

CS61065: Theory and Applications of Blockchain

BLOCKCHAIN INTEROPERABILITY

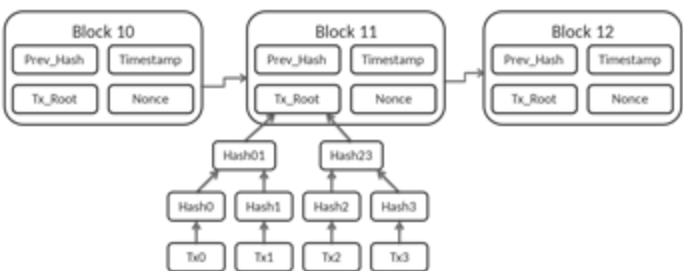
Department of
Computer Science and
Engineering



INDIAN INSTITUTE OF TECHNOLOGY
KHARAGPUR

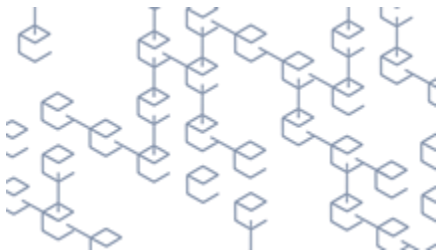
Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

Shamik Sural
shamik@cse.iitkgp.ac.in



Blockchains

**Public/Open
(Permissionless)**



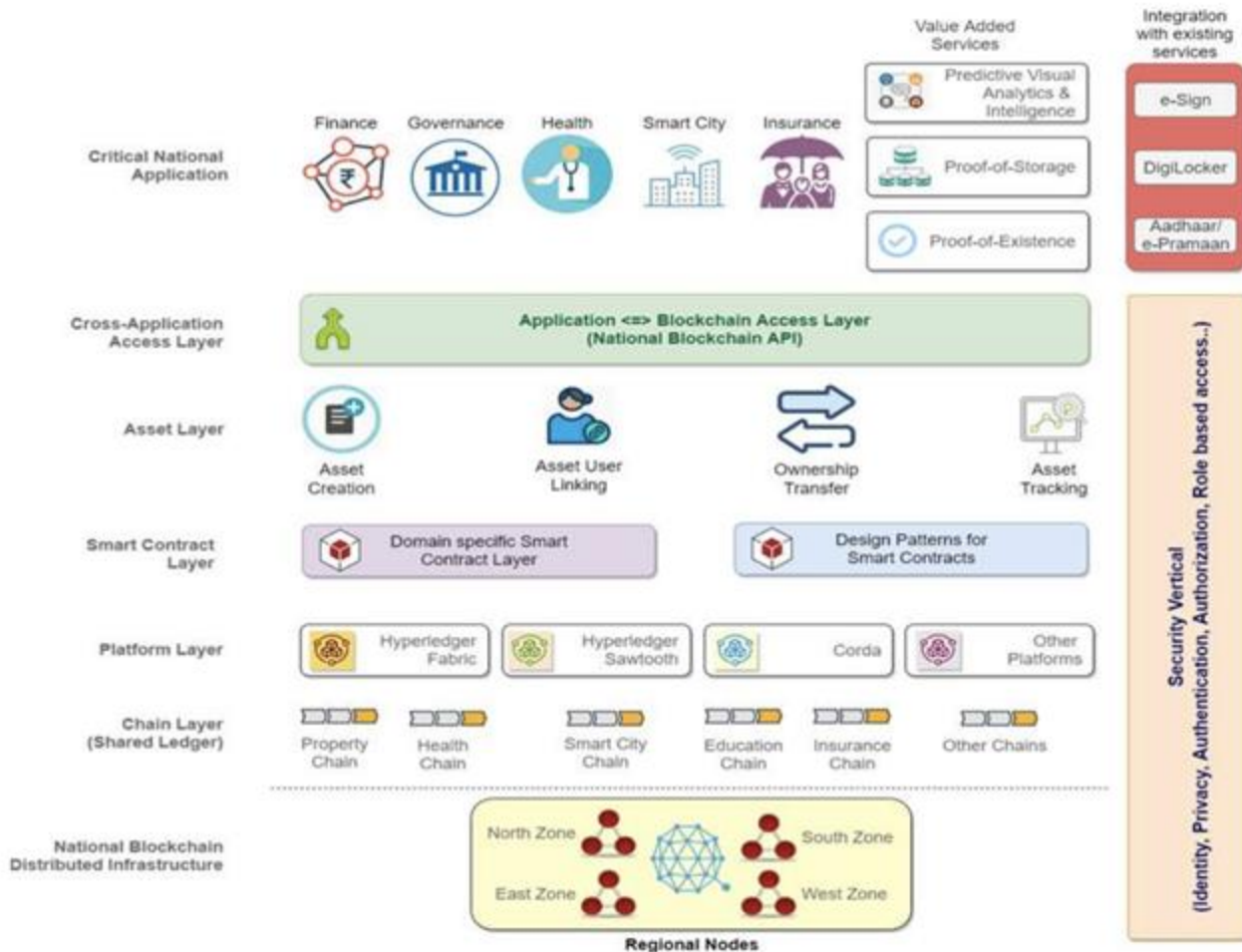
**Private/Closed
(Permissioned)**



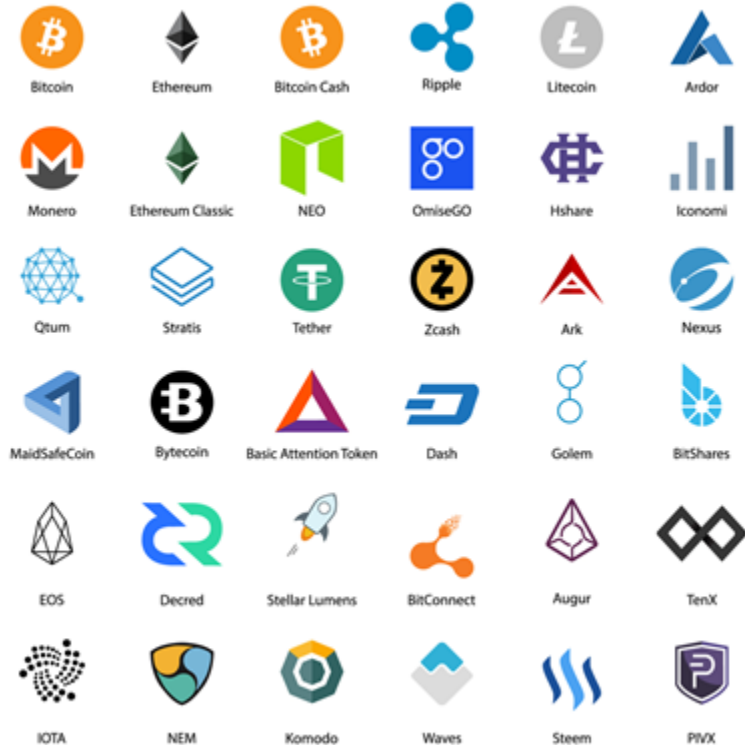
NATIONAL STRATEGY ON BLOCKCHAIN

Government of India
Ministry of Electronics and Information Technology (MeitY)

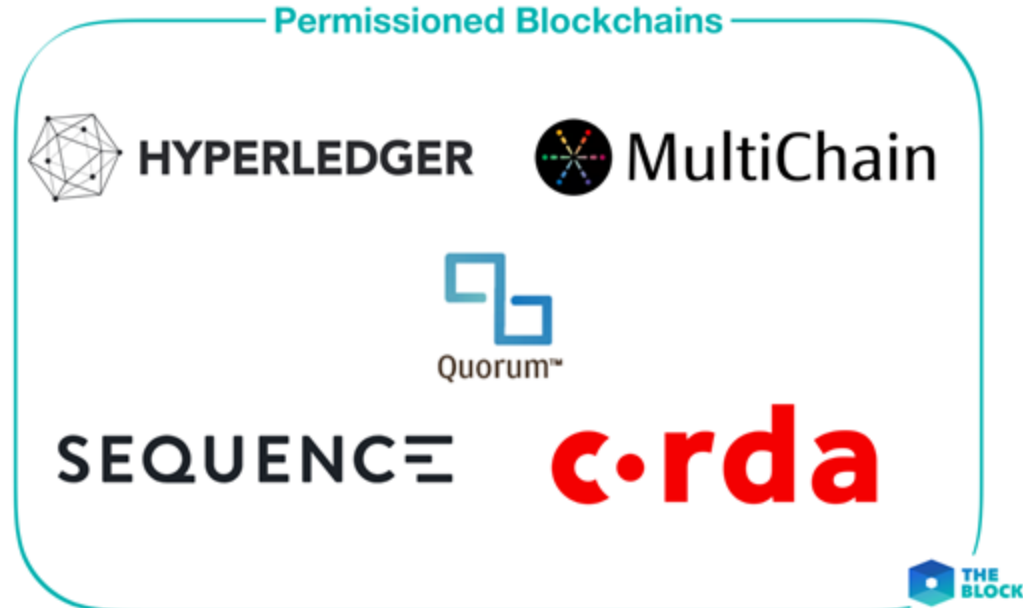
January 2021



Public/Open (Permissionless)



Private/Closed (Permissioned)



Interoperability in Permissionless and Permissioned Blockchains

- Permissionless Blockchain
 - Asset Transfer
 - Crypto currency driven
- Permissioned Blockchain
 - Data Transfer
 - Consensus-driven

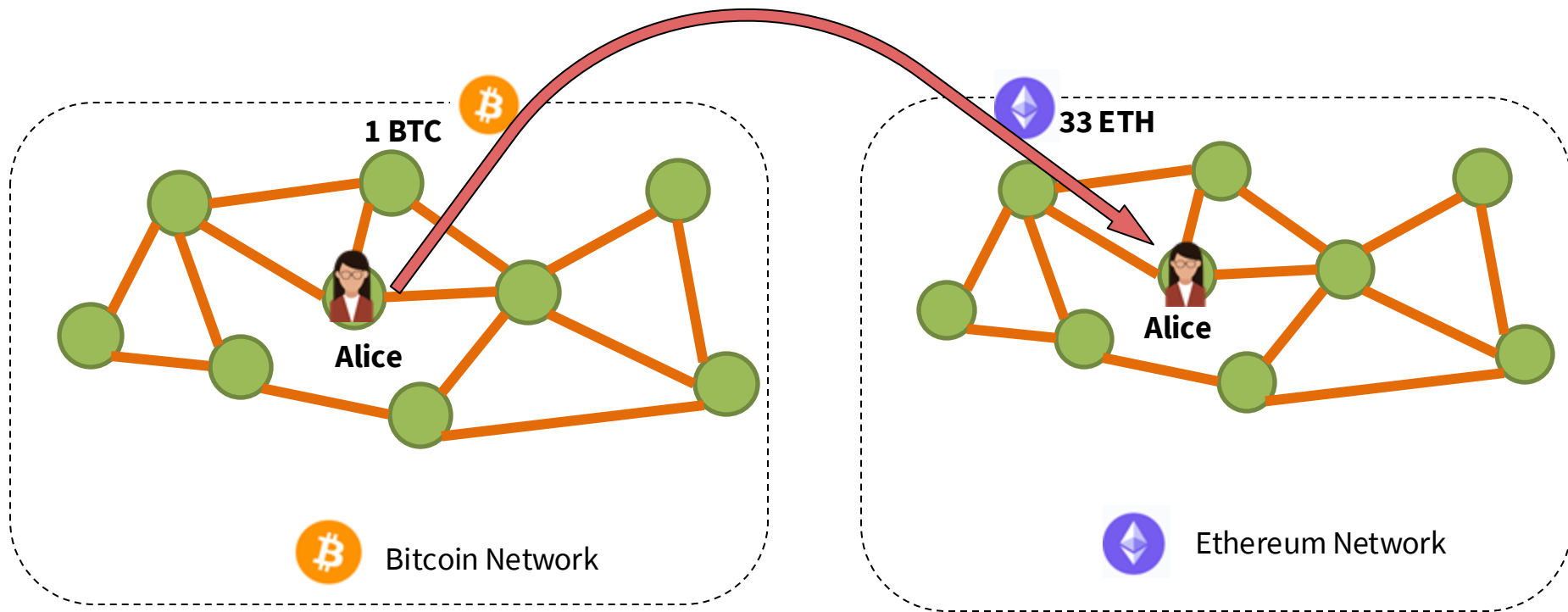
Permissionless Blockchain Interoperability

Public Blockchains as Isolated Silos

- Blockchain-based cryptocurrencies enable secure and trustless currency transactions between parties.
- There are currently **over 2000 different cryptocurrencies in operation.**
- Separate blockchain networks with often different protocols and standards
- **Continue to operate in complete isolation from one another.**

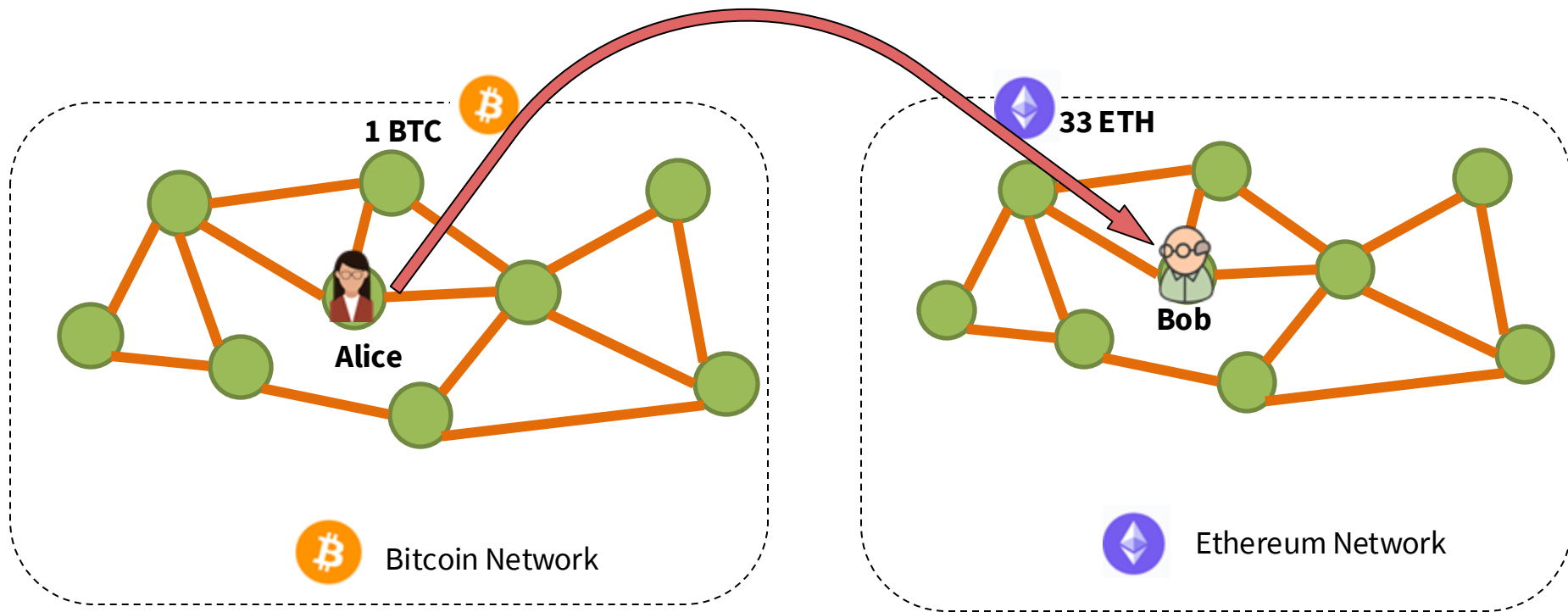


Cross Chain Asset Transfer



Cross Chain Asset Transfer

Possible between different accounts and holders also.



Cross Chain Asset Transfer

To do the transfer, **Alice** must use some **third party who owns ≥ 33 ETH**

1 BTC



33 ETH



Alice



Bob

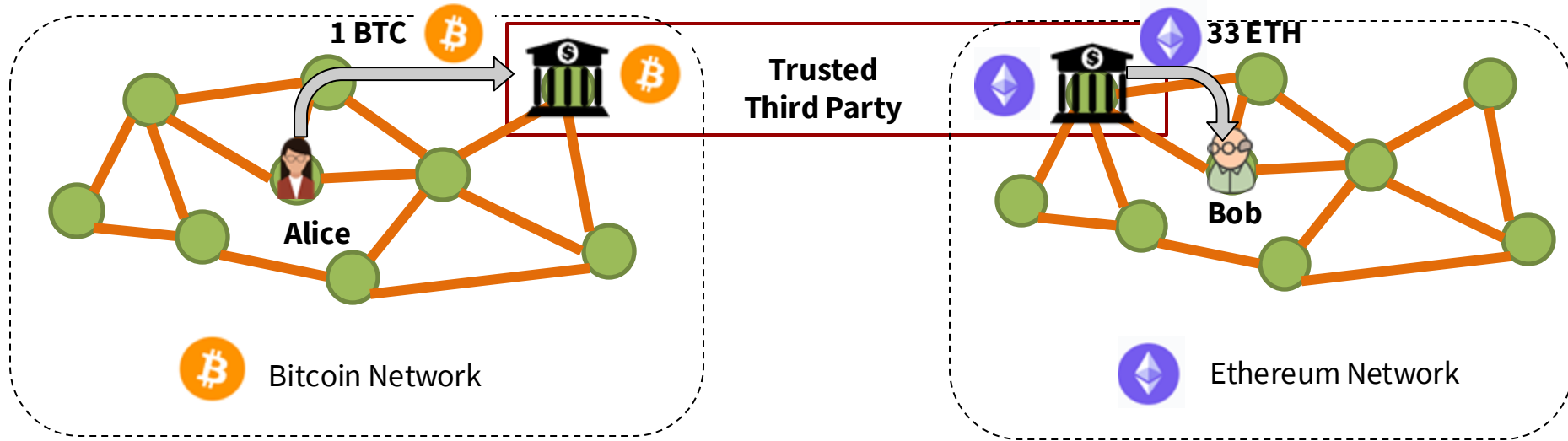


Bitcoin Network



Ethereum Network

Cross Chain Asset Transfer - TTP



TTP based Asset Transfer

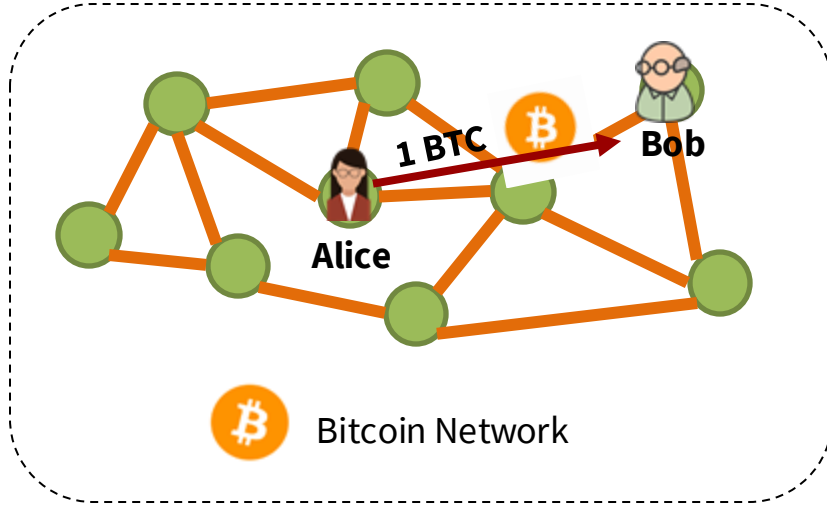
- There are hundreds of centralized cryptocurrency exchanges now.
- **Centralized**, users transfer ownership of their funds to the sole control of the exchange administrator.
- **Fast**, once the deposit is done, the transfer to the destination network is often very fast (in milliseconds).
- **Lack of security**: There has been **numerous cases of theft** from centralized exchanges.
- **650,000 bitcoins lost** when the **MtGox** exchange shut down in 2014.
- Users of the Bitfinex exchange lost approximately **120,000 bitcoins** in 2016

<https://bitcointalk.org/index.php?topic=576337>

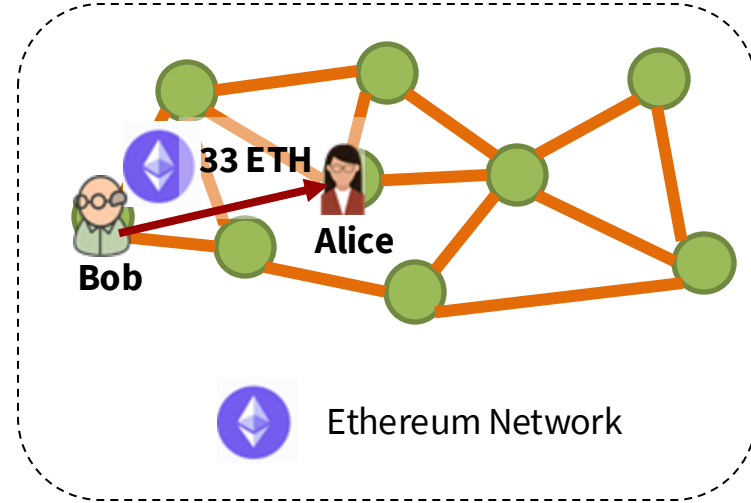
<https://www.reuters.com/article/us-bitcoin-mtgox-wallet-idUSBREA2K05N20140321>

<https://www.reuters.com/article/us-bitfinex-hacked-hongkong-idUSKCN10E0KP>

Asset Exchange



1



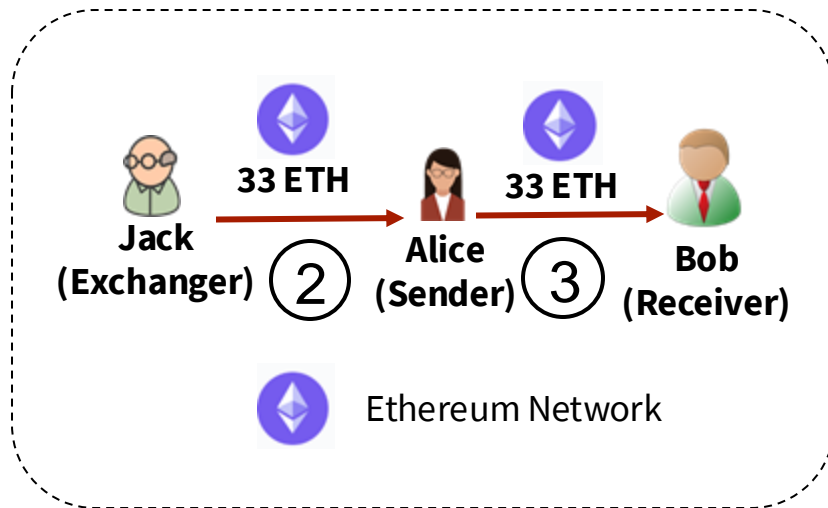
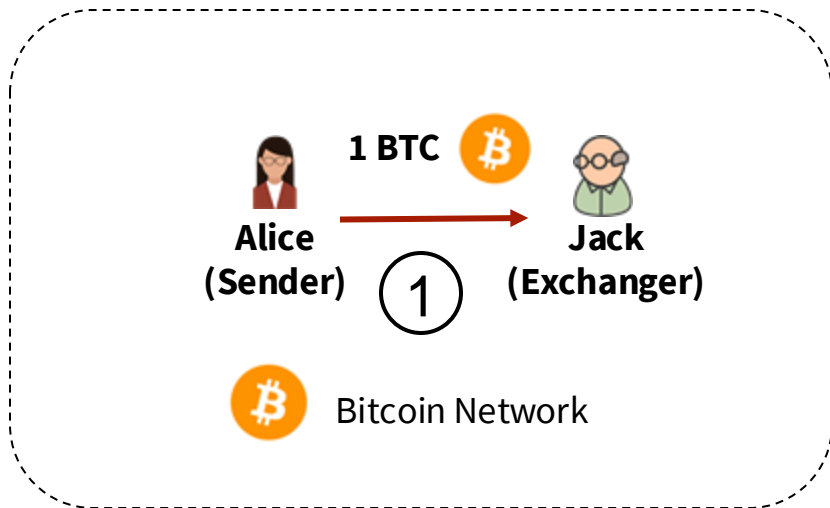
2

- Transfer in both the networks from Alice to Bob (1) and from Bob to Alice (2) must be **ATOMIC**.

Asset Exchange - Problems

- Without the presence of any Escrow, the funds are in control of the sender and receiver parties.
- One party might **abort** the exchange after receiving funds.
- Synchronization problems between the two networks, as well as sender and receiver.
- Difficulty in agreement on exchange rates which may keep on changing every second.

Cross Chain Asset Transfer using Atomic Exchange



① ② Atomic Exchange

③ Transfer

Therefore, solving atomic exchange will solve most challenges of asset transfer.

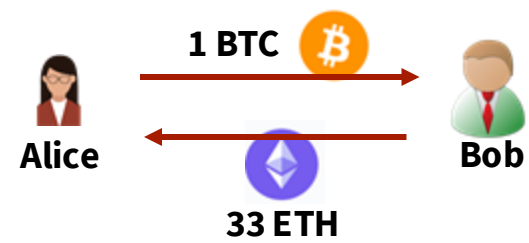
Atomic cross-chain swaps (PODC '18)

Atomicity: An atomic transaction is an indivisible series of operations, such that either all occur, or none occurs.

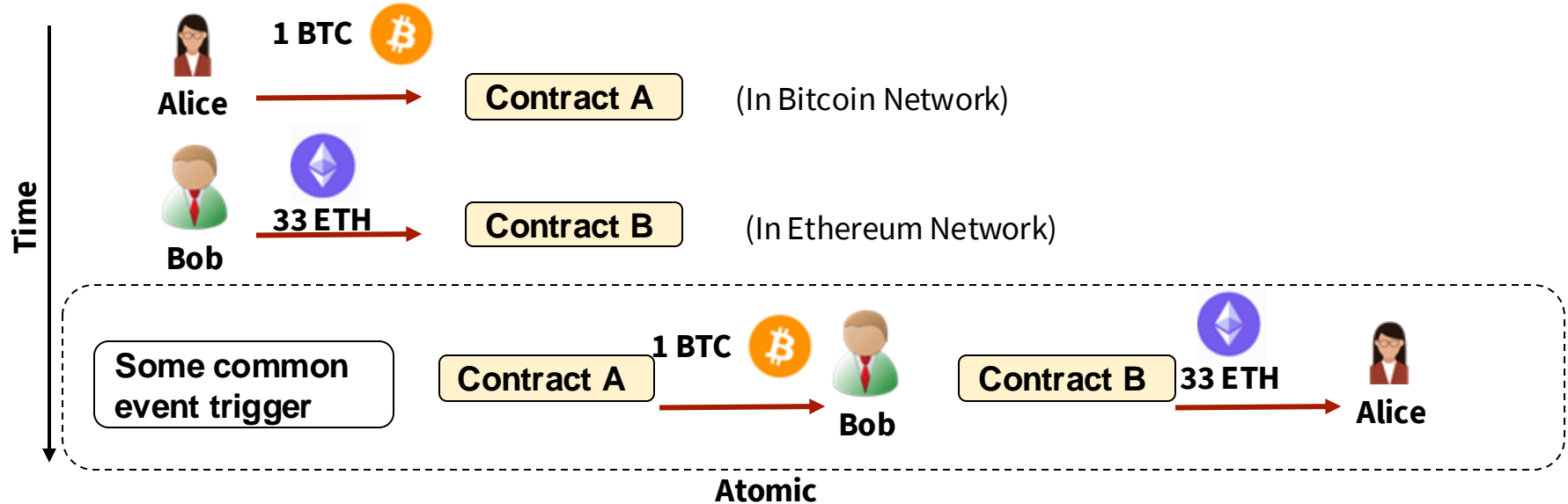
An atomic swap protocol guarantees:

- (1) if all parties conform to the protocol, then all swaps take place
- (2) if some parties deviate from the protocol, then no conforming party ends up worse off
- (3) no coalition has an incentive to deviate from the protocol

Basic Idea



1. Initialize smart contracts on both ends with the amount.
2. Add a **common spending condition**, such that when the condition is met, **both the parties are paid simultaneously**.



Hashlock and Timelock

- **Hashlock:** a function that restricts the spending of funds until a certain piece of data is publicly disclosed (as a cryptographic proof).
 - Hash of a secret pre-image is posted as a hashlock.
 - When the secret is revealed, the funds are released.
- **Timelock:** is a function that restricts the spending of funds until a specific time (**or block height**) in the future.

Hash Locks

Hashlock is a type of encumbrance that restricts the spending of an output until **a specified secret key is publicly revealed**.

Inherent Property: Once any hashlock is opened publicly, any other hashlock secured using the same key can also be opened.

Example:

Alice generates a secret **key** **“I love strawberries”**

Alice computes the Cryptographic Hash of the key: **f1b81571baac90bed544d1910f79ea5c31fa4509**

Alice initiates a Hash Locked contract of **1 BTC** (some amount) which has the **conditions**:
If **key is revealed** - **pay BOB** with 1 BTC.

The contract also contains the Hash, which allows any miner to verify the revealed key.

Time Locks

Timelock is a type of smart contract primitive that restricts the spending/transfer of some currency until a specified future time.

Block height may be used as a proxy for time.

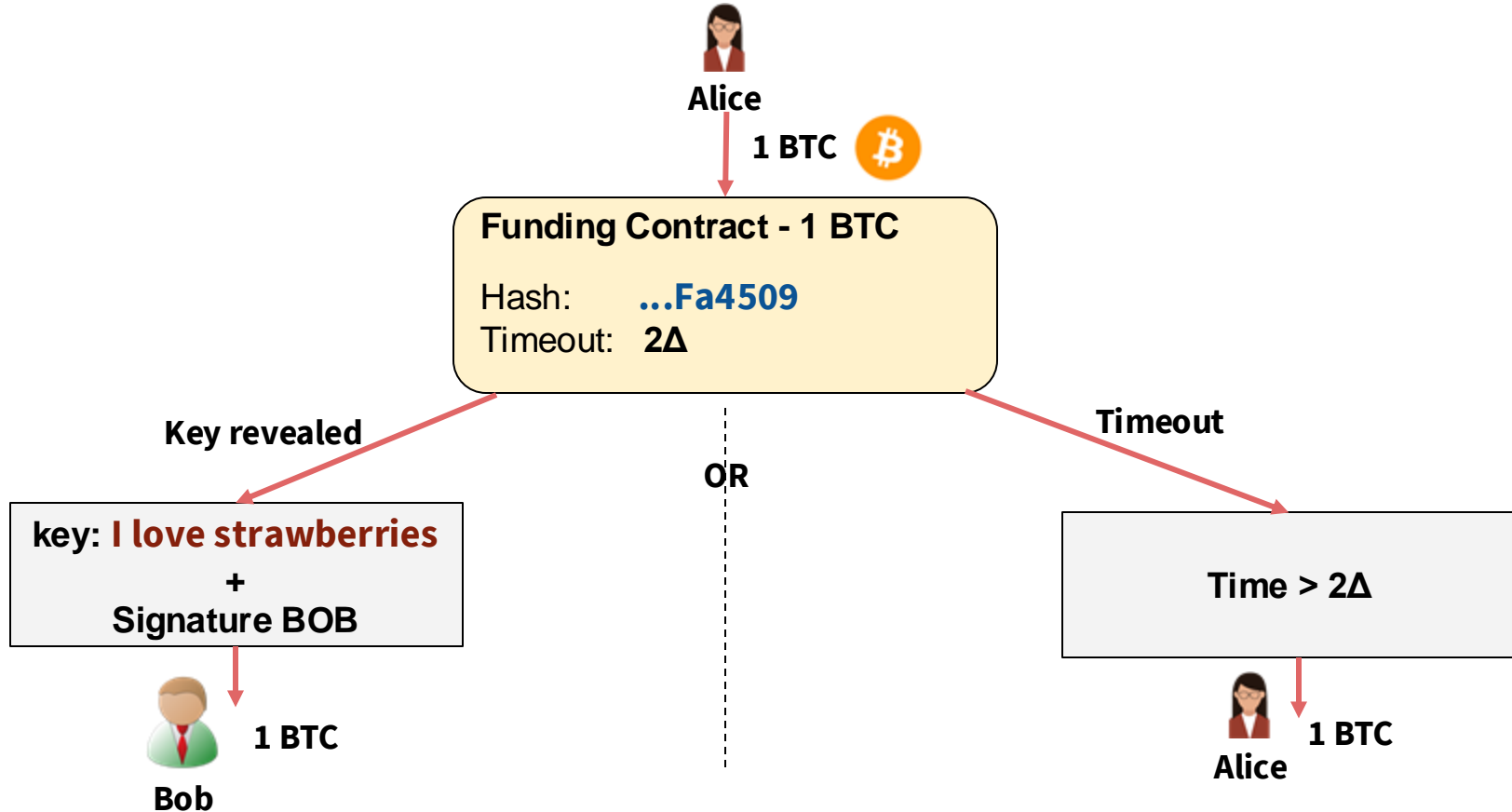
Example:

Alice generates a timelocked contract with 1 BTC, and time = 2Δ (Δ = some time unit)

Therefore, after 2Δ time, 1 BTC will be transferred to a **target account**.

(The target account can be Alice's own account)

HTLC - Hash Time Locked Contract

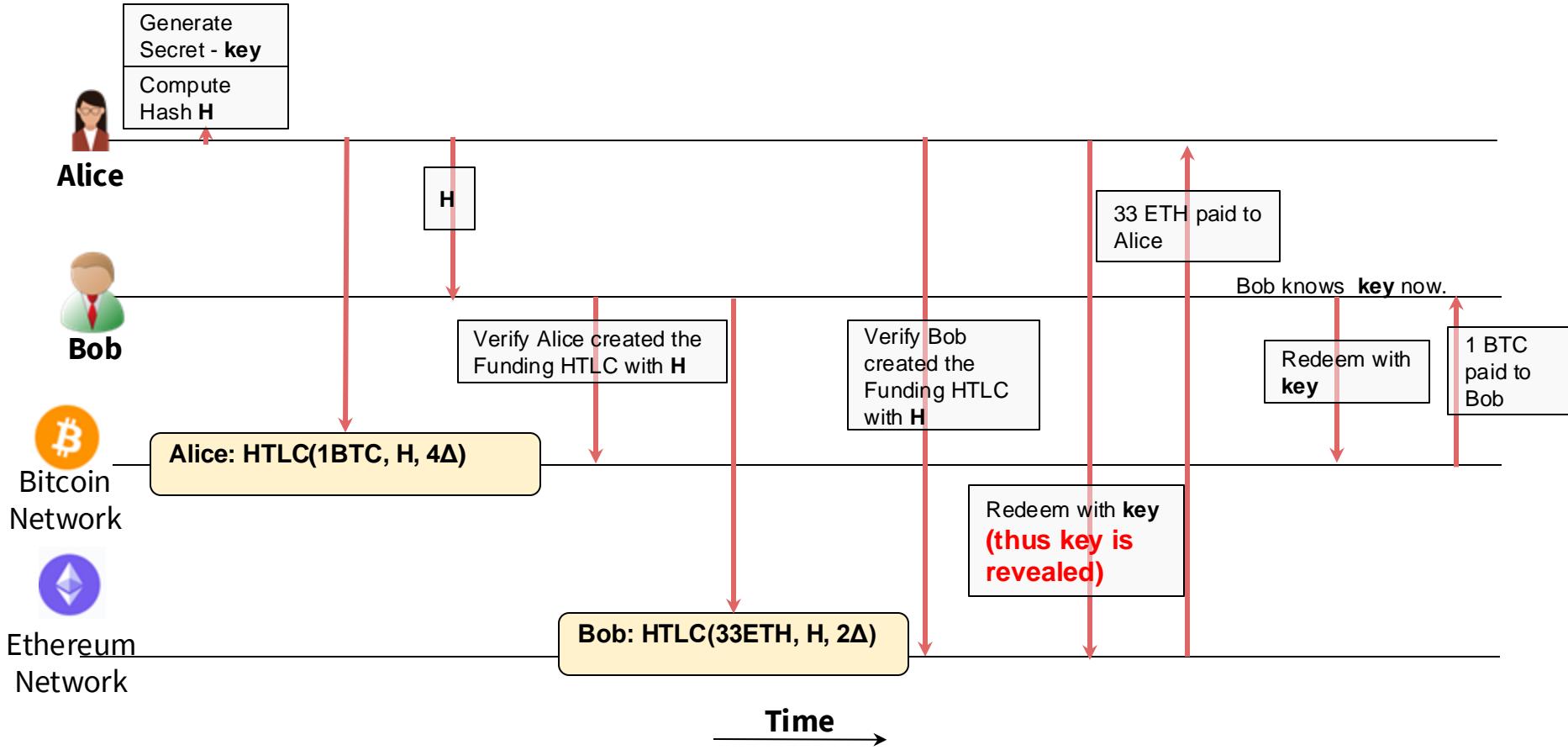


HTLC for Atomic Swap

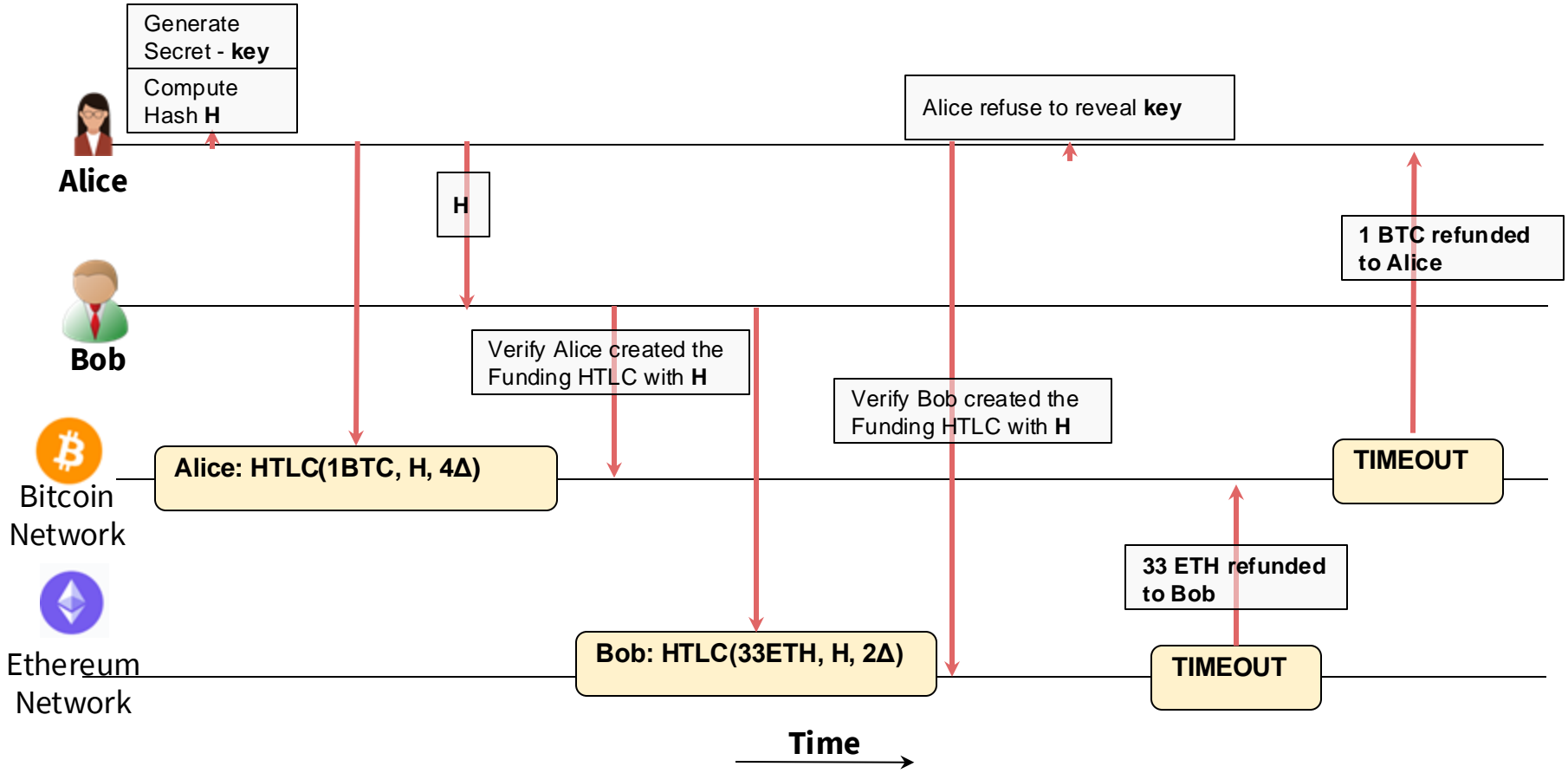
- (1) Alice samples a random $x \in \{0, 1\}^\lambda$, computes a hash commitment $Y = \text{hash}(x)$, and broadcasts a transaction TX_A that spends n_1 BTC into an output script that dictates:
 - Alice can gain back possession of her n_1 BTC after $c_0 + t_0 + s_0$ blocks.
 - Bob can redeem the n_1 BTC by supplying a preimage of Y and signing with his secret key.
- (2) After TX_A is buried under c_0 extra blocks and therefore becomes irreversible w.h.p., Bob broadcasts a transaction TX_B that spends his n_2 LTC into an output script that dictates:
 - Bob can gain back possession of his n_2 LTC after $4t_0$ blocks.
 - Alice can redeem the n_2 LTC by supplying a preimage of Y and signing with her secret key.
- (3) After TX_B is buried under $f(c_0)$ extra blocks and therefore becomes irreversible w.h.p., Alice redeems the n_2 LTC of Bob by revealing x .
- (4) Bob supplies x to redeem the n_1 BTC of Alice.

For a nice illustrative example
[Visit this site](#)

HTLC for atomic swap



What if Alice does not reveal key ?



Multi-Party Atomic Cross-chain Swap

- Carol wants to sell her Cadillac for bitcoins. Alice is willing to buy Carol's Cadillac, but she wants to pay in an "alt-coin" cryptocurrency
- Bob is willing to trade alt-coins for bitcoins
- Alice, Bob and Carol need to arrange a three-way swap:
 - Alice will transfer her alt-coins to Bob
 - Bob will transfer his bitcoins to Carol
 - Carol will transfer title of her Cadillac to Alice

Multi-Party Atomic Cross-chain Swap

- Alice creates a secret s , $h = H(s)$, and publishes a contract on the alt-coin blockchain with hashlock h and timelock 6Δ in the future, to transfer her alt-coins to Bob.
- When Bob confirms that Alice's contract has been published on the alt-coin blockchain, he publishes a contract on the Bitcoin blockchain with the same hashlock h but with timelock 5Δ in the future, to transfer his bitcoins to Carol.
- When Carol confirms that Bob's contract has been published on the Bitcoin blockchain, she publishes a contract on the automobile title blockchain with the same hashlock h , but with timeout 4Δ in the future, to transfer her Cadillac's title to Alice.
- When Alice confirms that Carol's contract has been published on the title blockchain, she sends s to Carol's contract, acquiring the title and revealing s to Carol.
- Carol then sends s to Bob's contract, acquiring the bitcoins and revealing s to Bob.
- Bob sends s to Alice's contract, acquiring the alt-coins and completing the swap.

Multi-Party Atomic Cross-chain Swap

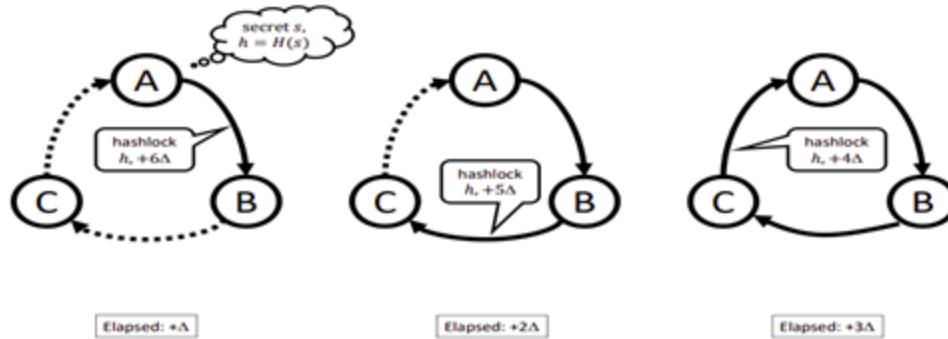


Figure 1: Atomic cross-chain swap: deploying contracts

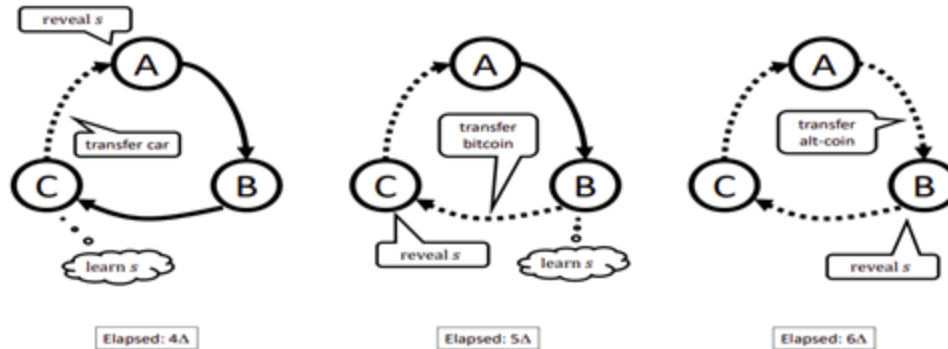


Figure 2: Atomic cross-chain swap: triggering arcs

Validity of the Protocol

- If any party halts while contracts are being deployed, then all contracts eventually time out and trigger refunds.
- If any party halts during triggering of contracts, only that party ends up worse off.
 - If Carol halts without triggering her contract, then Alice gets the Cadillac and Bob gets a refund, so Carol's misbehavior harms only herself.
- The order in which contracts are deployed matters.
 - If Carol were to post her contract with Alice before Bob posts his contract with Carol, then Alice could take ownership of the Cadillac without paying Carol.
- Timelock values matter.
 - If Carol's contract with Bob were to expire at the same time as Bob's contract with Alice, then Carol could reveal s to collect Bob's bitcoins at the very last moment, leaving Bob no time to collect his alt-coins from Alice.

Multi-Party Atomic Cross-chain Swap

- What if parties behave irrationally?
 - If Alice (irrationally) reveals s before the first phase completes, then Bob can take Alice's altcoins and perhaps Carol can take Bob's bitcoins, but Alice will not get her Cadillac, so only she is worse off.
- Atomic swap protocol guarantees
 - If all parties conform to the protocol, then all swaps take place
 - If some parties deviate from the protocol, then no conforming party ends up worse off
 - No coalition has an incentive to deviate from the protocol

Limitations of this approach

- Slow, often takes **hours to execute**.
- Parties cannot respond to real time price fluctuations due to slow transactions.
- Vulnerable to a weak denial-of-service attack where an adversarial party repeatedly proposes an attractive swap, and then fails to complete the protocol, triggering refunds, but temporarily rendering assets inaccessible.

Permissioned Blockchain Interoperability

Interoperation in Permissioned Blockchains

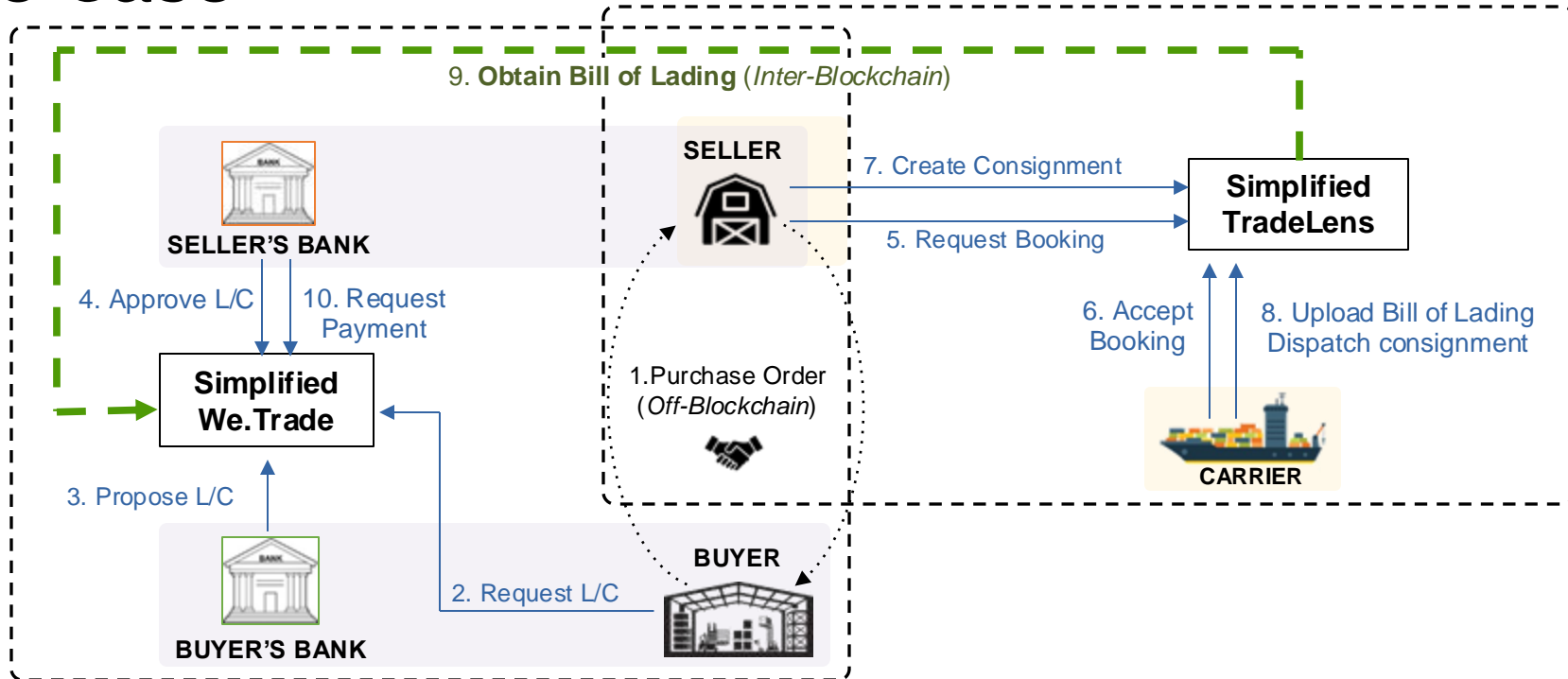
- Permissioned blockchain networks are designed to be **private**, for a closed consortium.
- **Different business sectors** tend to have different groups of organizations, thus have **separate blockchain networks**.
- Eg: **TradeLens** - Logistics, **We.Trade** Trade finance, **IBM Food Trust** - Food Supply Chain, etc..
- **Continue to operate in complete isolation from one another.**
- Often there is **need for interoperation** between different isolated networks to achieve business goals.

Interoperation in Permissioned Blockchains

Challenges

- Here interoperation is specifically **Verifiable Data Transfer** between two separate permissioned blockchain networks.
- Data in a blockchain network is generated by transactions going through **consensus process**.
- Data has to be **consistent with the source network's state**.
- **Multiparty Trust** - When one network consumes state from another, it would need to establish the validity of the state according to the shared consensus view of parties in the network.

Use Case

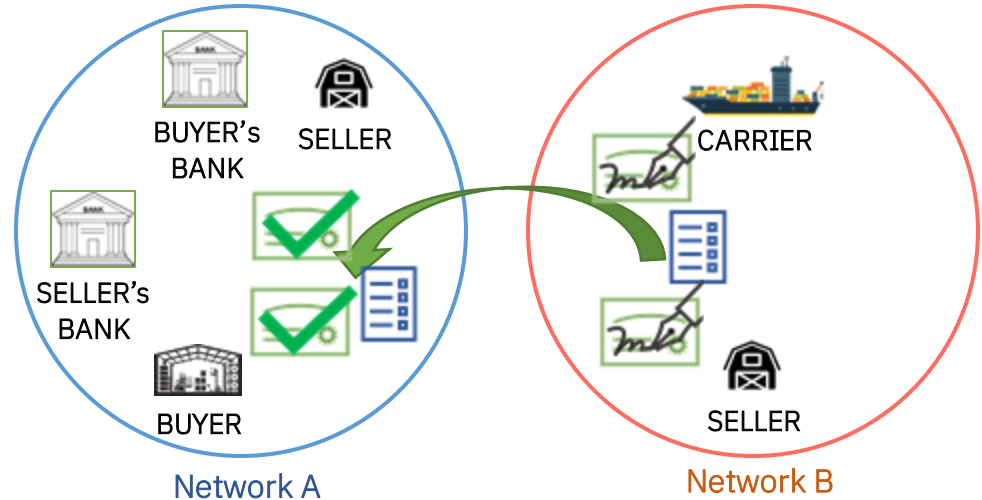


- **We.Trade** is a trade financing network
- **TradeLens** is trade logistics network
- **Letter of Credit (L/C)** – a bank confirmation that ensures payment will be made by the buyer
- **Bill of lading** proves that the seller has dispatched the goods via the carrier,
- It enforces an obligation on the buyer (as per letter of credit terms) to make a payment.

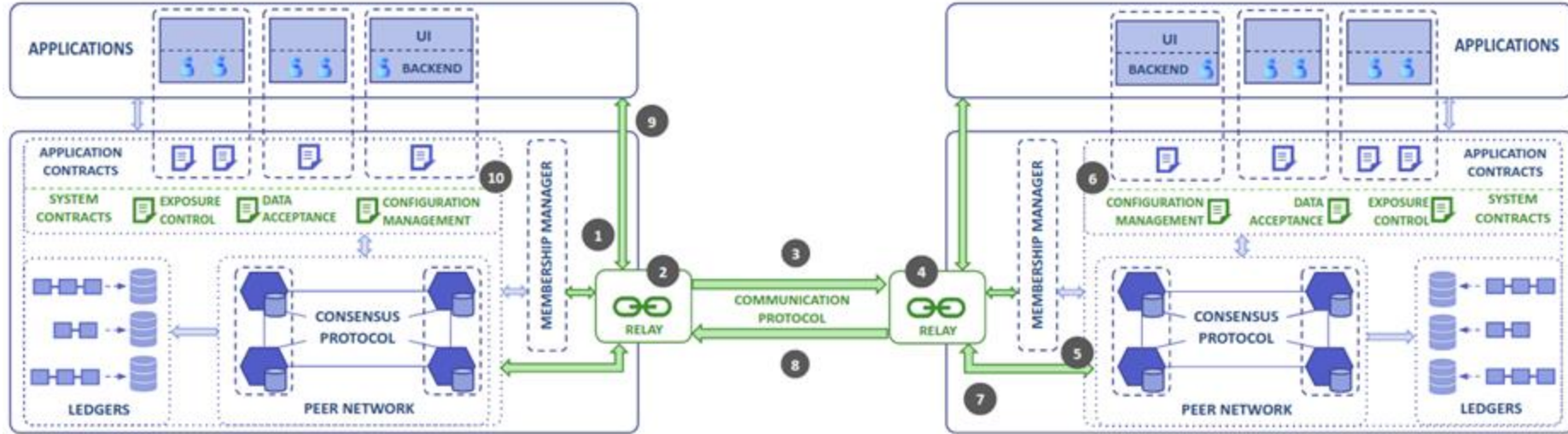
Verifiable Data Transfer in Permissioned Blockchain

Enabling Enterprise Blockchain Interoperability with Trusted Data Transfer (Middleware '2019)

- Each data (block) in a network has a set of **endorsements** (signatures) **for consensus**.
- This set of endorsements confirm the validity of a block in the network.
- Thus, from the source, **data is accompanied with the set of signatures - Attestations**.
- The attestations are **validated** in the destination network according to an **data acceptance policy**.



Architecture

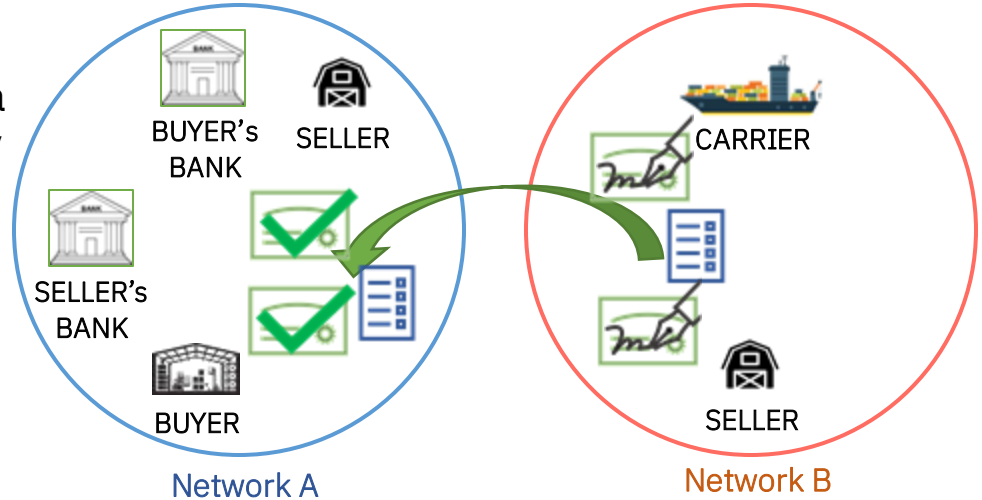


- **Relay Service:** Provides the means of communication between the two separate networks
- **Configuration Management Contract:** Maintain identity (public keys) of the interoperating networks.
- **Exposure Control Contract:** Define and enforce policies on what data to expose to which network.
- **Data Acceptance:** Validate the data received from a foreign network against the policy that defines how many signatures are required (and other conditions) to verify a data.

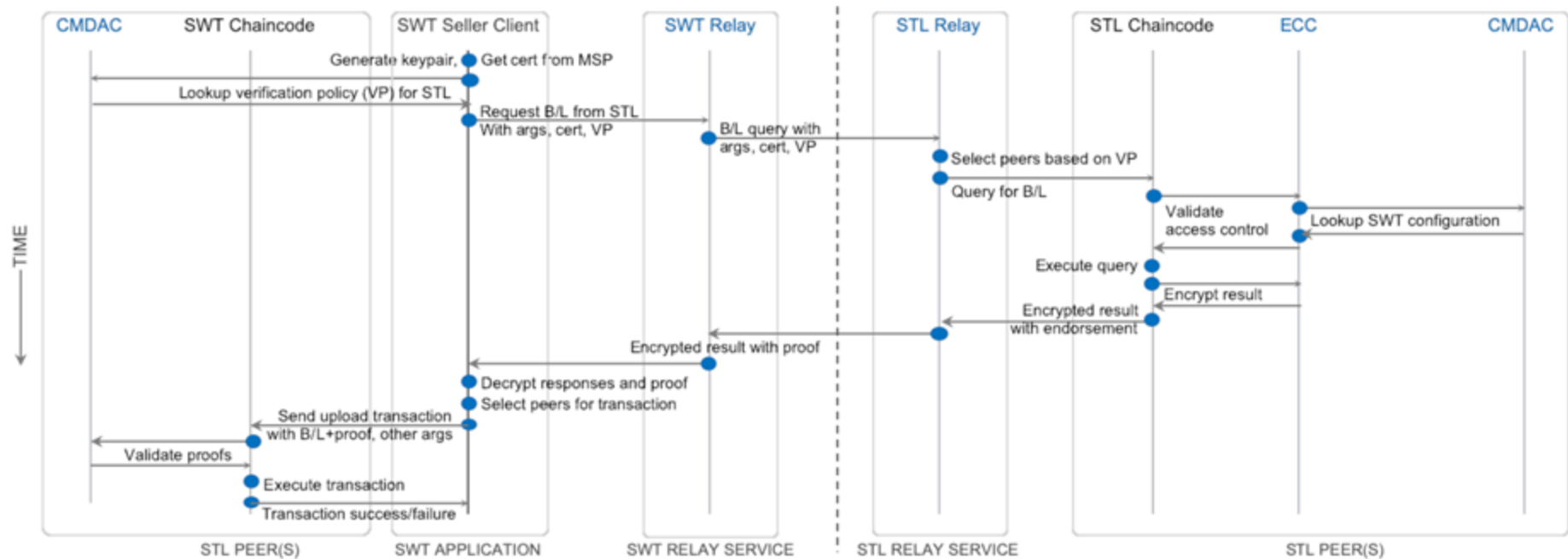
Protocol Overview

Steps:

1. Proof request is generated consisting of a verification policy which has to be met by the source network.
2. Access control policies are checked.
3. Response data along with proofs (endorsements) is sent back through the relay.
4. The proofs are validated against the verification policy.



Protocol Overview



Tesseract (CCS '19)

- **Real time** - as fast as centralized exchanges.
- Can observe the buy (a.k.a. “bid”) and sell (a.k.a. “ask”) prices in real time and **allow parties to modify their trading positions in milliseconds.**
- Supports **cross chain asset exchanges.**
- Secure against theft by exchange operators.

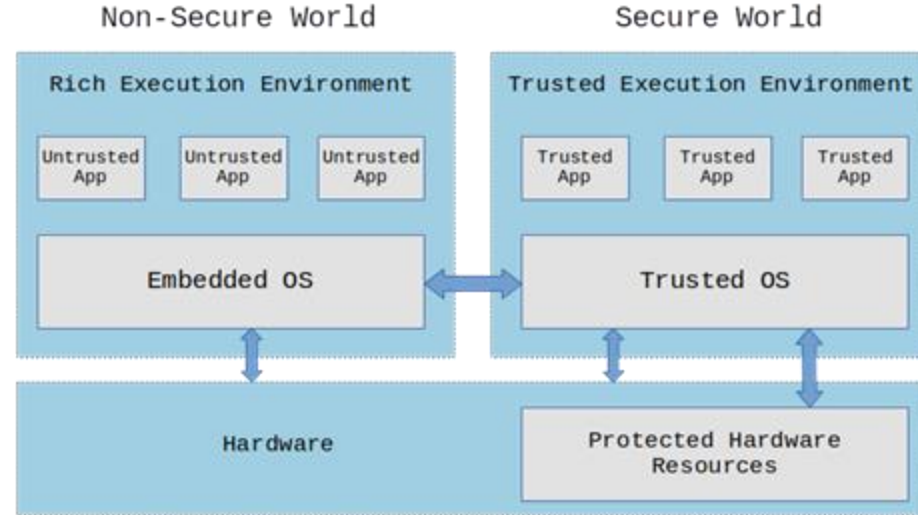
Tesseract - Overview

- Tesseract relies on a trusted execution environment (**TEE**)
- **TEE enables Tesseract to behave like a trusted third party**, controlling funds without exposing them to theft while preventing frontrunning by the exchange operator.
- Ensures ***all-or-nothing settlement***

Trusted Execution Environment - TEE

TEE is a **secure area of a main processor**.

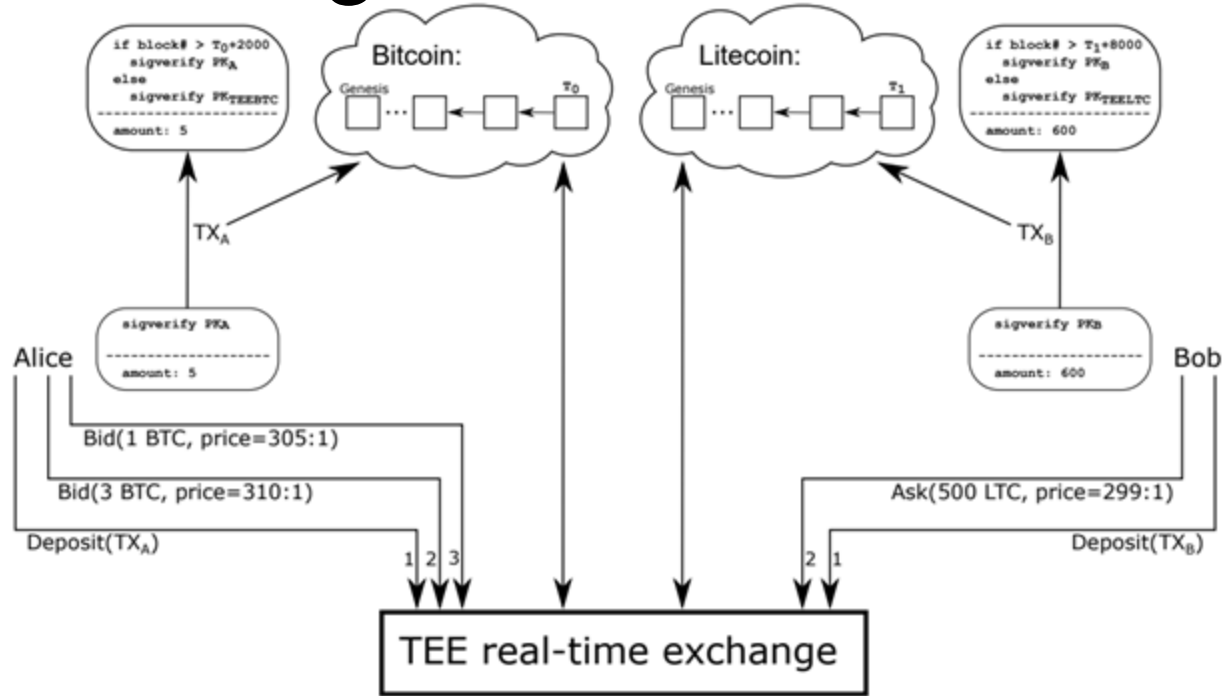
- Guarantees code and data loaded inside to be protected with respect to:
 - **Isolated execution**
 - **Integrity of applications**
 - **Confidentiality of their assets**
- **Provides attestation that an output represents the result of such an execution in the TEE.**



Tesseract - Threat Model

- Assumes the security of TEE, and *remote attestation*.
 - Code that runs inside the TEE enclave can neither be observed nor tampered with.
- Strong network adversary that can gain complete OS and network control of the host in which the funds are stored.
- Adversary's goal is to maximize her profit.

Tesseract - Design



Tesseract configured with Bitcoin and Litecoin networks.

The TEE forms the secure real-time exchange with which the parties interact to carry out asset swaps.

Tesseract - Steps

1

Initialization of exchange

1. **Tesseract enclave** is running light blockchain clients (**SPV** clients).
2. During initialization, the **enclave fetches the latest blockchain state**.
3. The enclave then invokes a **key generation** procedure to create a keypair (**sk, pk**) for each supported **cryptocurrency**.
4. The enclave will then **attest pk** as its deposit address, for each cryptocurrency.
5. The **attested pk** keys are published through multiple services - (such as websites, IPFS, and even Bitcoin and other blockchains).
6. The **pk** also forms the depositor address for Tesseract exchange in the corresponding cryptocurrency.

Tesseract - Steps

② User Initialization

1. A **new user opens a Tesseract account by depositing a significant amount** into a deposit address of the exchange.
2. The **proof of deposit is sent to the enclave** in the form of the *transaction, index of block*, and **Merkle tree**.
3. Tesseract credits the user's account (in the enclave) after verifying the deposit.
4. Tesseract also **protects against replay attacks**, by requiring strictly increasing block indices for the user's deposits.
5. Each user account is identified by the user's public key and hence the user can use its secret key to generate signed messages to interact with Tesseract.

Tesseract – Steps

③ Asset Exchange

1. Users post buy or sell orders along with their exchange rate. Orders are recorded in the enclave.
Eg: Bob bids to **sell** LTC for the price of **299 LTC per BTC**
Alice bids to **buy** LTC price of **305 LTC per BTC**.
2. Tesseract server publishes an anonymized version of the order book with remote attestation.
3. Realtime trading is executed by matching the buy and sell orders.
4. The trade results do not reflect in the blockchain networks immediately, and are only reflected in the tesseract exchange sandbox.
5. Periodically, *settlement* transactions are issued by tesseract in the blockchain networks to update users' balance.
6. Tesseract exchange can be configured to collect a proportional fee for each successful trade (e.g 0.1% of amount). This will help in exchange owners to maintain their infrastructure.

Tesseract - Eclipse Attacks

Resistance against Eclipse Attacks is the most interesting aspect of Tesseract.

- Tesseract enclave is **preloaded with the current difficulty parameter** of each PoW-based blockchain. This can be further verified by users before using the Tesseract exchange.
- Assume adversary **A controls $< \frac{1}{2}$ of computational power.**
- **A has total control over the network**

Eclipse Attack:

- **A** cuts the enclave off from the Bitcoin network and presents it with a fake blockchain containing a deposit transaction TX_{fake}
- Tesseract monitors the rate at which blocks are being mined in the network. (e.g. ~10 minutes for bitcoin)
- Fixed **difficulty parameter** -> **A** cannot change the difficulty, and forced to mine at the current difficulty.
- **A** cannot mine at the required rate (since **A** controls $< \frac{1}{2}$ of computational power of main network).
- TEE has a trusted clock. **Attack is detected by Tesseract.**