NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Shamik Sural
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 46: Blockchain Interoperability - I

CONCEPTS COVERED

- Basic Concept of Asset and Data Transfer
- Asset Transfer in Permissionless Blockchain
- Cross Chain Transfer and Exchange of Asset
- Trusted Third Party

KEYWORDS

- Interoperability
- Asset Transfer
- Cross Chain Transfer
- Trusted Third Party
- Asset Exchange

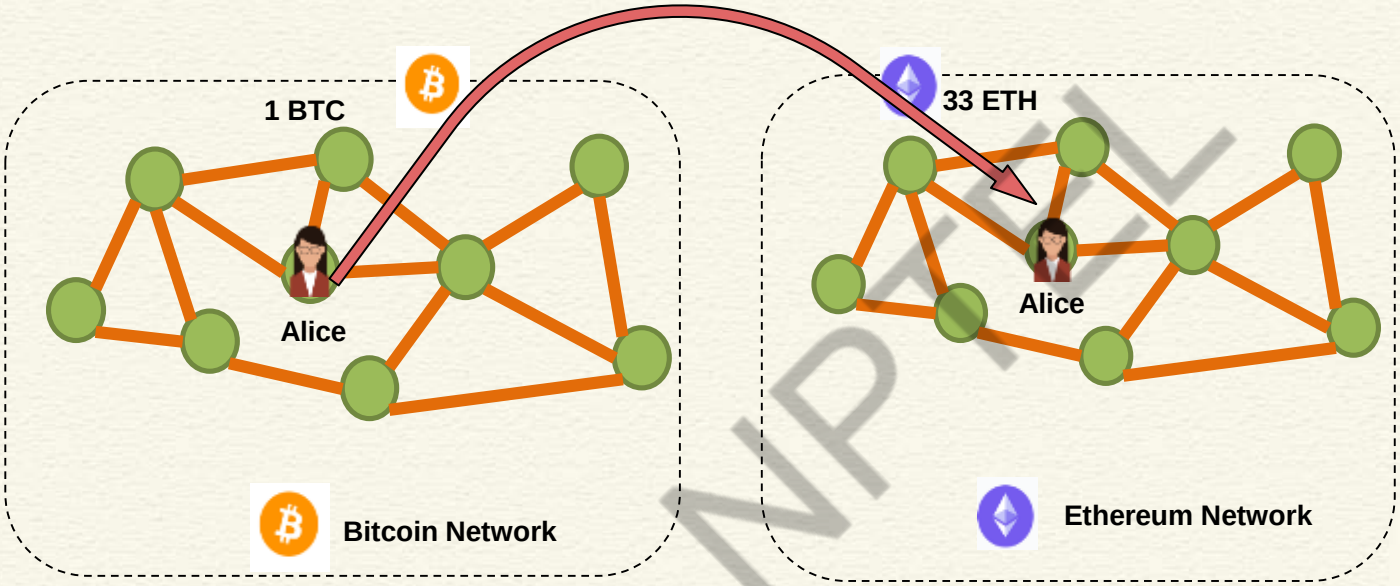
Interoperability in Permissionless and Permissioned Blockchains

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

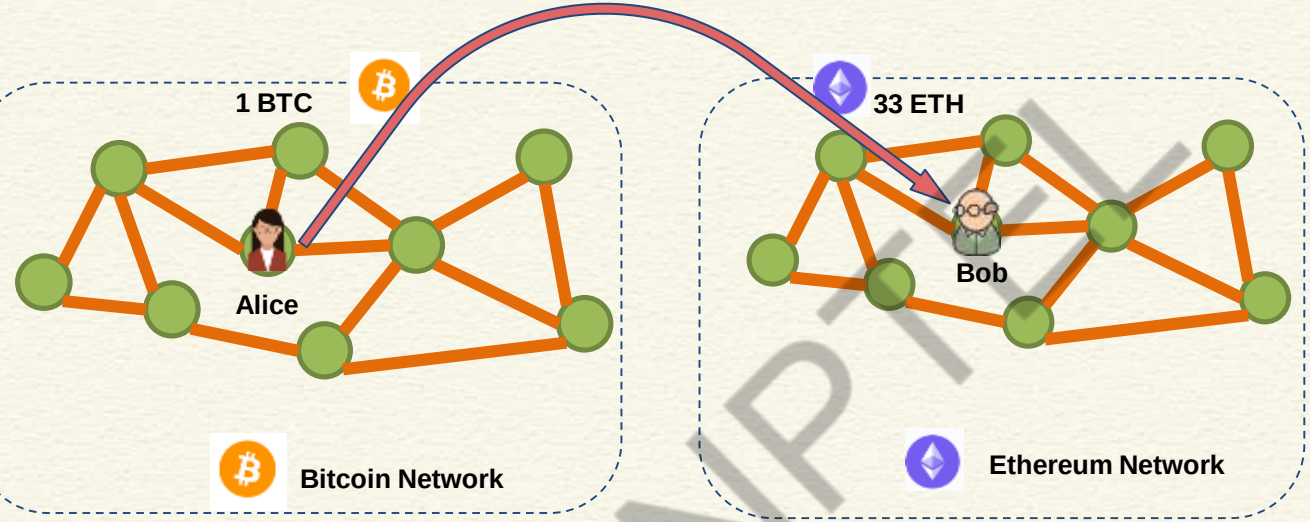
Public Blockchains as Isolated Silos

- 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

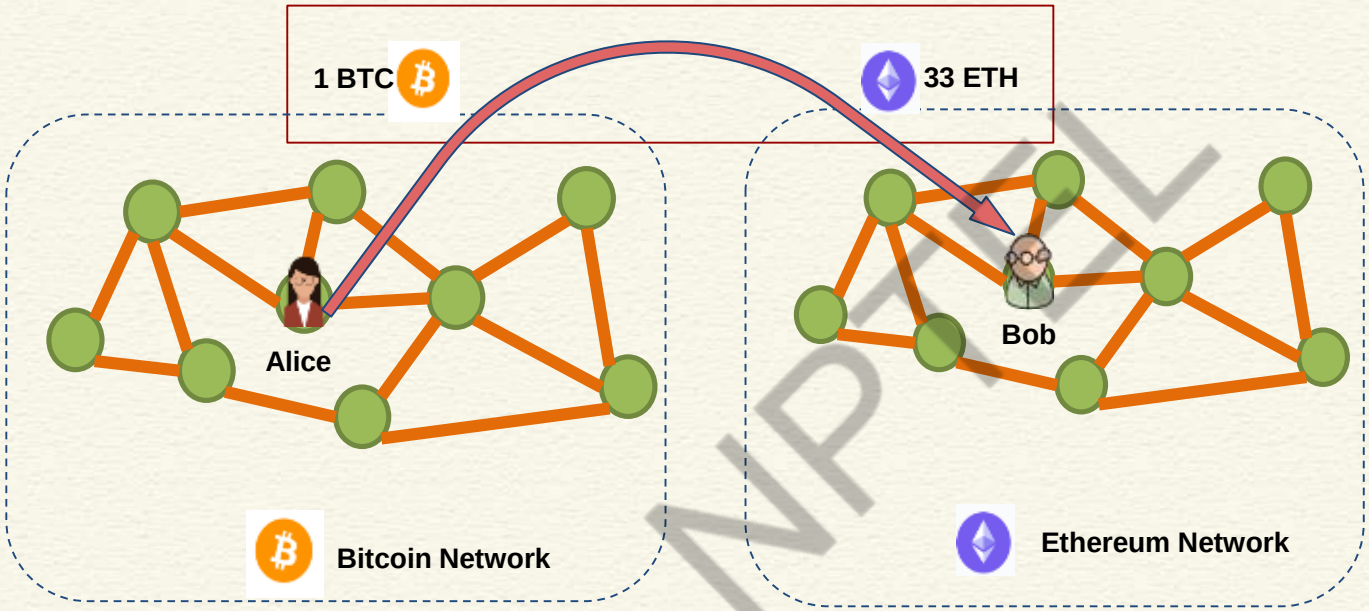


Cross Chain Asset Transfer



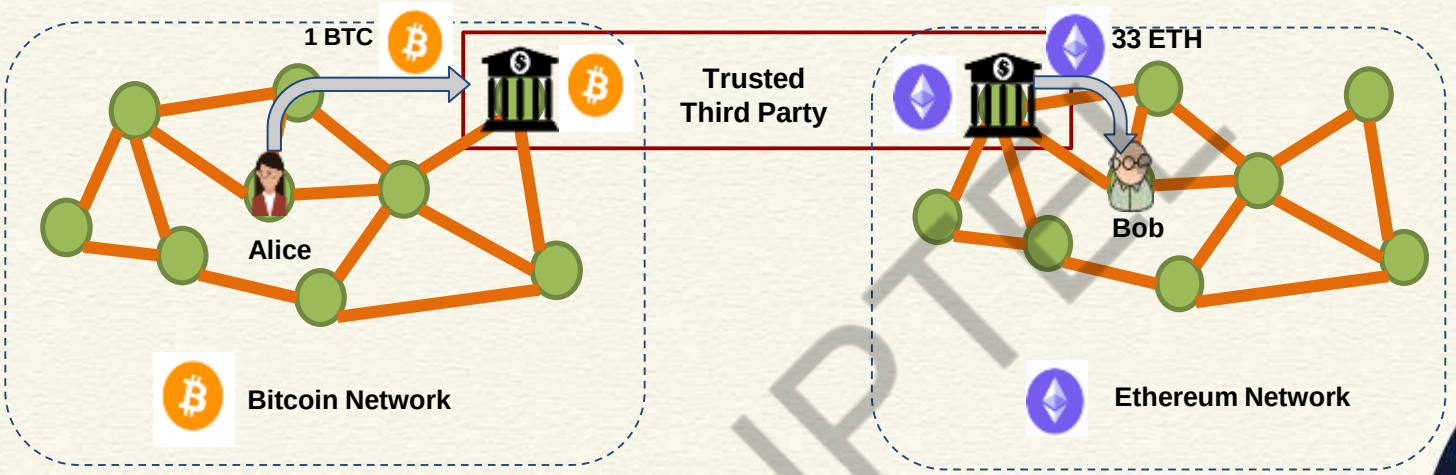
Possible between different account holders also

Cross Chain Asset Transfer



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

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).

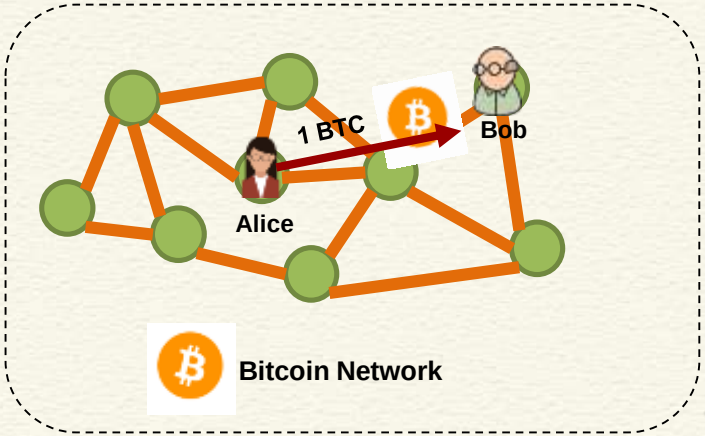
https://bitcointalk.org/index.php?topic=576337
https://www.reuters.com/article/us-bitcoin-mtgox-wallet-idUSBREA2K05N20140321
https://www.reuters.com/article/us-bitfinex-hacked-hongkong-idUSKCN10E0KP

TTP based Asset Transfer

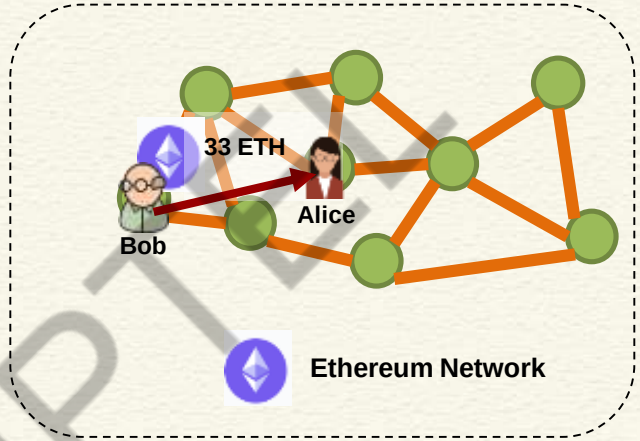
- **Lack of security:** There has been **numerous cases of theft** from centralized exchanges.
- **650,000 bitcoins lost** when the **MtGox** exchange shut down in 2014.
- Users of the Bitfinex exchange lost approximately **120,000 bitcoins** in 2016

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.

CONCLUSIONS

- Introduced the basic concepts of interoperability
- Asset transfer in permissionless blockchains
- Trusted third party based asset transfer
- Asset exchange and its challenges

REFERENCES

- **Web resources and research papers as mentioned from time to time**

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Shamik Sural
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 47: Blockchain Interoperability - II

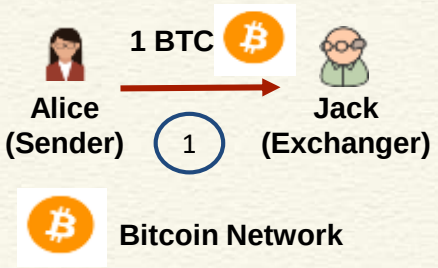
CONCEPTS COVERED

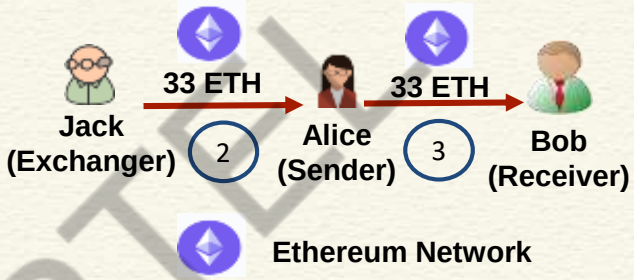
- Cross Chain Asset Exchange
- Atomic Swap
- Hashlock and Timelock
- Atomic Exchange

KEYWORDS

- Atomic Exchange
- Hashlock and Timelock
- Hashed Timelock Contract (HTLC)
- Two-party Atomic Exchange

Cross Chain Asset Transfer using Atomic Exchange





1 2 **Atomic Exchange**

3 **Transfer**

Solving atomic exchange will solve most challenges of asset transfer.

Atomic Cross-chain Swaps (PODC ‘18)

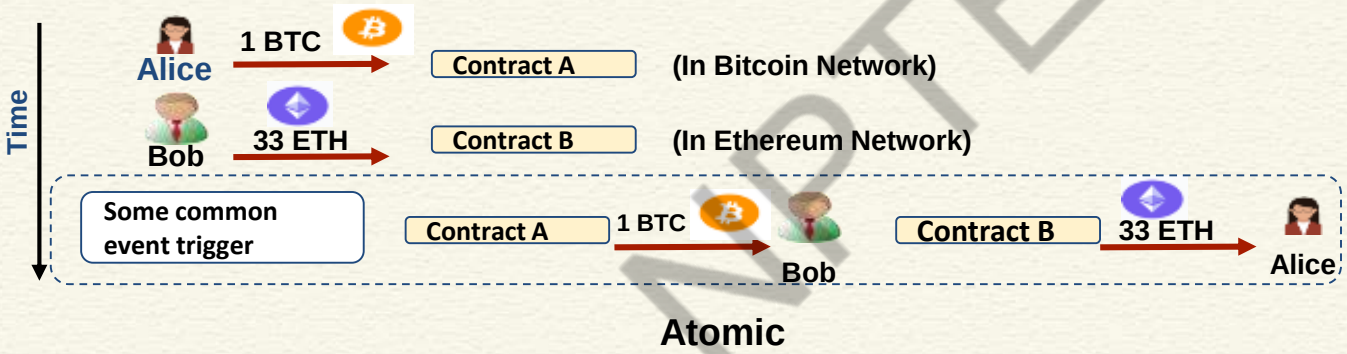
Atomicity: An atomic transaction is an indivisible series of operations, such that either all occur, or none occurs.

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: 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

Hash Locks

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

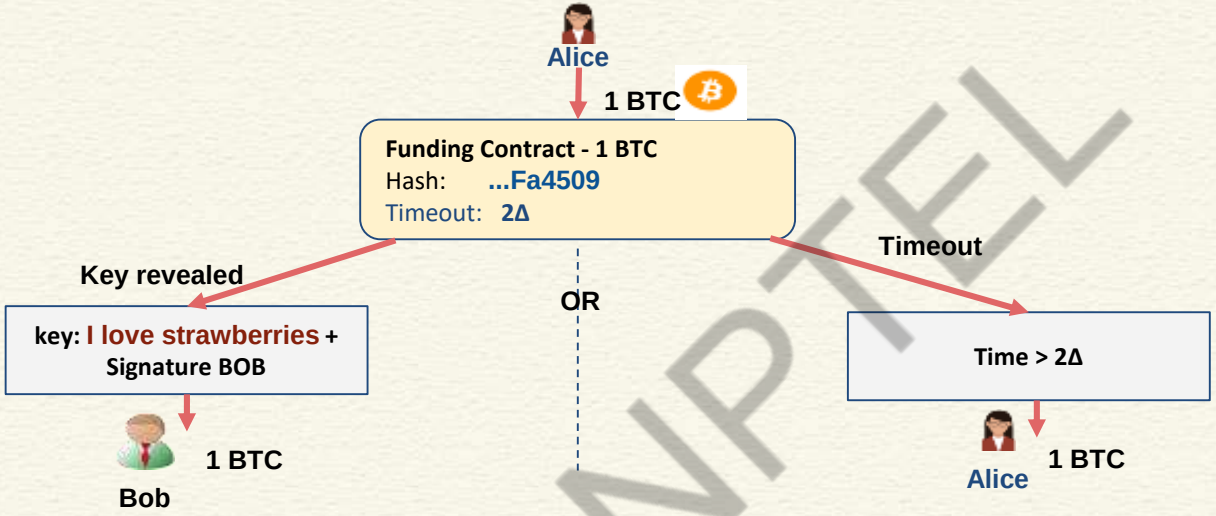
- **Timelock** is a type of smart contract primitive that restricts the spending/transfer of some currency until a specified future time
- Block height may be used as a proxy for time

Example:

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

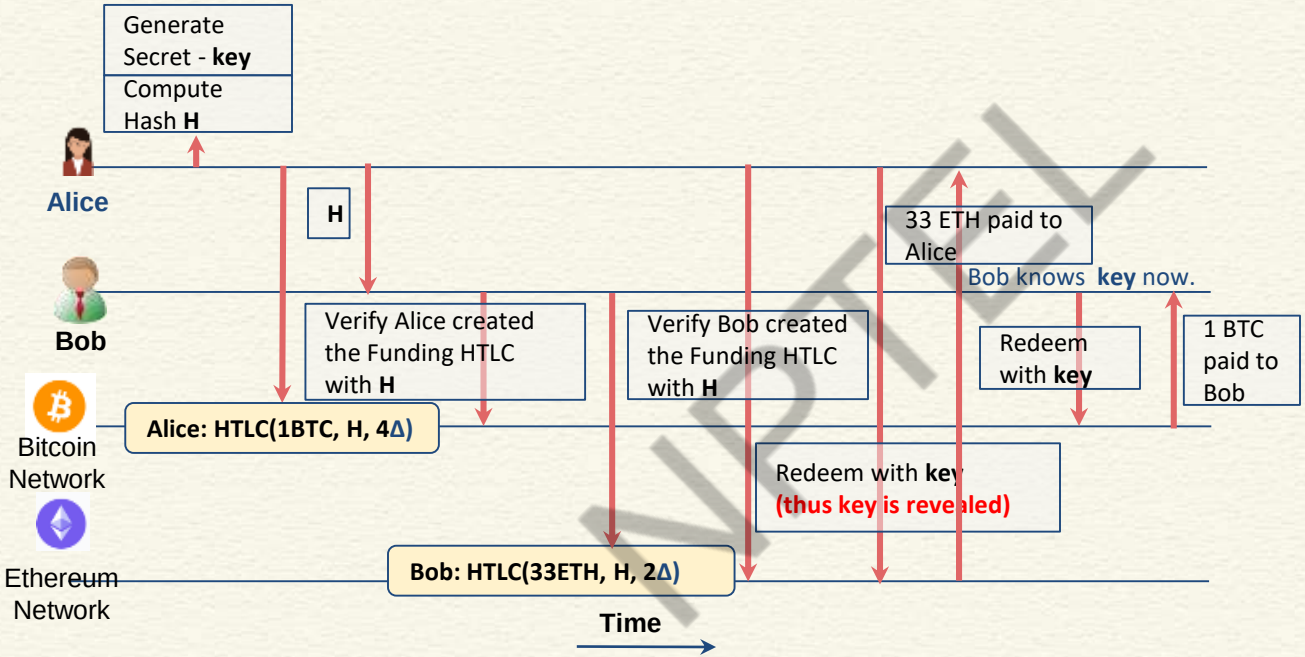
- After 2Δ time, 1 BTC will be transferred to a **target account**. (Target account can be Alice's own account)

HTLC - Hashed Timelock Contract

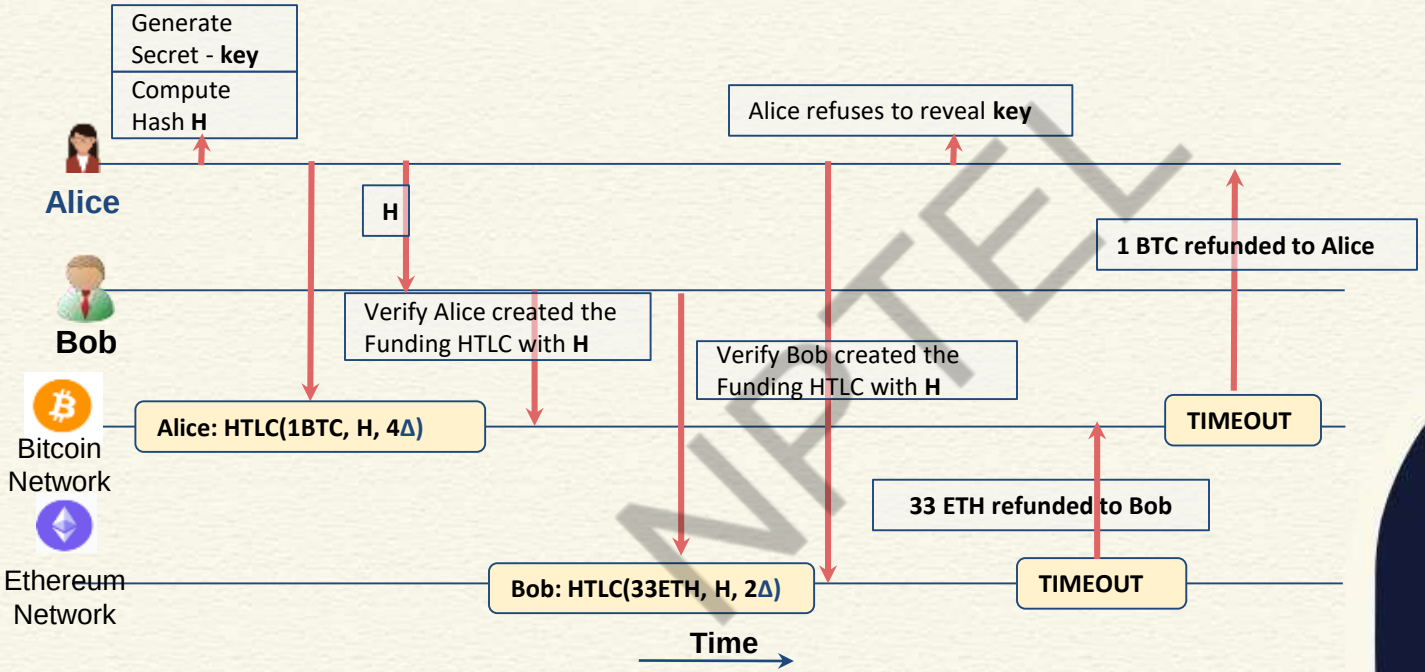


Poon, Joseph, and Thaddeus Dryja. "The bitcoin lightning network: Scalable off-chain instant payments." (2016).

HTLC for Atomic Swap



What if Alice does not Reveal Key?



CONCLUSIONS

- Explained how hashed timelock contracts work
- Cross-chain atomic swap operations
- Two-party atomic exchange

REFERENCES

- **Web resources as mentioned from time to time**

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Shamik Sural
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 48: Blockchain Interoperability - III

CONCEPTS COVERED

- Multi-party Cross-chain Swap
- Permissioned Blockchain Interoperability
- Data Transfer Across Multiple Hyperledger Fabric Networks in Two Verticals

KEYWORDS

- Cross Chain Swap
- Permissioned Blockchain Interoperability
- Interconnection Relay

Multi-Party Atomic Cross-chain Swap

- Carol wants to sell her Cadillac for bitcoins
- Alice can buy Carol's Cadillac, but wants to pay in an "alt-coin" cryptocurrency
- Bob ready to trade alt-coins for bitcoins
- Alice, Bob and Carol 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

M. Herlihy, "Atomic cross-chain swaps," in PODC, 2018

Multi-Party Atomic Cross-chain Swap

- Alice creates a secret s, h = H(s)
- Publishes a contract on the alt-coin blockchain with hashlock h and timelock 6Δ in the future, to transfer her alt-coins to Bob
- Bob first confirms that Alice's contract has been published on the alt-coin blockchain
- He then 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

M. Herlihy, "Atomic cross-chain swaps," in PODC, 2018

Multi-Party Atomic Cross-chain Swap

- Carol confirms Bob's contract is published on Bitcoin
- She publishes a contract on the automobile title blockchain with the same hashlock h, but with timeout 4Δ to transfer the Cadillac's title to Alice
- 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 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

M. Herlihy, "Atomic cross-chain swaps," in PODC, 2018

Multi-Party Atomic Cross-chain Swap

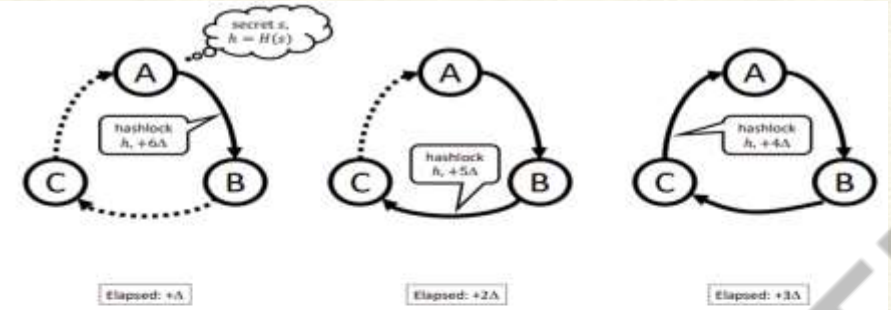


Figure 1: Atomic cross-chain swap: deploying contracts

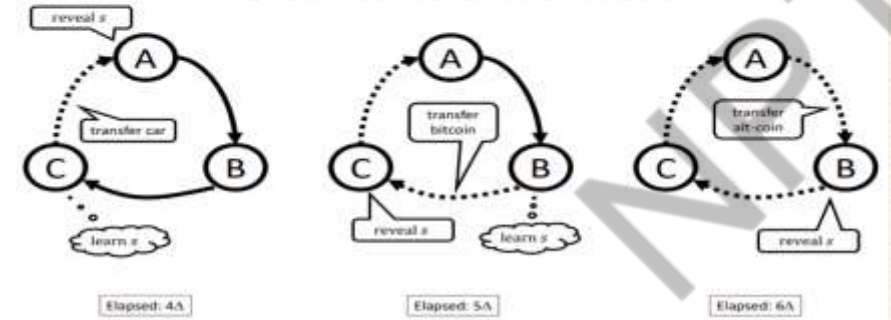


Figure 2: Atomic cross-chain swap: triggering arcs

M. Herlihy, “Atomic cross-chain swaps,” in PODC, 2018.

Validity of the Protocool

- 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

M. Herlihy, "Atomic cross-chain swaps," in PODC, 2018

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**
- **TradeLens -** Logistics, **We.Trade** Trade finance, **IBM Food Trust** - Food Supply Chain, etc.
- **Continue to operate in complete isolation**
- N**eed for interoperation** between different isolated networks to achieve business goals

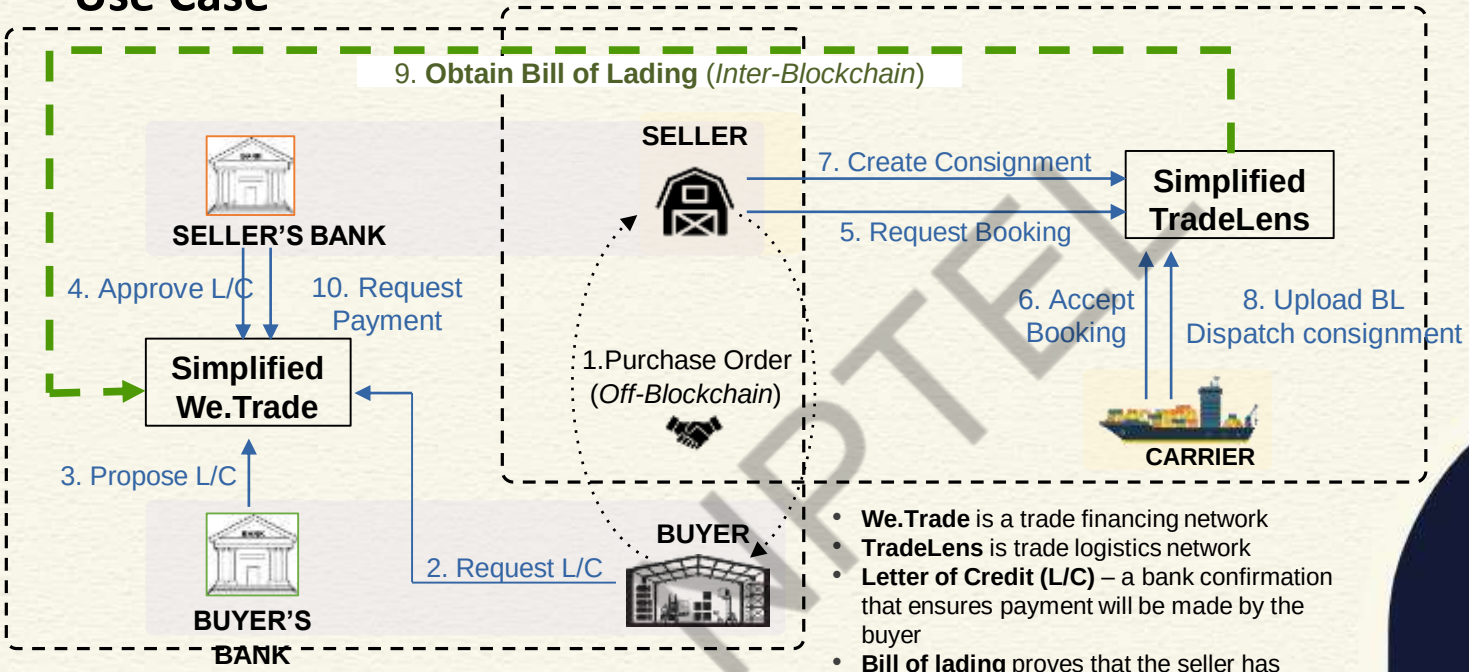
Abebe, E. et al., 2019, December. Enabling enterprise blockchain interoperability with trusted data transfer (industry track). In Proceedings of the 20th International Middleware Conference Industrial Track

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 **consistent with the source network's state**
- **Multiparty Trust -** When one network consumes state from another, it needs to establish the state validity as per shared consensus view of parties in the network

Use Case



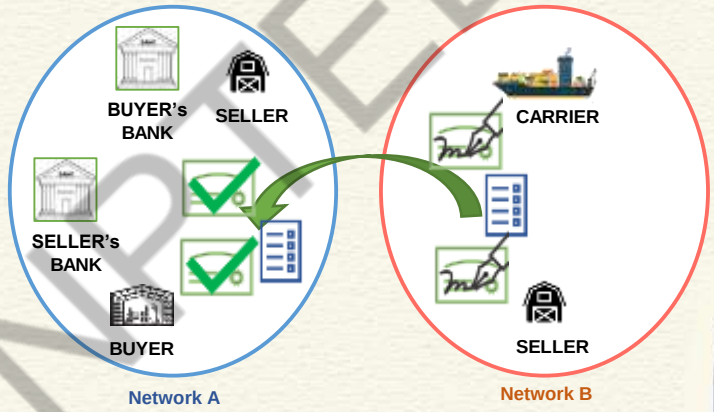
- **We.Trade** is a trade financing network
- **TradeLens** is trade logistics network
- **Letter of Credit (L/C)** – a bank confirmation that ensures payment will be made by the buyer
- **Bill of lading** proves that the seller has dispatched the goods via the carrier,
- It enforces an obligation on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

Abebe, E. et al., 2019, December. Enabling enterprise blockchain interoperability with trusted data transfer (industry track). In Proceedings of the 20th International Middleware Conference Industrial Track

Verifiable Data Transfer in Permissioned Blockchain

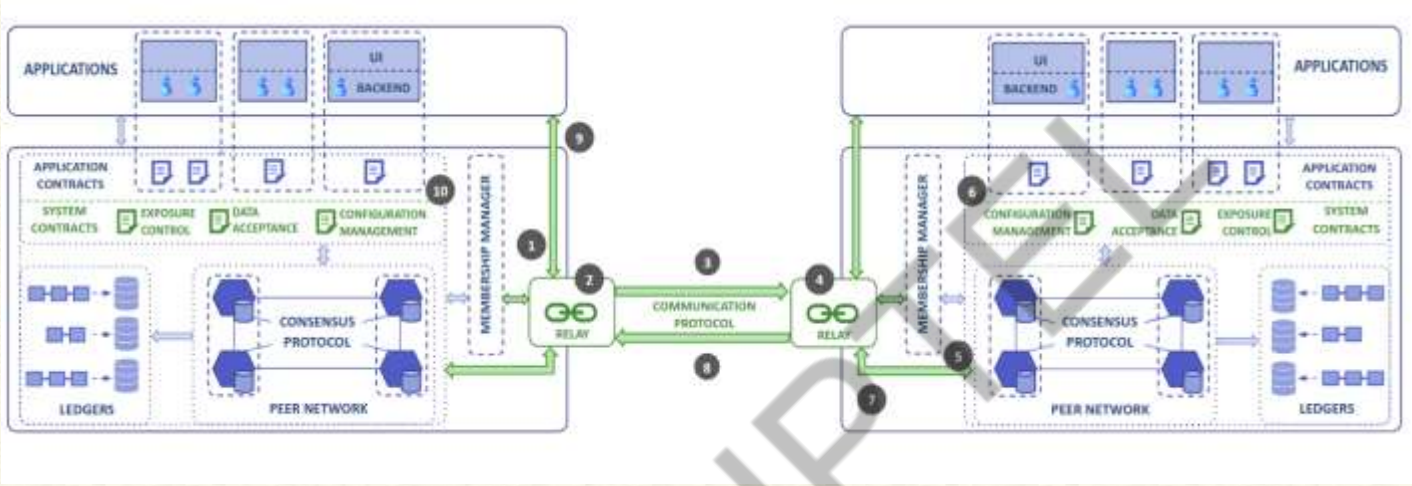
- Each data (block) in a network has a set of **endorsements** (signatures) **for consensus**
- This set of endorsements confirm the validity of a block in the network
- Thus, from the source, **data is accompanied with the set of signatures - Attestations**
- The attestations are **validated** in the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

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



Abebe, E. et al., 2019, December. Enabling enterprise blockchain interoperability with trusted data transfer (industry track). In Proceedings of the 20th International Middleware Conference Industrial Track

Architecture



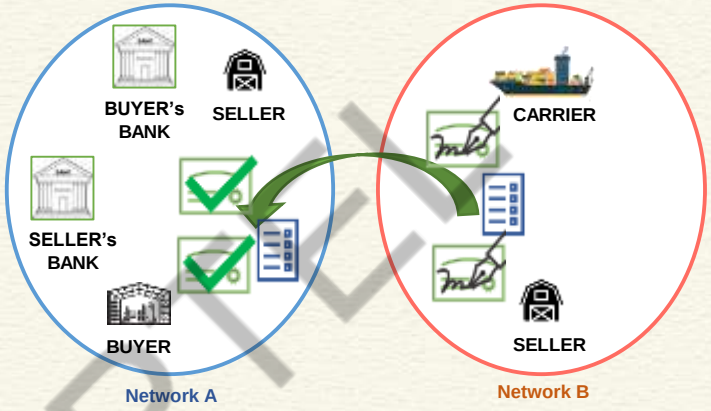
Abebe, E. et al., 2019, December. Enabling enterprise blockchain interoperability with trusted data transfer (industry track). In Proceedings of the 20th International Middleware Conference Industrial Track

- **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

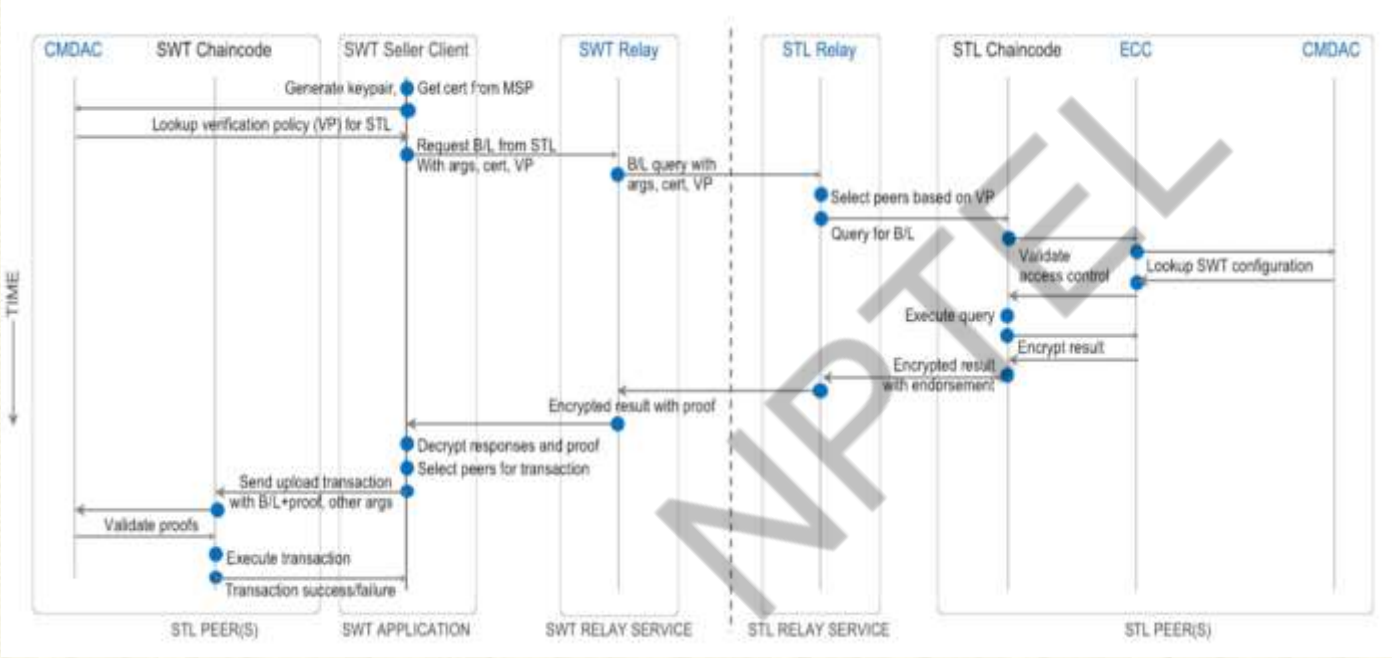
Steps:

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) sent back through relay
4. Proofs validated against verification policy



Abebe, E. et al., 2019, December. Enabling enterprise blockchain interoperability with trusted data transfer (industry track). In Proceedings of the 20th International Middleware Conference Industrial Track

Protocol Details



Abebe, E. et al., 2019, December. Enabling enterprise blockchain interoperability with trusted data transfer (industry track). In Proceedings of the 20th International Middleware Conference Industrial Track

CONCLUSIONS

- Explained how HTLC is used for three-party swap
- Permissioned blockchain interoperability
- Data transfer across different Hyperledger Fabric networks

REFERENCES

- **Web resources as mentioned from time to time**

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Bishakh Chandra Ghosh

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 49: Hyperledger Indy 1

CONCEPTS COVERED

- Hyperledger Indy Overview
- DIDs in Indy
- Hands-on tutorial on Indy

KEYWORDS

- Identity
- Indy
- DIDs

Hyperledger Indy

Hyperledger Indy provides

- tools
- libraries
- reusable components

for providing **digital identities** **rooted on blockchains** so that they are interoperable across administrative domains, applications, and any other silo.

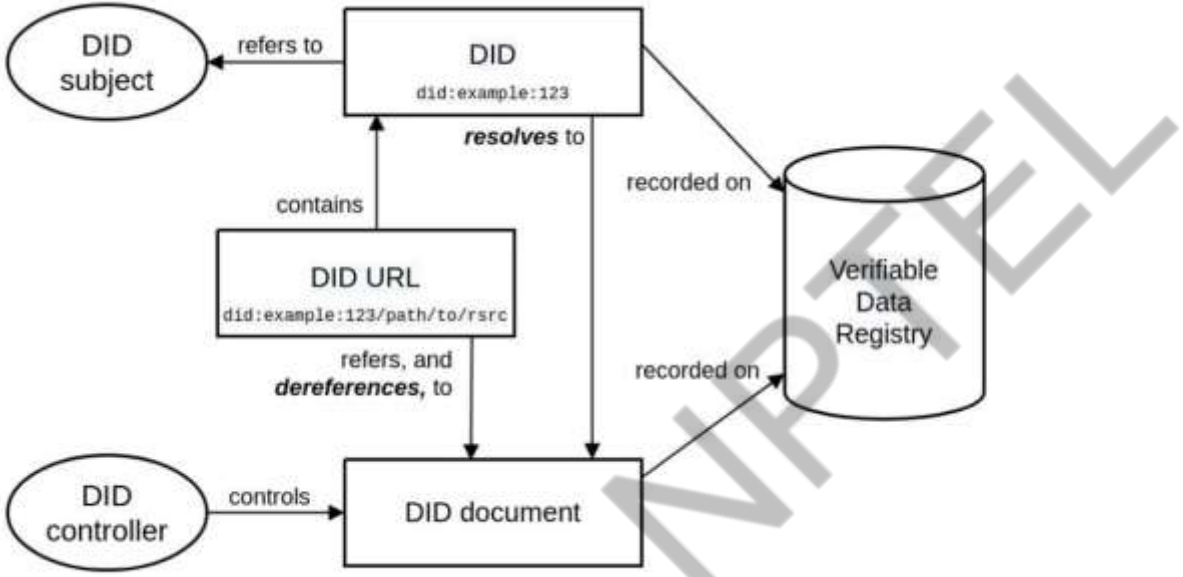
https://wiki.hyperledger.org/display/indy

Indy Key Characteristics

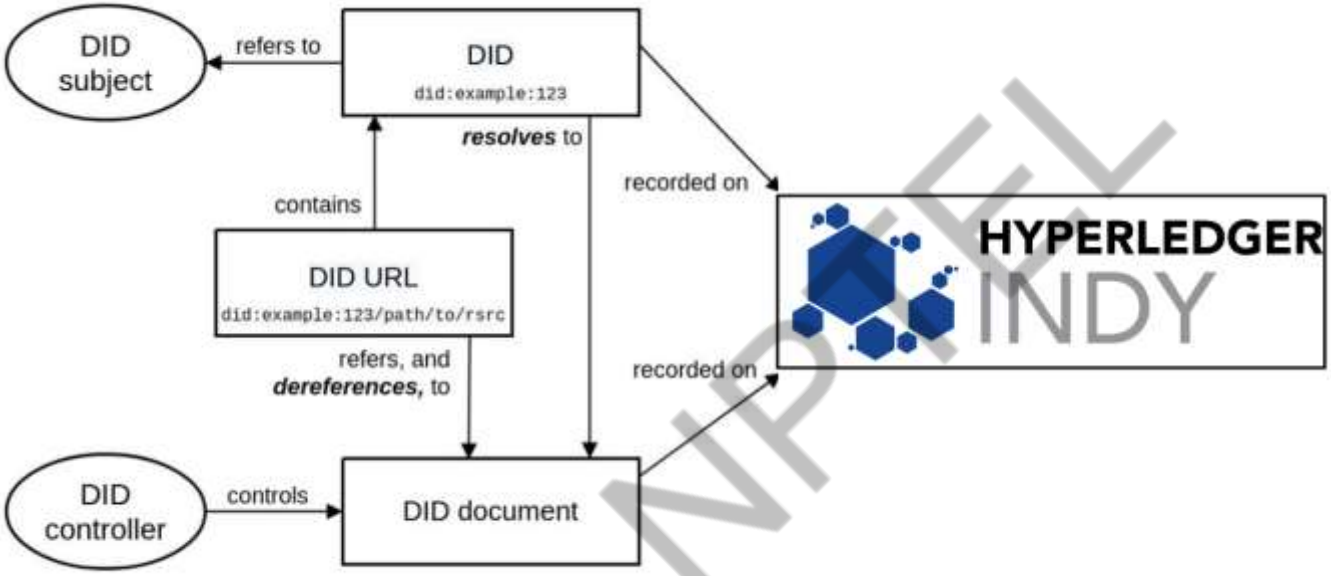
- **Distributed ledger purpose-built for decentralized identity**
- BFT by design
- DIDs that are globally unique and resolvable (via a ledger) without requiring any centralized resolution authority
- Verifiable Credentials in an interoperable format
- **Zero Knowledge Proofs** for Verifiable Presentations, which prove that some or all of the data in a set of Claims is true without revealing any additional information, including the identity of the Prover

https://wiki.hyperledger.org/display/indy

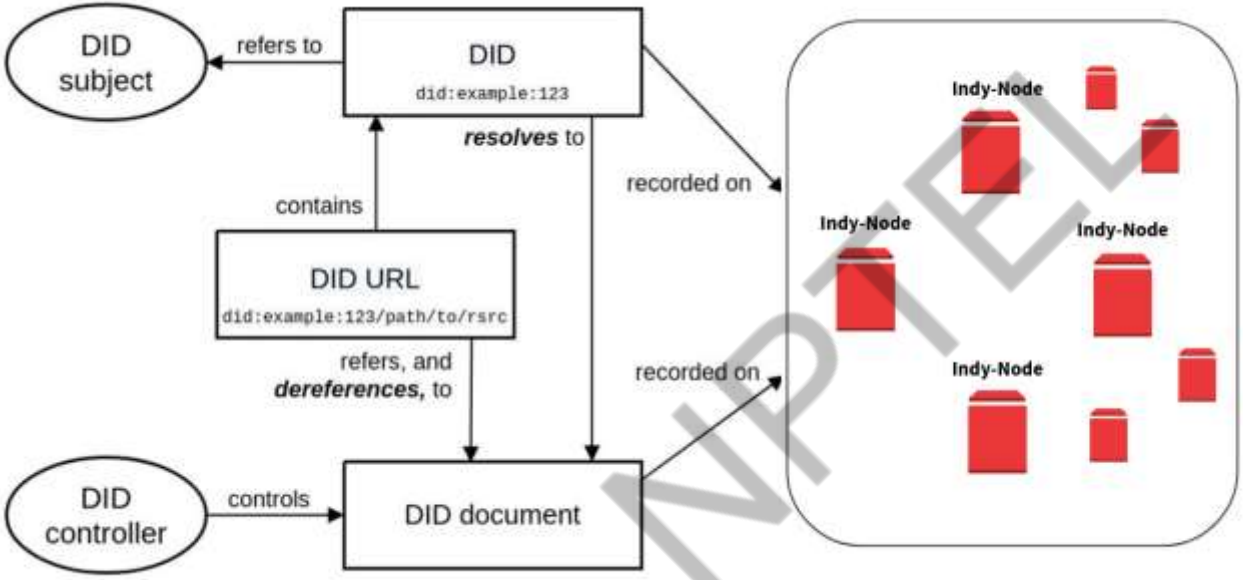
Indy Overview



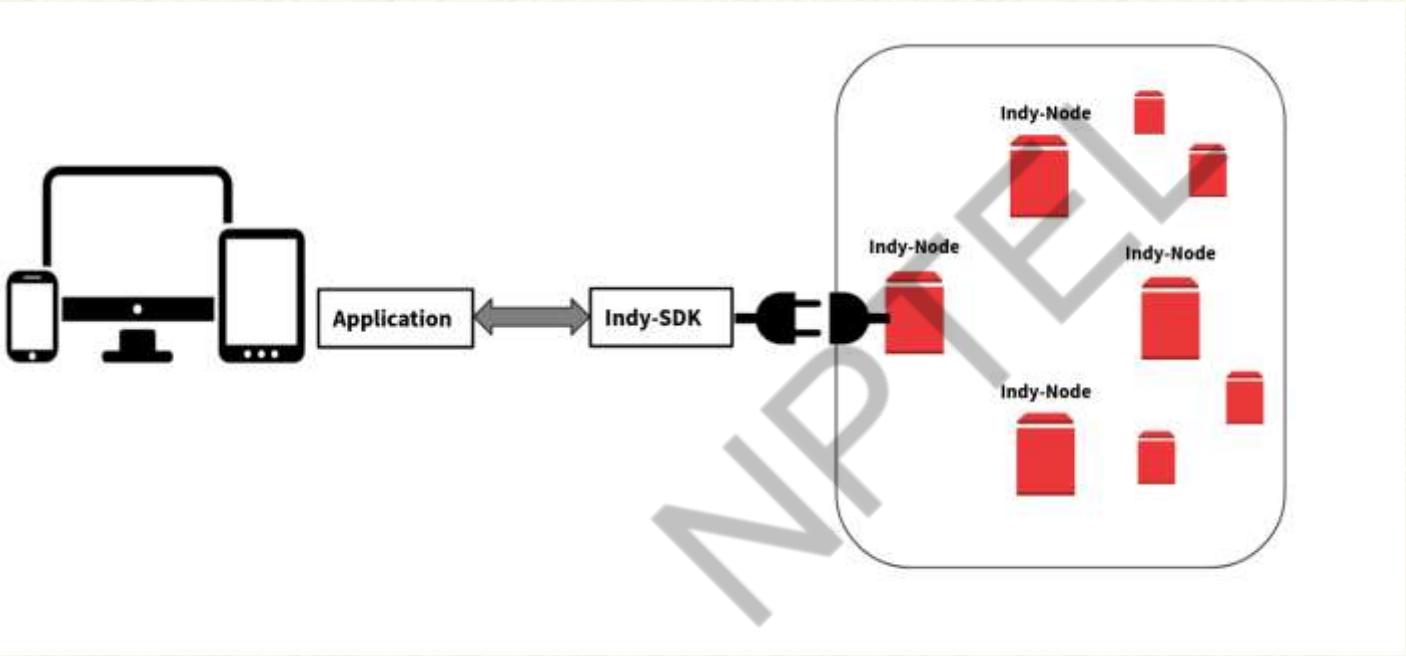
Indy Overview



Indy Overview



Indy Overview



Indy Projects

- **Indy-Plenum:**
  - Implements Byzantine Fault Tolerant Protocol
  - Used for consensus in Indy
  - Based on RBFT
  - https://github.com/Hyperledger/indy-plenum

- **Indy-Node:**
  - Implements the blockchain with Indy-Plenum consensus
  - Defines identity specific transactions.
  - https://github.com/Hyperledger/indy-node

- **Indy-SDK**
  - Provides APIs to applications for accessing Indy network
  - Indy- https://github.com/Hyperledger/indy-sdk

Install Indy – Starting an Indy Pool

Clone indy-sdk

```
git clone https://github.com/hyperledger/indy-sdk.git
cd indy-sdk
```

Build and run indy pool docker image

```
docker build -f ci/indy-pool.dockerfile -t indy_pool .
docker run -itd -p 9701-9708:9701-9708 indy_pool
```

Install Indy – Starting an Indy Pool

Easier Alternatives:

1. Starting a pre-configured docker image:

```
docker run -itd -p 9701-9708:9701-9708 ghoshbishakh/indy_pool
```

2. Start from indy-node repository:

Clone indy-node

```
git clone https://github.com/hyperledger/indy-node.git
```

Move to the directory indy-node/environment/docker/pool

```
./pool_start.sh [number of nodes in pool] [IP addresses of nodes] [number of clients] [port for the first node]
```

Eg.

```
./pool_start.sh 4 10.0.0.2,10.0.0.3,10.0.0.4,10.0.0.5 10 9701
```

Install Indy SDK

Ubuntu based distributions (Ubuntu 16.04 and 18.04)

It is recommended to install the SDK packages with APT:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys CE7709D068DB5E88
sudo add-apt-repository "deb https://repo.sovrin.org/sdk/deb (xenial|bionic)
{release channel}"
sudo apt-get update
sudo apt-get install -y {library}
```

- {library} must be replaced with libindy, libnullpay, libvcx or indy-cli.
- (xenial|bionic) xenial for 16.04 Ubuntu and bionic for 18.04 Ubuntu.
- {release channel} must be replaced with master, rc or stable to define corresponded release channel. Please See the section "Release channels" above for more details.

Install Python3 Wrapper

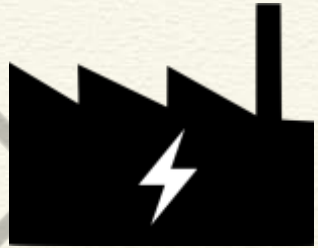
```
pip install python3-indy
```

https://github.com/hyperledger/indy-sdk

Scenario

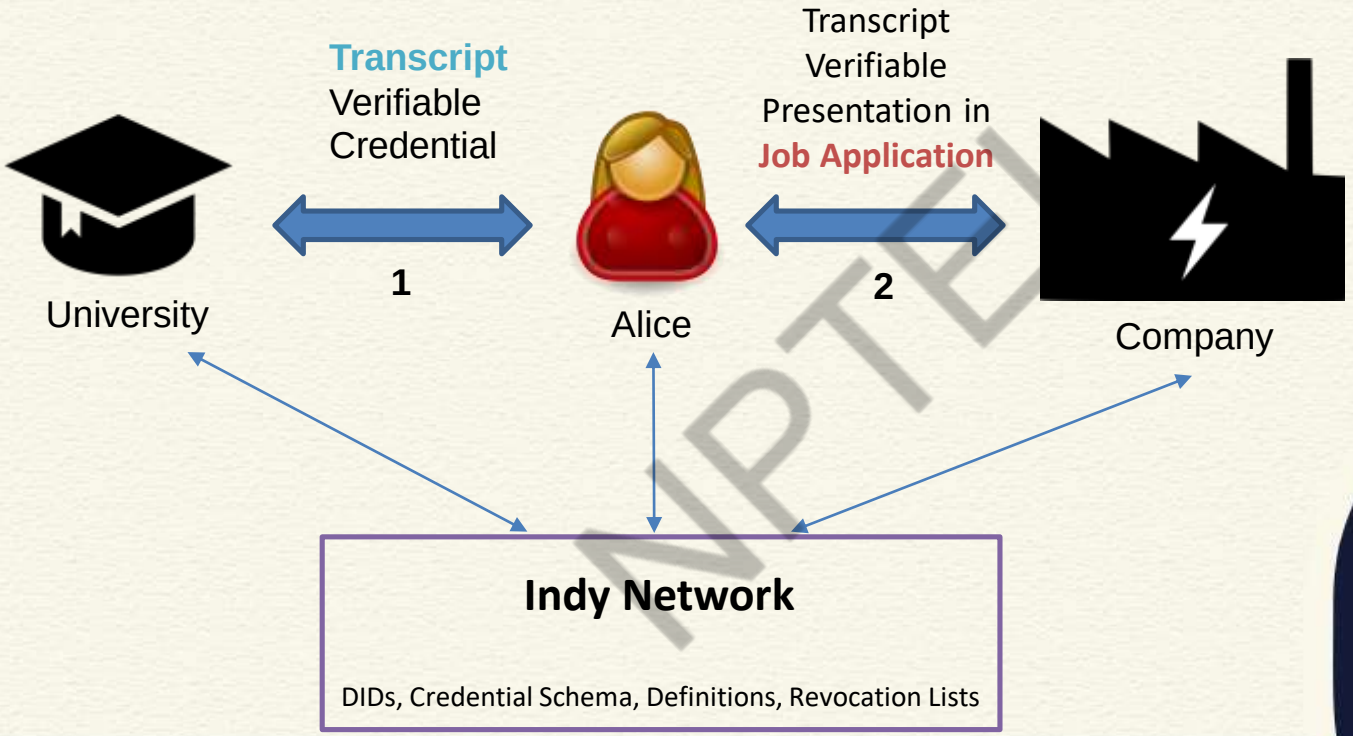
University

Alice



Company

Scenario



Configuring Identities in Indy

Roles:

STEWARDS

- **Public permissioned network**
- Only pre-approved participants, known as **stewards**, are permitted to participate in the validation process.

Trust Anchor(TA)

- Link between User and Stewards.
- E.g. banks, universities, hospitals, service providers, insurance companies.
- Onboarded by approvals of Stewards.
- Accepts the request from user and forwards this request to Stewards in case of writing into the ledger.

Configuring Identities in Indy

STEP1 - Connect to Indy Pool

- Genesis txn

STEP2 - Get ownership of Steward's DID

STEP3 - Register DID for Government, University and Company

- Nym Transactions

Conclusion

- Indy – public permissioned network
- Stewards and Trust anchors
- DID registration through Stewards

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Bishakh Chandra Ghosh

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 50: Hyperledger Indy 2

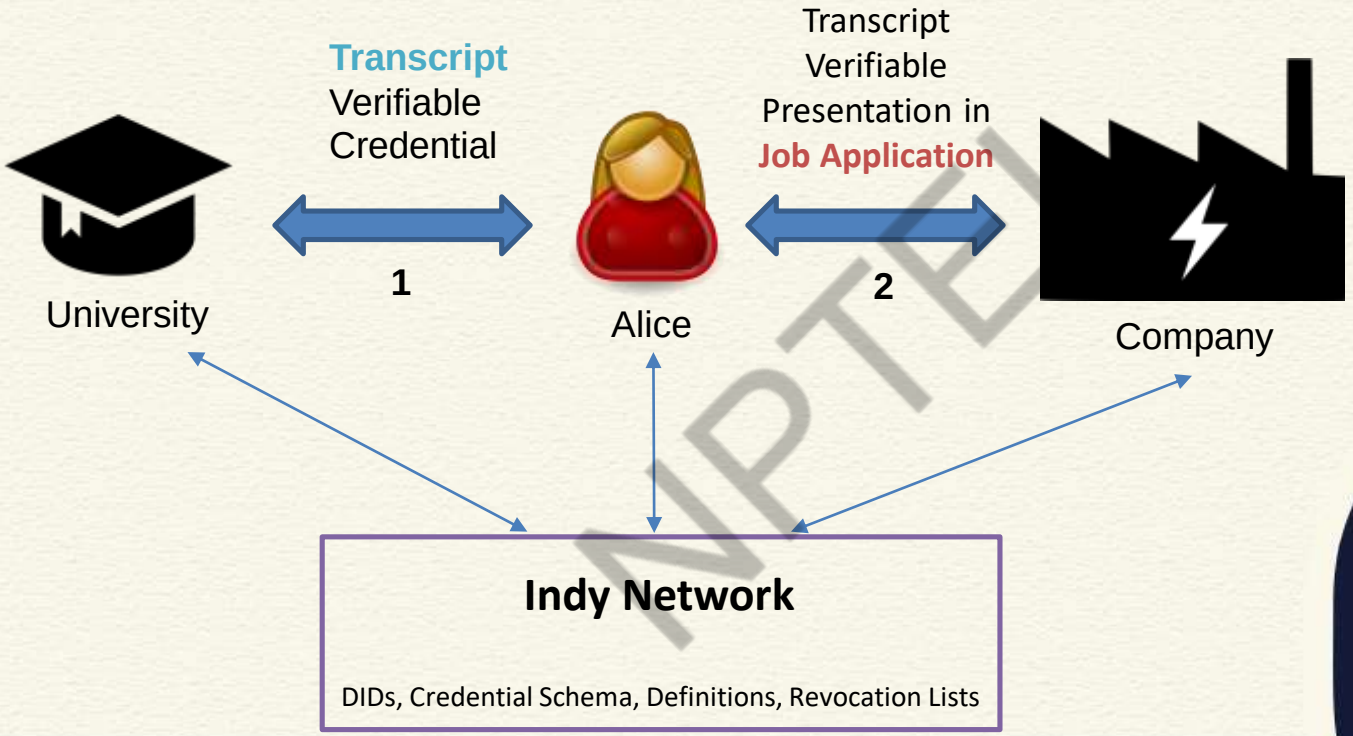
CONCEPTS COVERED

- **Indy Verifiable Credentials**
- **Presentations**

KEYWORDS

- Indy
- Verifiable Credentials
- Verifiable Presentations

Scenario



Indy Credential - Presentation Flow

STEP4 – Register Credential Schema

- Government creates transcript schema.

STEP5 – Create Credential Definition

- University creates a credential definition.

STEP6 - Issue Credential

- University issues transcript credential to Alice.

STEP7 – Verifiable Presentation

Alice sends Verifiable Presentation to Company.

STEP8 – Validate Presentation

Company validates Alice's claims from the presentation.

Source Code: https://github.com/ghoshbishakh/nptelindy

Conclusion

- Verifiable Credentials
- Verifiable Presentations
- Communication between participants