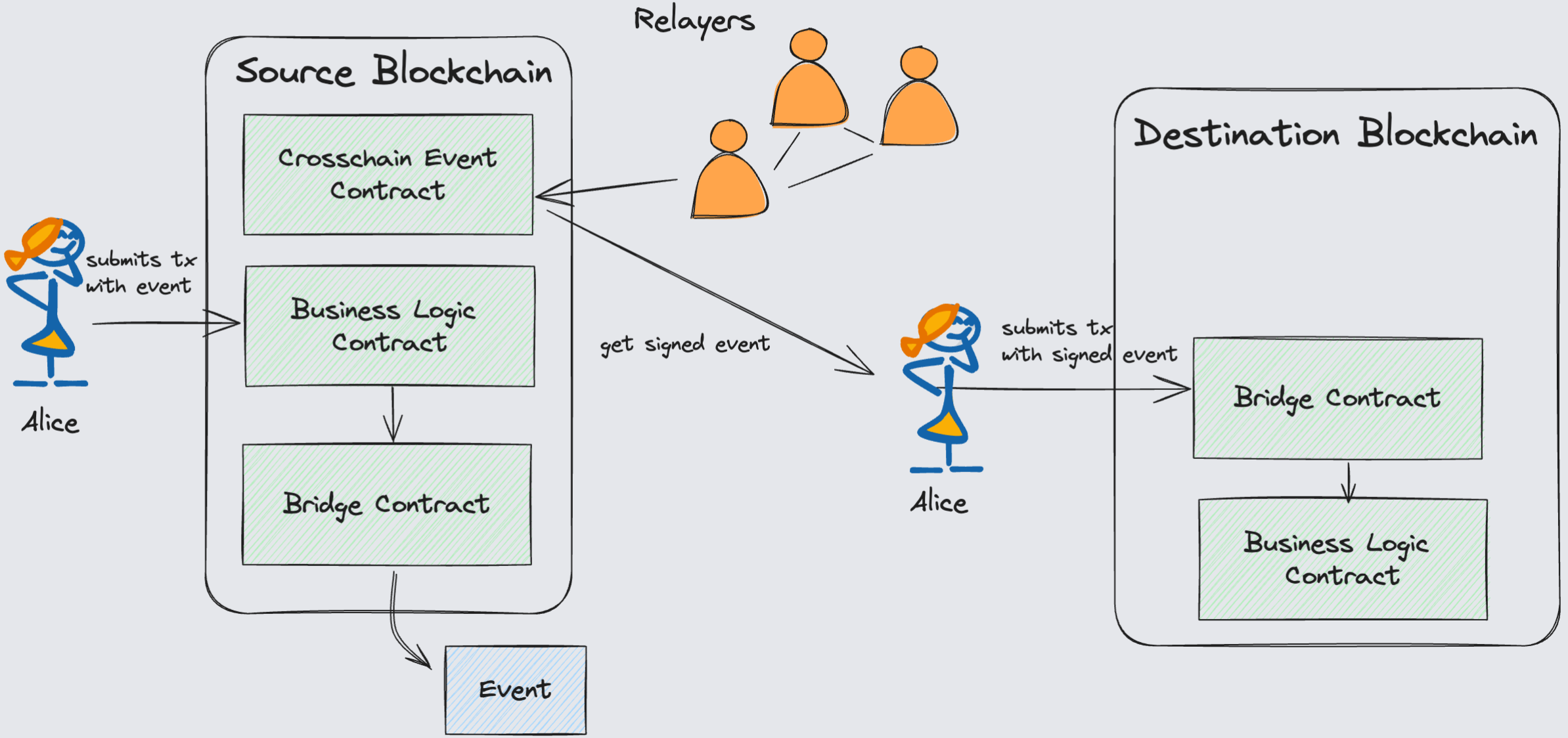
Pros

- One transaction per event.
- Users do not need to be able to submit transactions on the destination blockchain.
- Users do not interact with Relayers.

Cons

- Relayers need to cooperate.
- Relayers submit transactions on the destination blockchain.

Event sign: relayers submit to source



Event sign: relayers submit to source

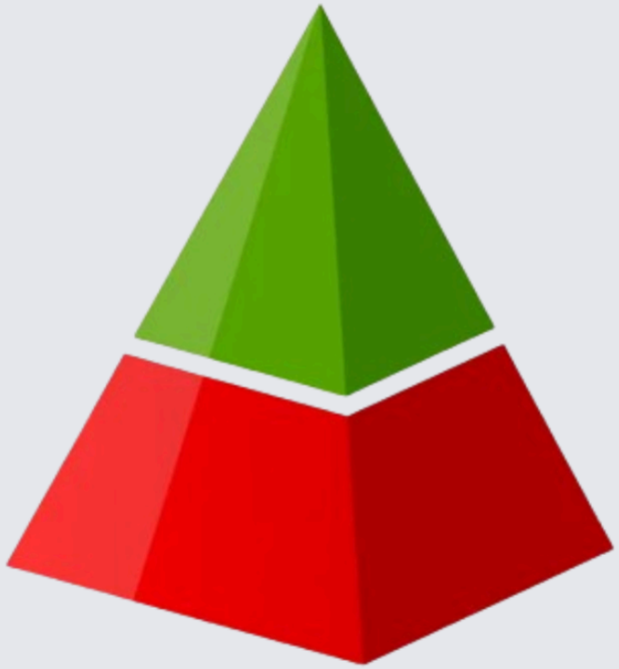
Pros

- Relayers do not submit any transactions to the destination blockchain.
- Users do not interact with Relayers.

Cons

- Relayers need to cooperate.
- User has to be able to submit a transaction on the destination blockchain.

Layer 2



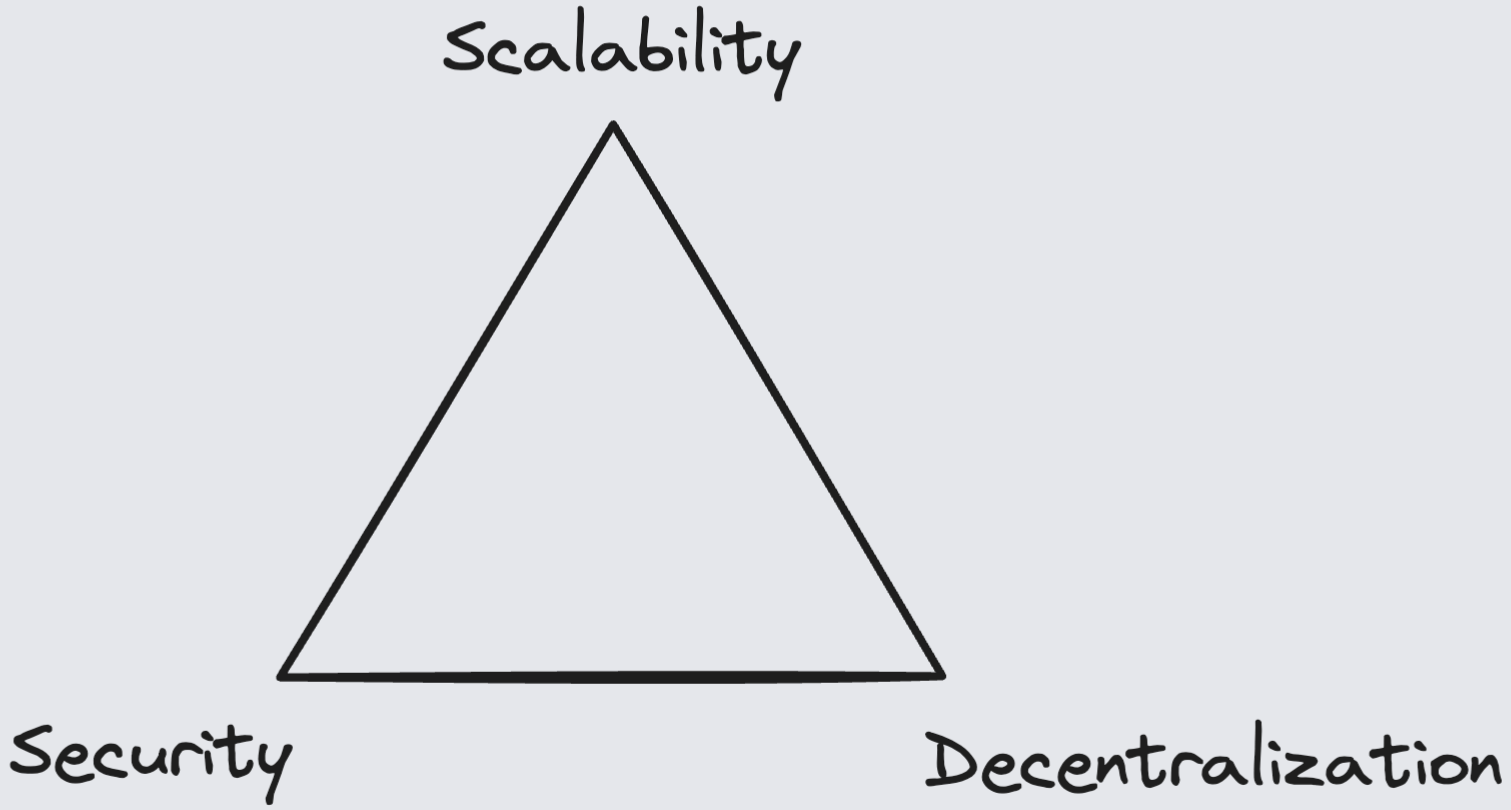
Scaling

Transaction rates (Tx/sec)

- Bitcoin: ~5 Tx/sec
- Ethereum: ~20 Tx/sec

Compared to Visa: 24,000 Tx/sec

The scalability trilemma

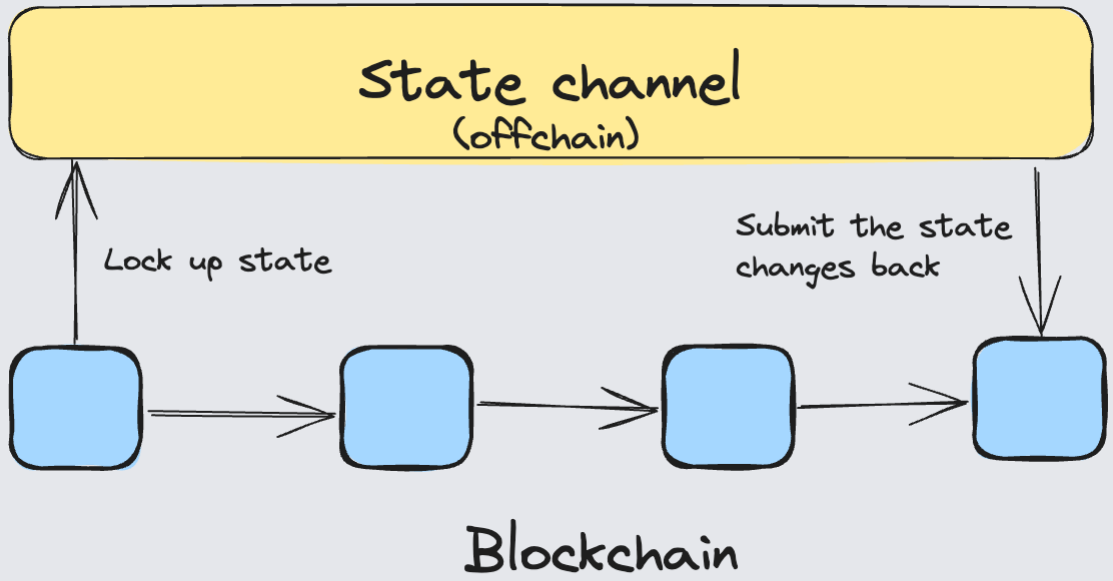


Scaling approaches

- Faster consensus
- State channels
- Sidechains
- Rollups

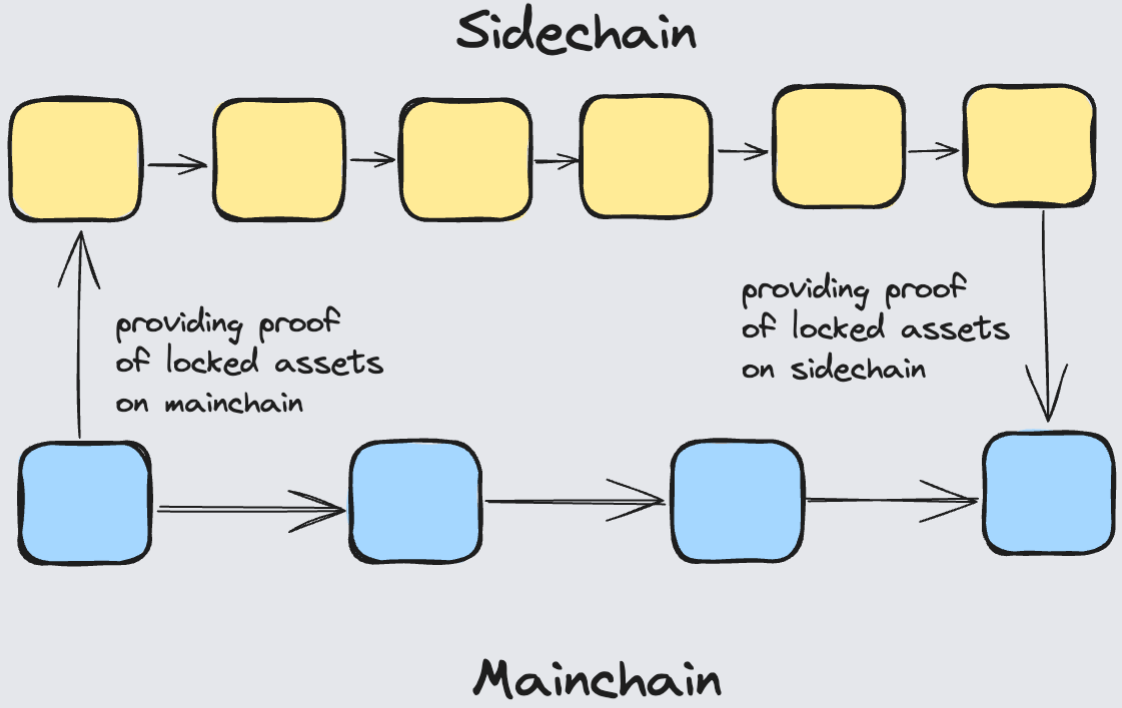
State channel

State channels allow participants to securely transact off-chain while keeping interaction with the blockchain.



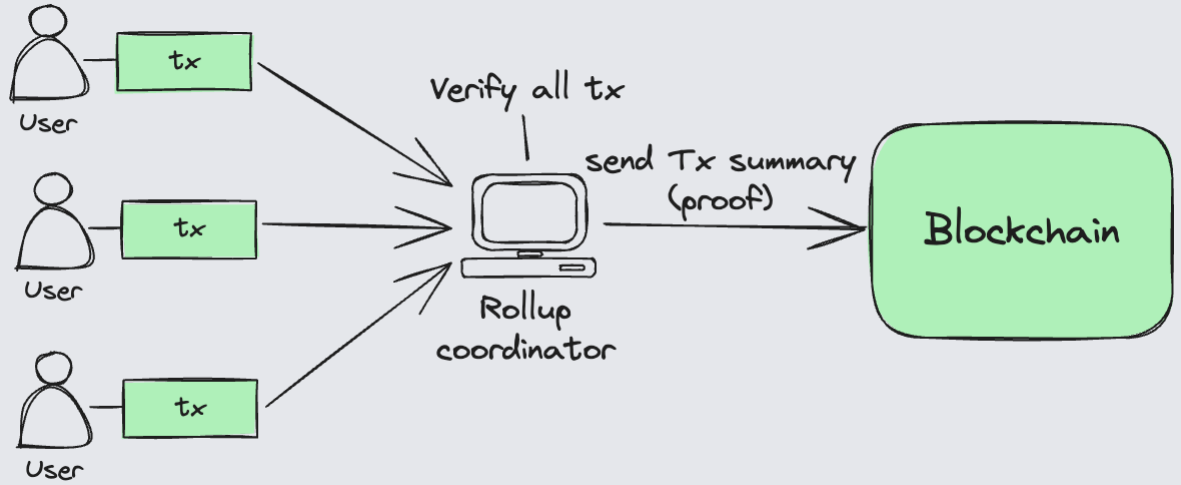
Sidechain

A sidechain is a separate blockchain that is attached to its parent blockchain(mainchain) using a two-way peg.



Rollup

Rollup compresses a bunch of Tx into one on-chain Tx.



zkRollup

- Zero-Knowledge Rollups, or ZK-Rollups, are rollups where validators (operators) prove a transaction’s authenticity without revealing any transaction details.
- They use special cryptographic zero-knowledge proof technologies, such as SNARKs or STARKs.

Optimistic rollup

- Optimistic rollups are called “optimistic” because they assume all the Layer 2 transactions are valid until proven otherwise.
- Anyone can submit a fraud proof and win a reward, rollup server gets slashed.