NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Sandip Chakraborty

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 41: ByzCoin

CONCEPTS COVERED

- Byzcoin: Combining PoW with PBFT
- Scalability: How far can we achieve?

KEYWORDS

- Byzcoin
- Open consensus group
- The blockchain performance triangle

Revisiting the Requirements for Blockchain Consensus

- **Byzantine fault tolerant** – the system should work even in the presence of malicious users while operating across multiple administrative domains

- Should provide **strong consistency guarantee** across replicas

- Should **scale well to increasing workloads** in terms of transactions processed per unit time

- Should **scale well to increasing network size**

Bitcoin-NG: The issue with a Faulty Key Block

- **Problem with Bitcoin-NG:** A faulty key block is verified only after end of the round
- A faulty miner can introduce several correct microblocks following a faulty microblock in the system
    - certainly an overhead for the application - **a fork alleviates the problem further**

Bitcoin-NG: The issue with a Faulty Key Block

- **Problem with Bitcoin-NG:** A faulty key block is verified only

Solve this problem by a set of PBFT verifiers - who will verify a block and then only the block is added in the Blockchain

Issues with PBFT

- PBFT requires a **static consensus group** (because of message passing)

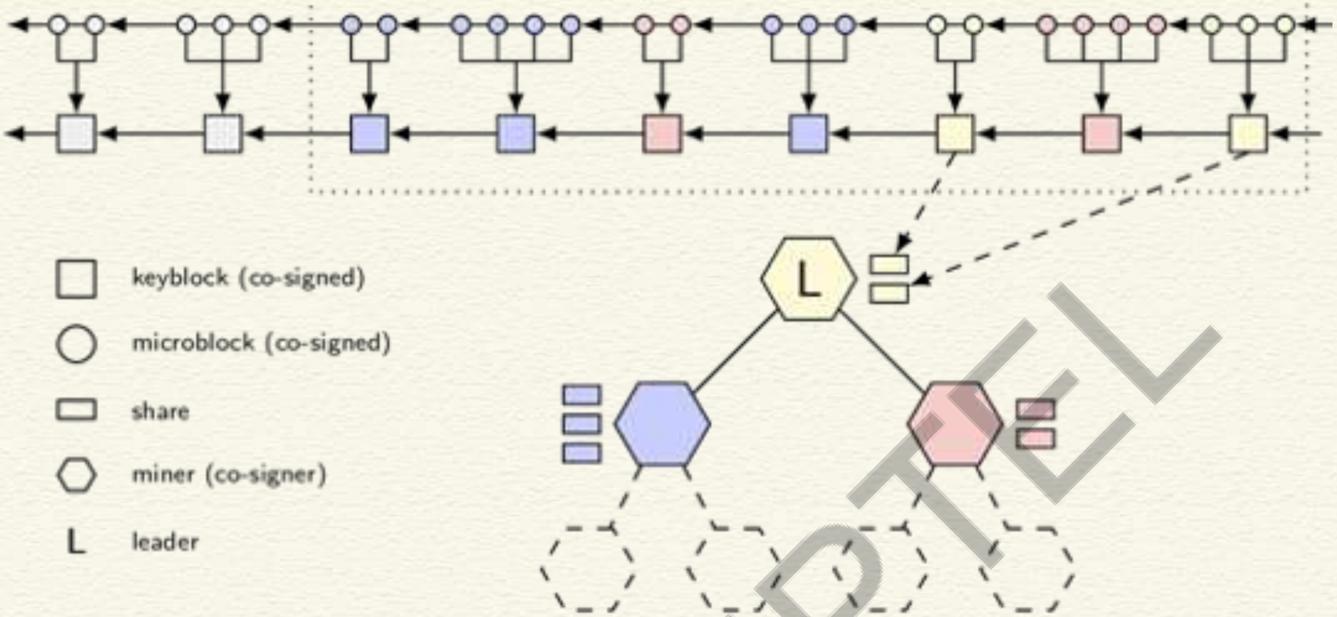
- **Scalability** (in terms of nodes) is a problem for PBFT
  - $O(n^2)$ communication complexity
  - $O(n)$ verification complexity
  - Absence of third-party verifiable proofs (PBFT uses MAC - need to share the keys among the miners)

- **Sybil attack** - create multiple pseudonymous identities to subvert the ***3f+1*** requirements of PBFT

Open the Consensus Group

- Use PoW based system to give a *proof of membership* of a miner as a part of the trustees

- Maintains a “balance of power” within the BFT consensus group
  - Use a fixed-size sliding window
  - Each time a miner finds a new block, it receives a *consensus group share*
  - The share proves the miner’s membership in the trustee group



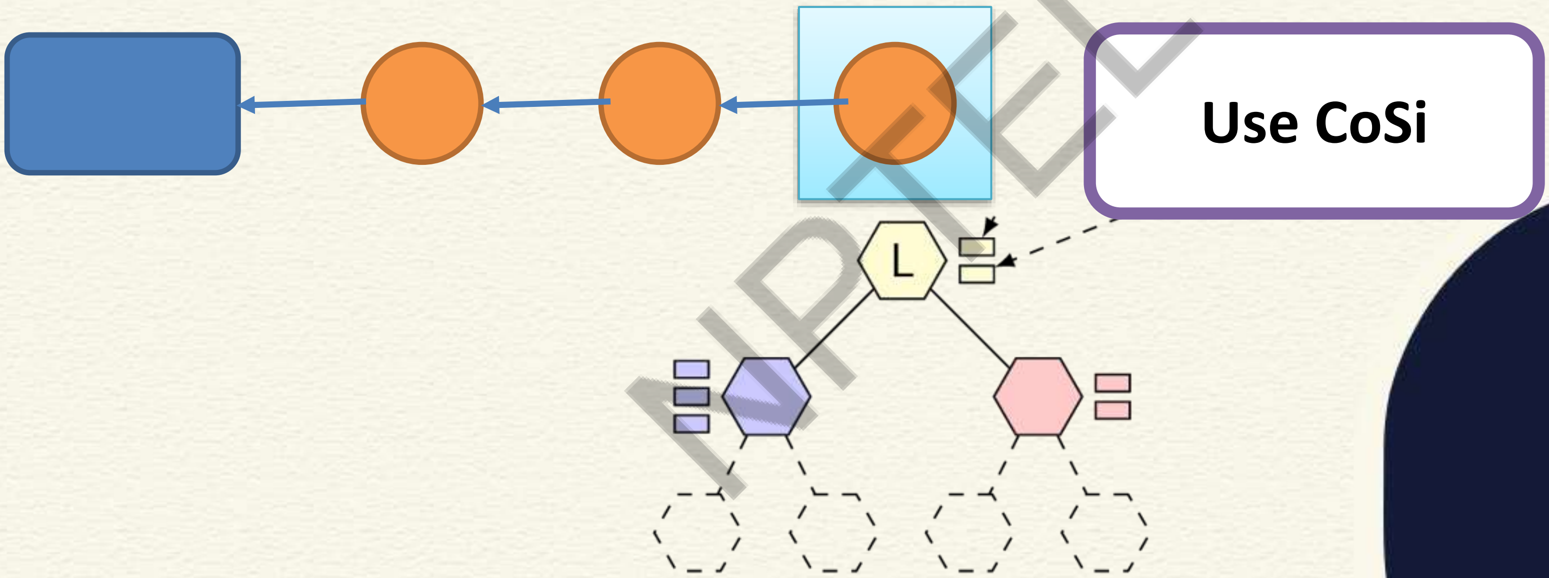
Kogias, E. K., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., & Ford, B. (2016, August). **Enhancing bitcoin security and performance with strong consistency via collective signing**. In *25th USENIX Security Symposium 2016*

Merging BFT Consensus with PoW

- Validate each microblock by a set of witness consigners

Merging BFT Consensus with PoW

- Validate each microblock by a set of witness consigners

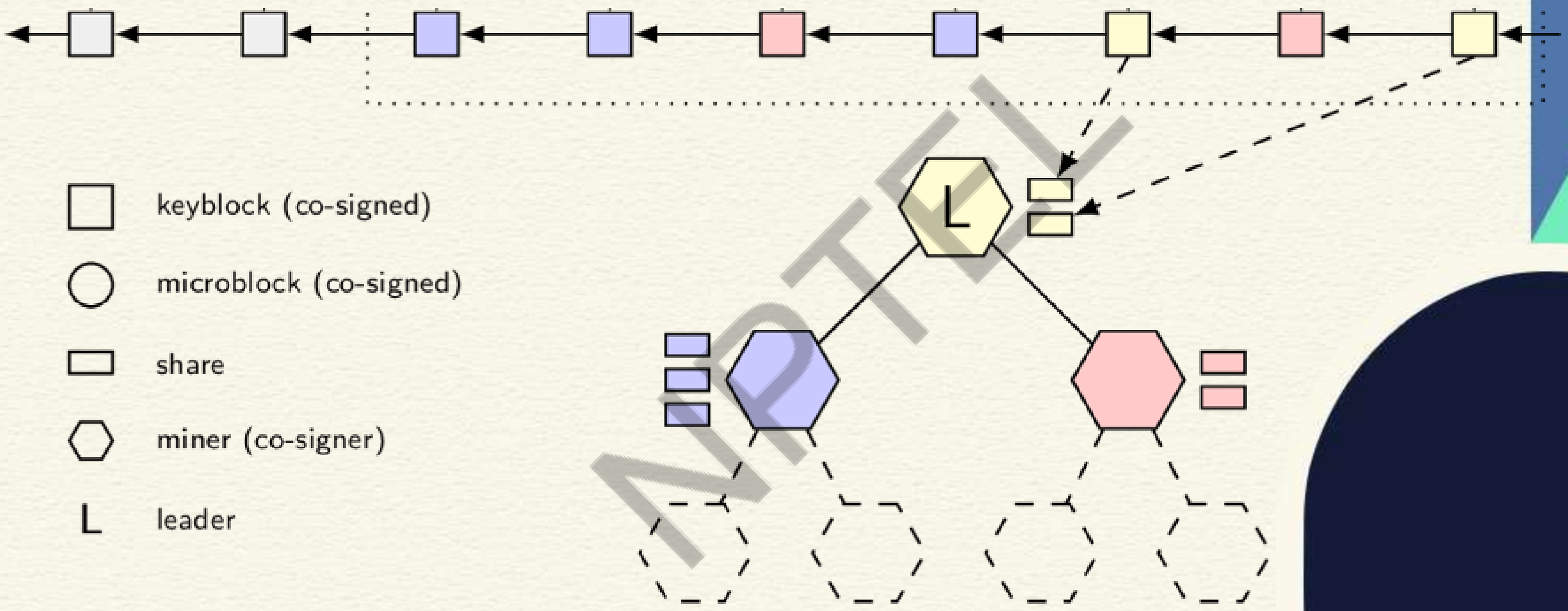


Merging BFT Consensus with PoW

- Validate each microblock by a set of witness consigners

- **How do we select the witness cosigners?**

Selecting a Consensus Group



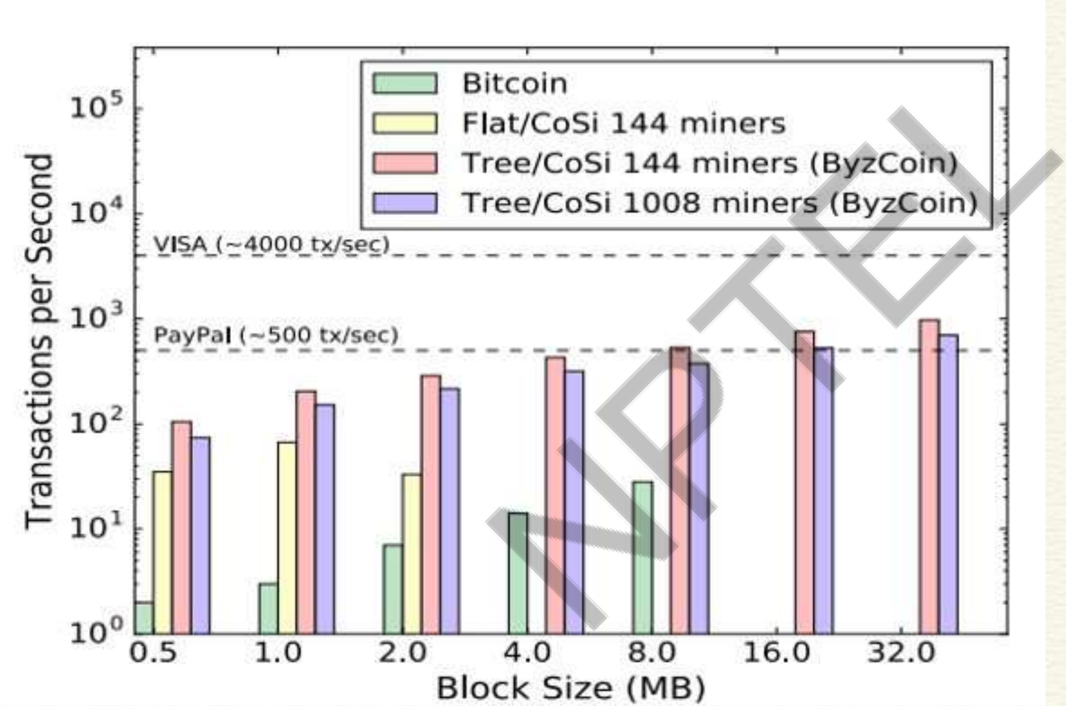
Improving Efficiency of BFT Consensus

- **Improve O(n) communication complexity**
  - Use tree-based multicast protocol - share information with O(log n)

- **Improve O(n) complexity for verification**
  - Use Schnorr multi-signatures
  - Verification can be done in O(1) through signature aggregation

- Multi-signatures + Communication trees = **CoSi**

ByzCoin Performance



ByzCoin Summary

- ByzCoin solves the problem of introducing a faulty microblocks in Bitcoin-NG

- Combine PoW with PBFT
    - Open the consensus group with the help of CoSi

ByzCoin Summary

- ByzCoin solves the problem of introducing a faulty microblocks in Bitcoin-NG
- Combine PoW with PBFT
  - Open the consensus group with the help of CoSi
- **How can we achieve Internet-scale scalability?**
  - **Both performance and network size**

Bitcoin Recap

- **Key Idea**:
  - Consensus through proof-of-work (PoW)
- **Communication**:
  - Gossip protocol
- **Key Assumption**:
  - Honest majority of mining computation power

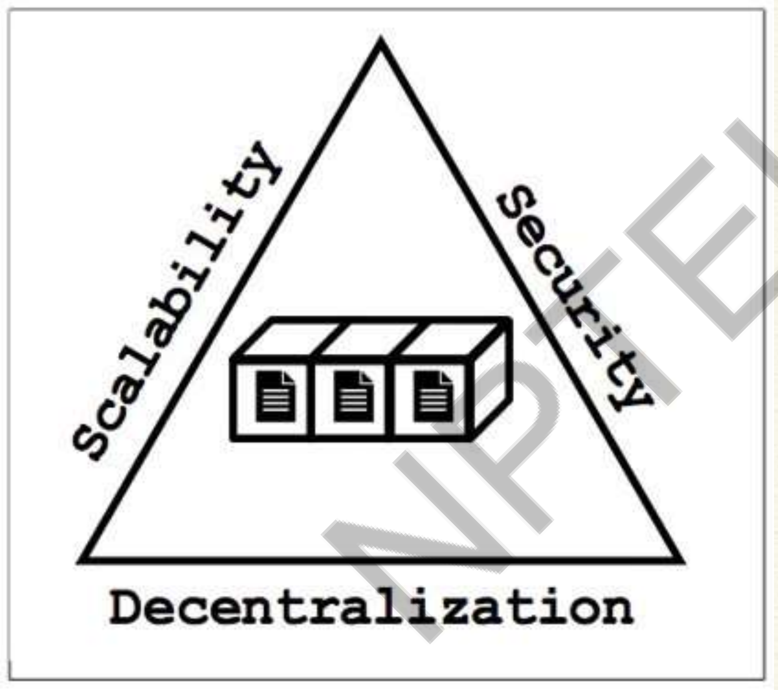
Bitcoin Limitations

- **Resource wastage**:
  - high computational, electricity cost
- **Concentration of power**
  - only ~5 mining pools control the entire system
- **Vulnerable**
  - easy to track miners, concentrated to a few mining pools - https://www.blockchain.com/btc/blocks?page=1

Bitcoin Limitations

- **Scalability**
  - number of users not clear (1M, 10M, 100M??), high latency(~10minutes)
- **Ambiguity**
  - fork in blockchain

Conclusion: The Blockchain Performance Triangle



Is it ever possible to achieve all three simultaneously?


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Sandip Chakraborty

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 42: Algorand

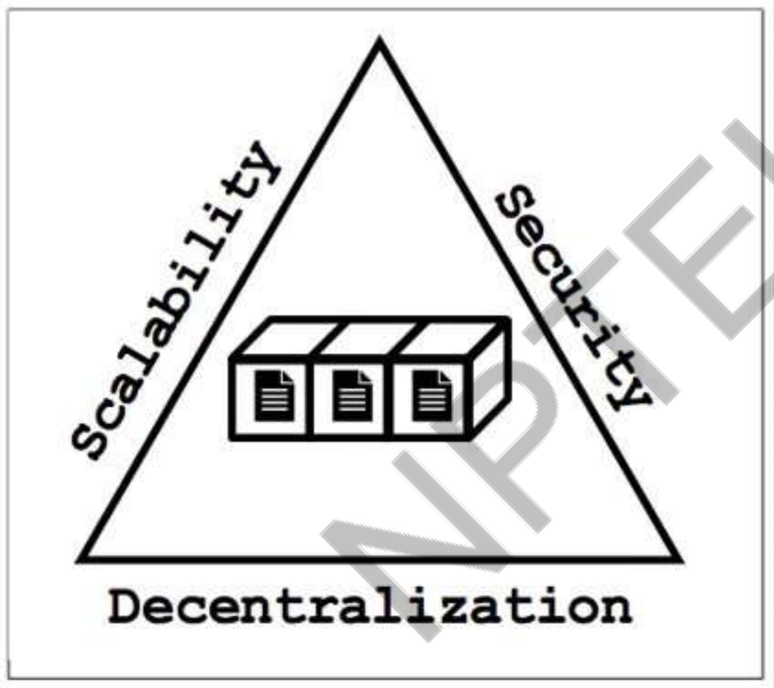
CONCEPTS COVERED

- **Algorand**

KEYWORDS

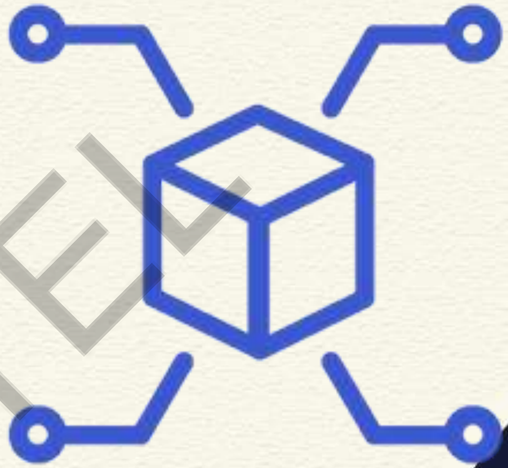
- **Cryptographic Sortition**
- **BA***

The Blockchain Performance Triangle



Is it ever possible to achieve all three simultaneously?

Algorand: Scaling Byzantine Agreements for Cryptocurrencies



Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017, October). *Algorand: Scaling byzantine agreements for cryptocurrencies*. In *Proceedings of the 26th Symposium on Operating Systems Principles* (pp. 51-68). ACM.

Algorand: Overview

- **Key Idea**:
  - Consensus through Byzantine Agreement Protocol
- **Communication**:
  - Gossip protocol
- **Key Assumption**:
  - Honest majority of money

Algorand: Technical Advancement

- **Trivial computation**
  - simple operation like add, count
- **True decentralization**
  - no concentration of mining pool power, all equal miners and users
- **Finality of payment**
  - fork with very low probability, block appears, and the payment is fixed forever

Algorand: Technical Advancement

- **Scalability**
  - millions of users, only network latency (~1minute)
- **Security**
  - against bad adversary

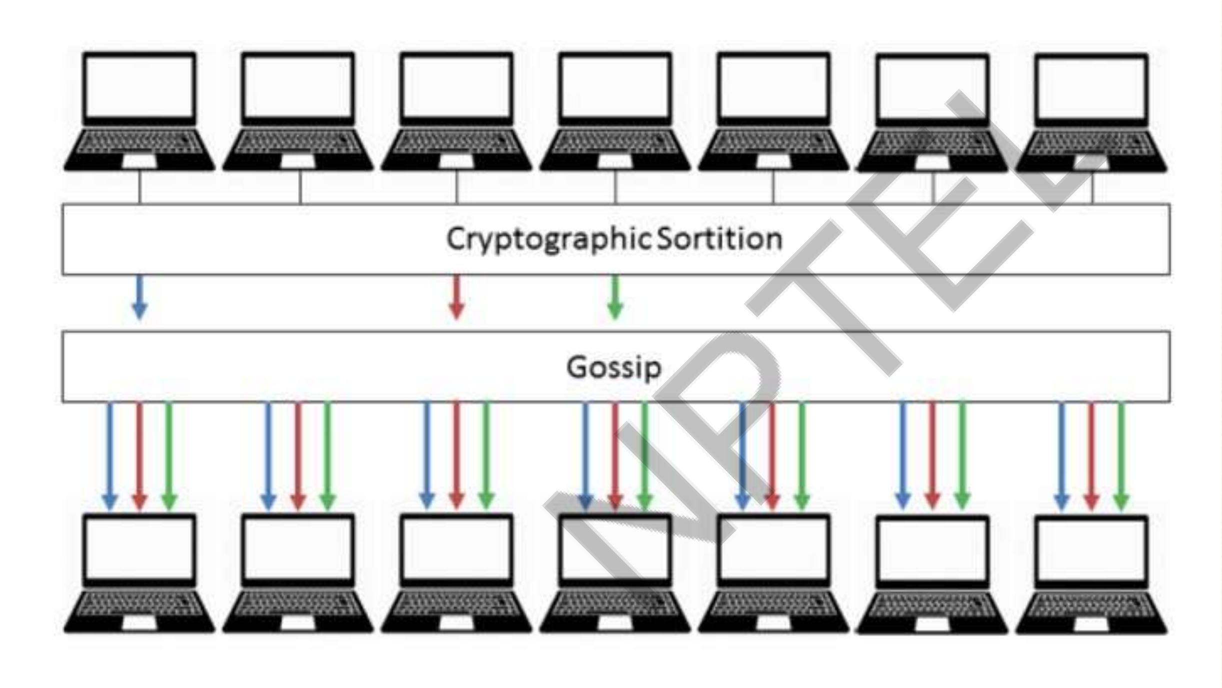
Architecture of Algorand

- Select a **random user**
  - prepare a block
  - propagate block through gossiping

- Select **random committee** with small number of users (~10k)
  - run Byzantine Agreement on the block
  - digitally sign the result
  - propagate digital signatures

- **Who select the committee?**

Cryptographic Sortition in Algorand

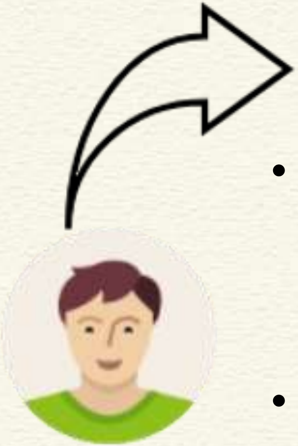


Cryptographic Sortition

- Each committee member selects himself according to per-user weights
  - Implemented using **Verifiable Random Functions** (**VRFs**)

- <hash,proof> ← $VRF_{sk}(x)$
  - **x:** input string
  - **(pki,ski):** public/private key pair
  - **hash:** hashlenbit-long value that is uniquely determined by sk and x
  - **proof:** enables to check that the hash indeed corresponds to x

Committee Member Selection



<hash,proof,j> <--- Sortition(sk,seed,threshold,role,w,W)

- **seed:** publicly known random value
    - seed published at Algorand’s round r using VRFs with the seed of the previous round r − 1
- **threshold:** determines the expected number of users selected for that role
- **role:** user for proposing a block/ committee member
- **w:** weight of a user
- **W:** weight of all users
- **j:** user gets to participate as j different “sub-users.”

Byzantine Agreement in Algorand: BA*

- **Two phase**:
  - Two phase agreement –
    - *Final Consensus*
    - *Tentative Consensus*

Byzantine Agreement in Algorand: BA*

- **Strong Synchrony**: Most honest users (say, 95%) can send message that will be received by most other honest users within a known time bound
  - Adversary can not control the network for long
  - Ensures liveness of the protocol

Byzantine Agreement in Algorand: BA*

- **Weak Synchrony**: The network can be asynchronous for long (entirely controlled by adversary) but bounded period of time
  - **There must be a strong synchrony period after a weak synchrony period**
  - Algorand is **safe** under weak synchrony

Final Consensus

- One user reaches final consensus
  - Any other user that reaches final or tentative consensus in the same round must agree on the same block value (<u>**ensures safety**</u>)
  - Confirm a transaction when the block reaches to the final consensus

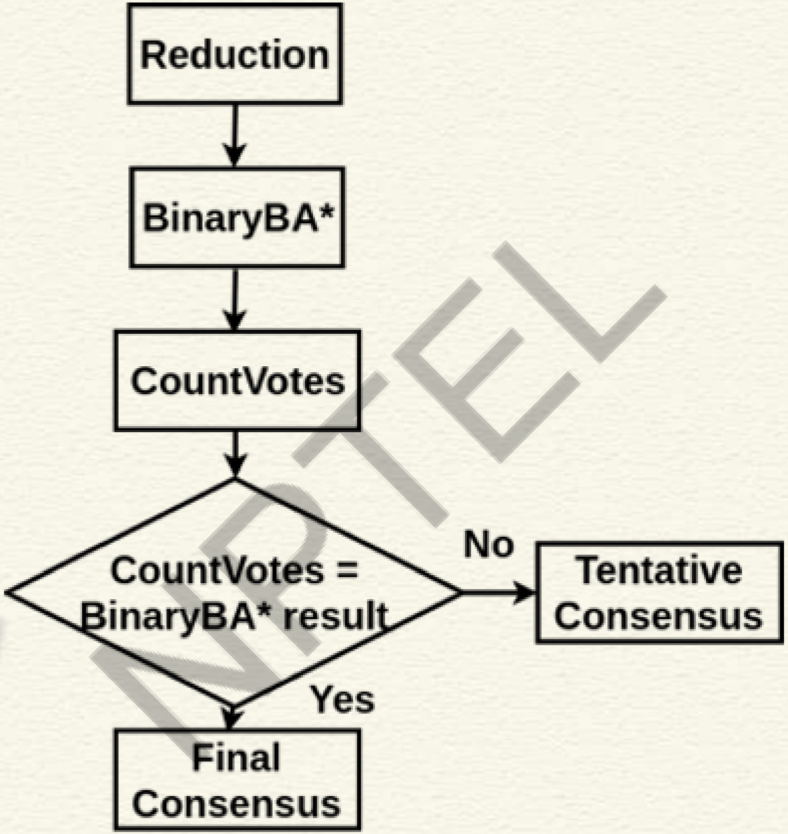
Tentative Consensus

- One user reaches tentative consensus
  - Other users may have reached consensus on a **<u>different (but correct)</u>** block
  - Can be in two cases
    - **<u>The network is strongly synchronous</u>** - adversary may be able to cause BA* to reach tentative consensus on a block - BA* is unable to confirm that the network was strongly synchronous
    - **<u>The network was weakly synchronous</u>** - BA* can form multiple forks and reach tentative consensus on two different blocks - users are split into groups

Coming out of Tentative Consensus

- Run BA* periodically to come out of tentative consensus - run the next round
  - Network cannot be under weak synchrony all the times
  - Cryptographic sortition ensures different committee members at different rounds of the BA*

BA* Overview


Reduction
BinaryBA*
CountVotes
CountVotes = BinaryBA* result
No
Tentative Consensus
Yes
Final Consensus

Conclusion

- Algorand has multiple advantages
  - Bitcoin like scalability
  - BFT like throughput
  - No fork

- **<u>Caution:</u> Needs a really large network**


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

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

Lecture 43: Identity Management - I

CONCEPTS COVERED

- Basic Concepts of Identity
- Centralized Identity Management
- Introduction to Decentralized Identity Managment

KEYWORDS

- Identity
- Centralized Identity Management
- Single Sign on
- Self-Sovereign Principle
- Decentralized Identifier

What is Identity?

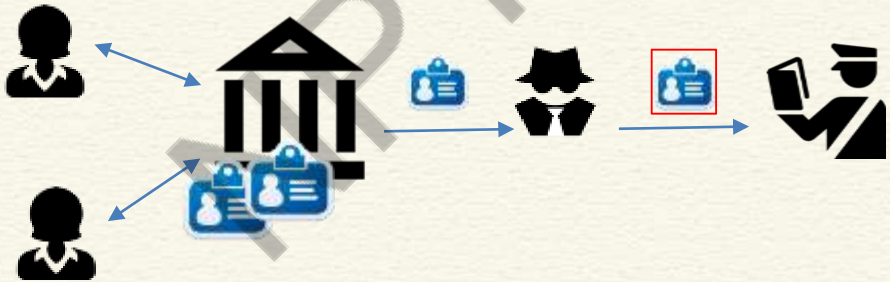
- People are known by their identities - drives every business and social interaction
- Physical Identity is a collection of attributes
  - Name
  - Age
  - Financial history
  - Work history
  - Address history
  - Social history
  - ….

Centralized Digital Identity

- Individuals do not have any control over the information that comprises their identities
- Identity fraud - no visibility over the identity attributes
- Authentication
- Authorization
- Verification

Centralized Digital Identity

- Individuals do not have any control over the information that comprises their identities
- Identity fraud - no visibility over the identity attributes
- Authentication
- Authorization
- Verification

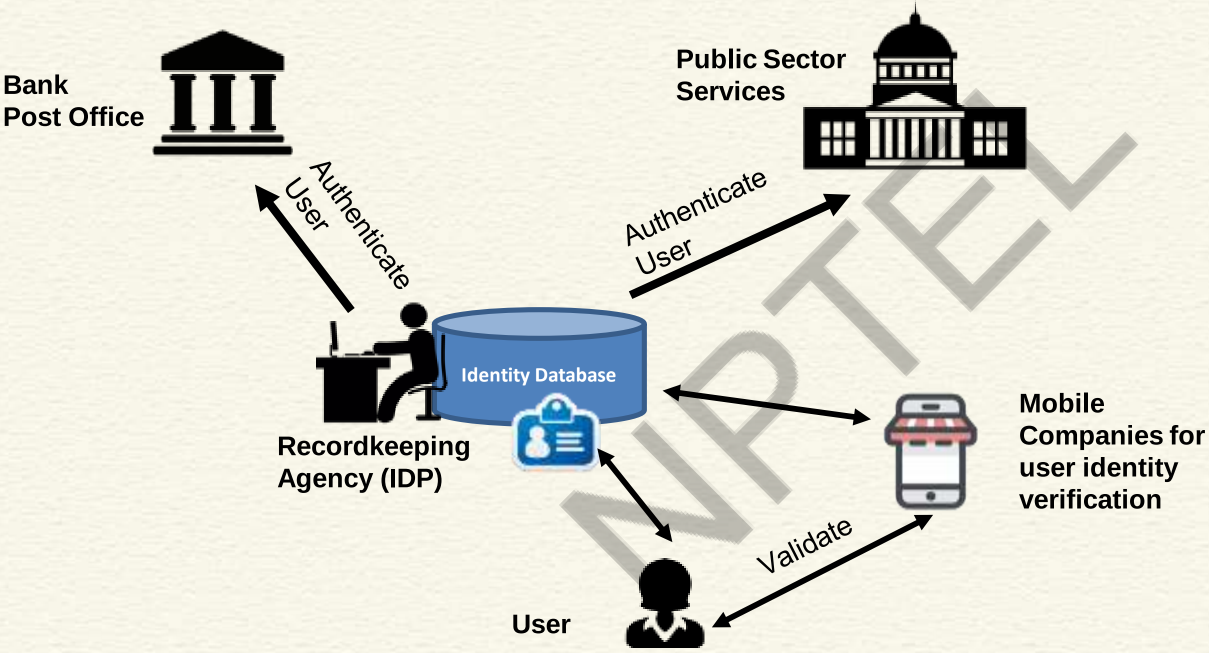


Digital Identity - Single Sign On (SSO)

- Single identity for various purposes
- No need to maintain multiple identity documents
- Widely conceptualized in software industry
- One password to access multiple services
- Single identity provider (IDP) maintains the identity
- Identity consumers (services) use the IDP to authenticate the identity holder
- During authentication, the identity is not exposed to the services

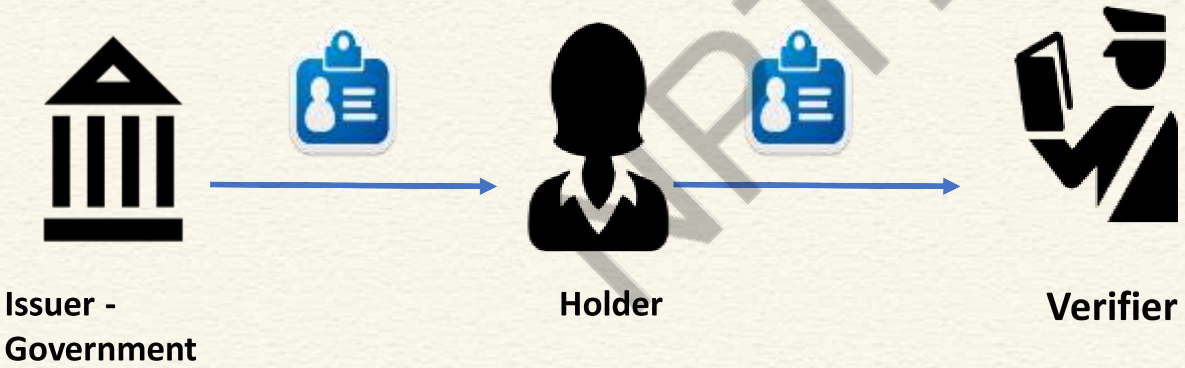
Image Source: https://www.e-spincorp.com/global-theme-and-feature-topics/single-sign-on-sso/

SSO and Decentralization



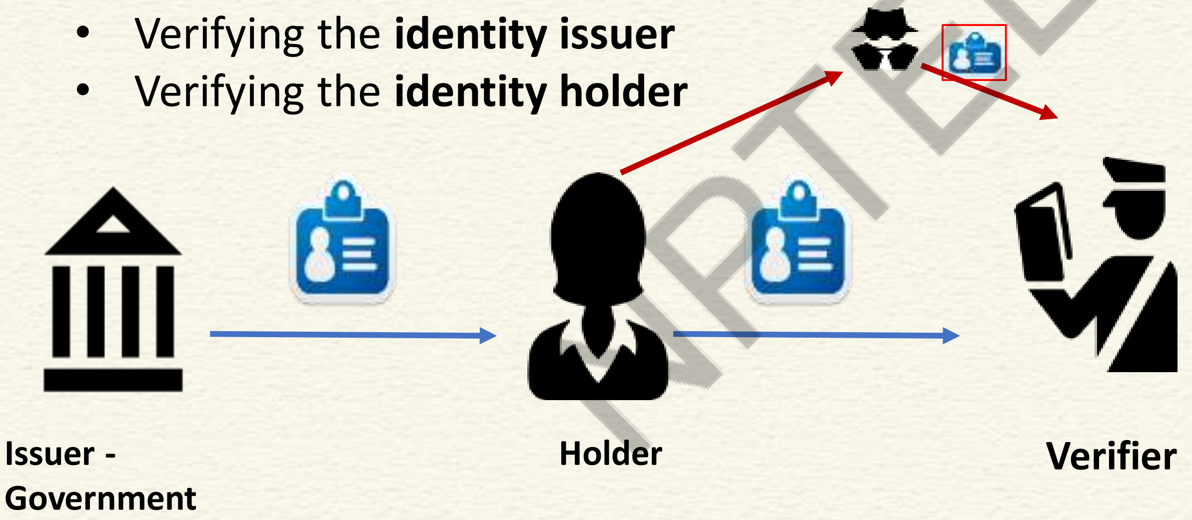
Decentralizing Digital Identity

- No Centralized Trusted Identity Provider / Registry
- Digital representation of physical identity
- Two major problems:
  - Verifying the **identity issuer**
  - Verifying the **identity holder**



Decentralizing Digital Identity

- No Centralized Trusted Identity Provider / Registry
- Digital representation of physical identity
- Two major problems:
  - Verifying the **identity issuer**
  - Verifying the **identity holder**



Fundamental Principles of Digital Identity Management

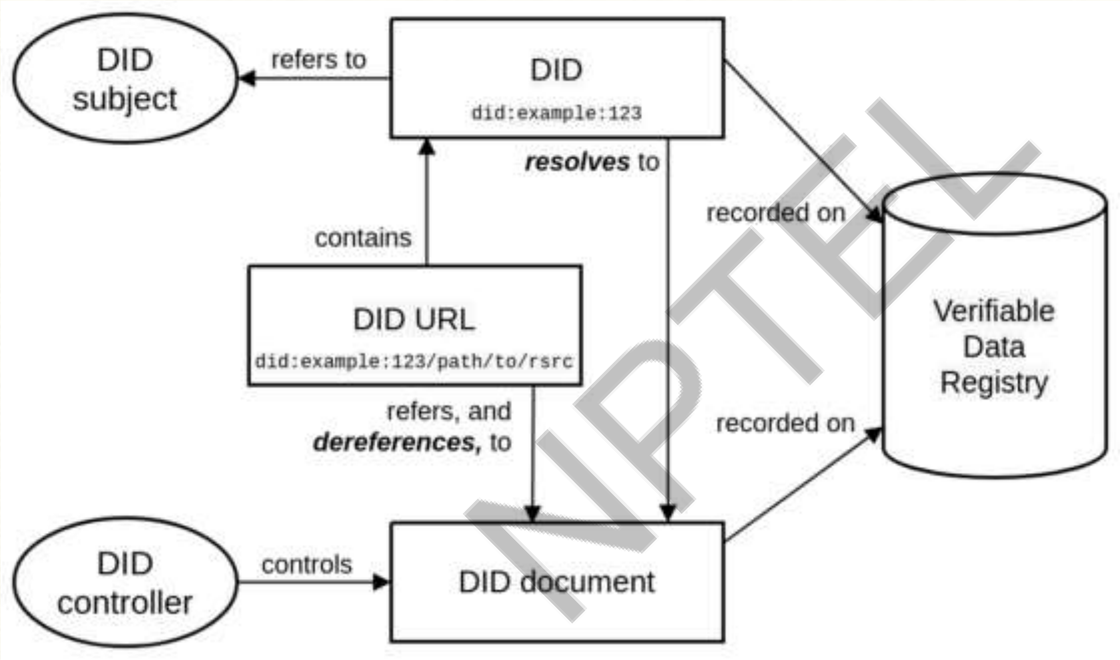
- Self-Sovereign Identity (Privacy Control)
- Individual should have full control and ownership of their identity information
- Individuals can control the usage of their own identity profile for business and social interactions (Consent - agreement for information usage)
- Identical to how we use our physical identity
- Holder possesses the ID
- Holder chooses whom to present the ID
- Burden at individual user?

Decentralized Identifiers (DIDs)

- Provides Verifiable, Decentralized Digital identity
- Designed to be decoupled from:
  - centralized registries
  - identity providers
  - certificate authorities
- Holder of DID can prove its ownership on the DID without the help of any other party
- W3C Proposed Recommendation

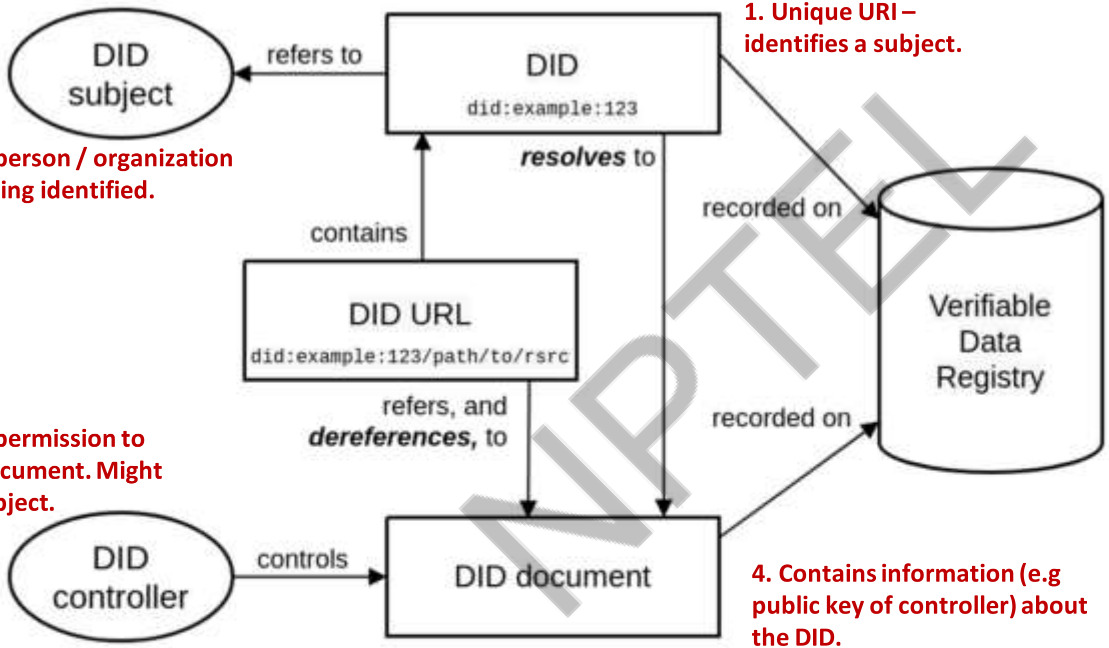
https://www.w3.org/TR/did-core/

DID Architecture



https://www.w3.org/TR/did-core/

DID Architecture



1. Unique URI – identifies a subject.

2. Entity (person / organization / etc..) being identified.

3. Entity with permission to change DID document. Might be same as subject.

4. Contains information (e.g public key of controller) about the DID.

5. System where DID documents are stored.

https://www.w3.org/TR/did-core/

CONCLUSIONS

- **Introduced the fundamental concepts of identity management**
- **Centralized vs. decentralized identity management**
- **DID as a W3C recommendation**

REFERENCES

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


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

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

Lecture 44: Identity Management - II

CONCEPTS COVERED

- How DID Works
- DID Work Flow
- Decentralized DID Registry – Use of Blockchain
- Verifiable Credentials

KEYWORDS

- DID
- DID Registry
- Hyperledger Indy
- Verifiable Credential (VC)

DID URI

- Controller controls a **DID Document**.
- A **DID** is a unique address (URI) to the location of that document.

Scheme

$\underbrace{\text{did}}_{\text{Scheme}}:\underbrace{\text{example}}_{\text{DID Method}}:\underbrace{\text{123456789abcdefghi}}_{\text{DID Method-Specific Identifier}}$

DID Method

DID Method-Specific Identifier

System which supports DID scheme. - Eg. DID resolution.

Uniquely identifies a DID doc within the DID Method.

http://www.faqs.org/rfcs/rfc2396.html

https://www.w3.org/TR/did-core/

DID Document

- A set of data describing the DID subject, including mechanisms such as cryptographic public keys, that the DID subject or a DID delegate can use to authenticate itself and prove its association with the DID.
- DID document consists of a map of entries, each entry consisting of a key/value pair.

Representation-specific entries include JSON, XML, etc

https://www.w3.org/TR/did-core/

DID Document Example (JSON)

```
{
    "id": "did:example:123456789abcdefghi",
    "authentication": [{
        "id": "did:example:123456789abcdefghi#keys-1",
        "type": "Ed25519VerificationKey2020",
        "controller": "did:example:123456789abcdefghi",
        "publicKeyMultibase":
          "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV
          "
    }],
    "service": [{
      "id":"did:example:123456789abcdefghi#linked-domain",
      "type": "LinkedDomains", // external (property value)
      "serviceEndpoint": https://bar.example.com
    }]
}
```

DID for a particular DID subject

Verification Method specifying how the DID subject can authenticate itself.

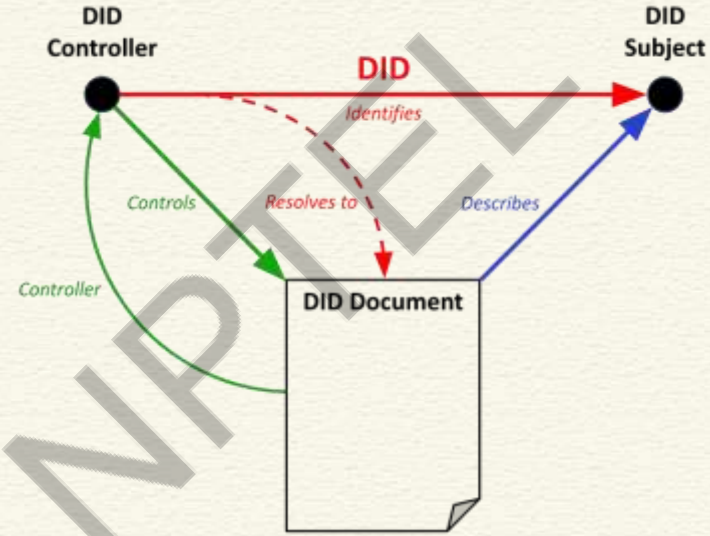
Service Endpoint denoting ways of communicating with the DID subject

It tells how to reach the subject. Otherwise, there is no meaningful use of authentication

https://www.w3.org/TR/did-core/

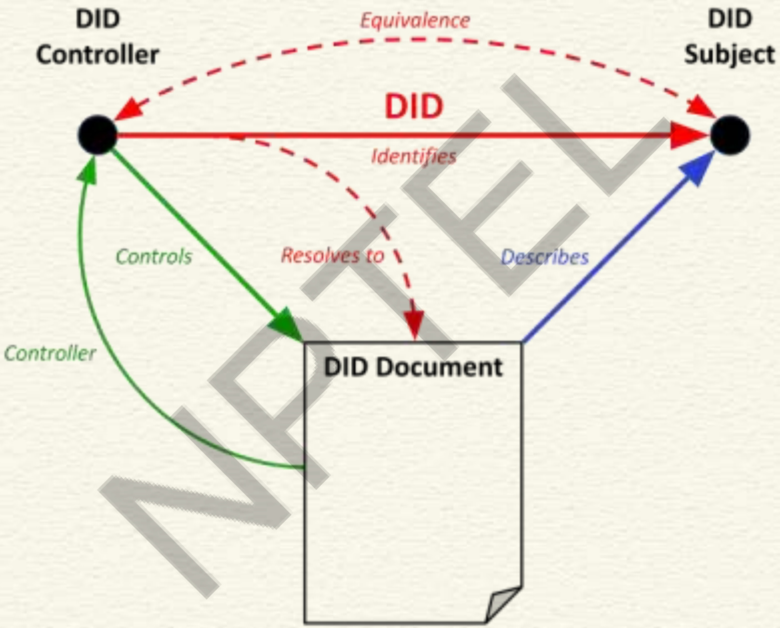
Relationship between Different Components of DID

- A DID is an identifier assigned by a DID controller to refer to a DID subject and resolve to a DID document that describes the DID subject.
- The DID document is an artifact of DID resolution and not a separate resource distinct from the DID subject.
- DID document resides inside verifiable data registry

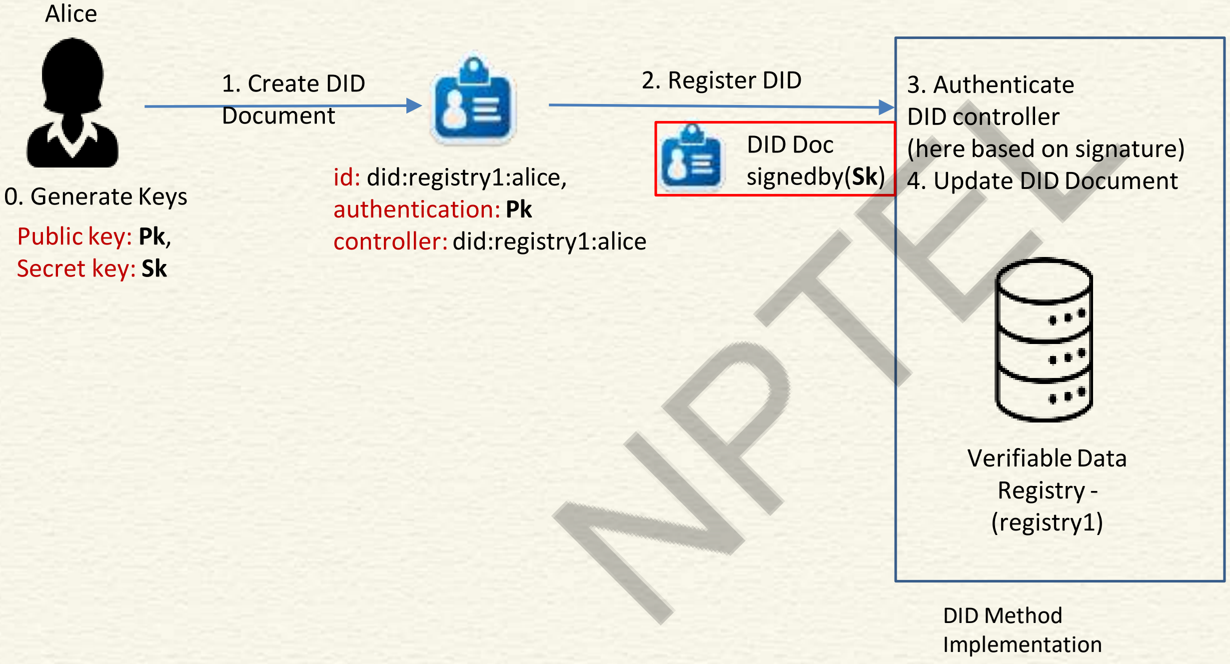


Relationship between Different Components of DID

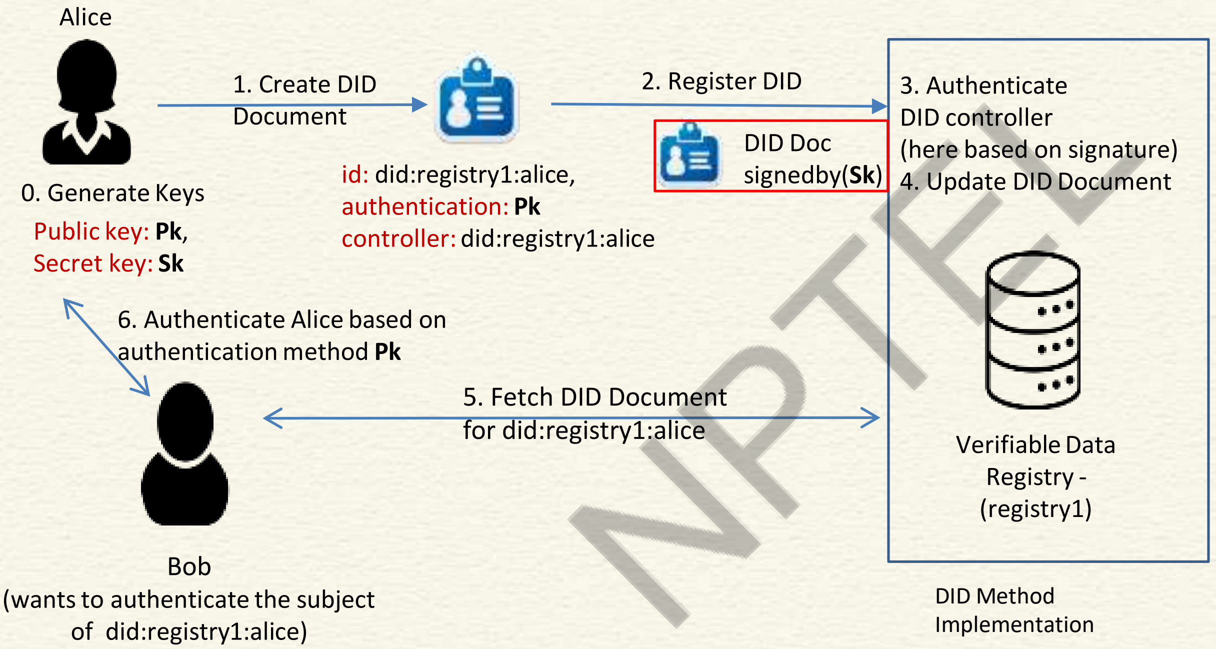
- Often the DID Subject and the DID Controller are the same entity



DID Flow – DID Registration

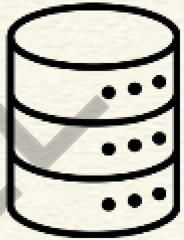


DID Flow – DID Registration



DID Method Security

- DID Registry ideally enforces DID Method protocols.
- Centralized DID Registry brings in risks
  - Manipulating DID Documents
    - Changing authentication methods
  - Censoring DID Documents
    - Refusing to resolve certain DID Documents
- Lack of Transparency



Decentralized DID Registry

- Blockchain based Implementation of Verifiable Data Registry
- DID Methods are implemented as smart contracts.
  - Smart contracts enforce how authorization is performed to execute all operations, including any necessary cryptographic processes.
- Transparent Immutable Ledger allows verifiability of DID Documents
  - Any party can validate if a DID Document's creation / updation transactions were authenticated or not.

Verifiable Data Registry

Blockchain based DID Registry

Public permissioned ledger based registry.

- Any party can read the ledger.
- Only selected (registered) parties and write to the ledger.

HYPERLEDGER INDY

https://hyperledger-indy.readthedocs.io/en/latest/

DIF

Sidetree

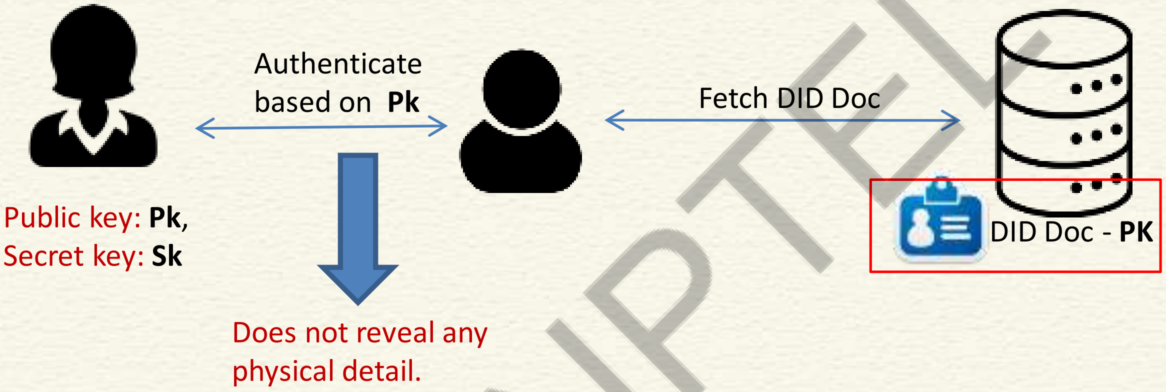
Protocol for creating scalable DIDnetworks that can run atop any existing permissionless blockchain. (e.g. Bitcoin, Ethereum, etc.)

https://identity.foundation/sidetree/spec/

Binding DID to Physical Identity

DIDs only allow a DID controller to prove its control over its DID Document.

This is useful to authenticate an entity with respect to its DID

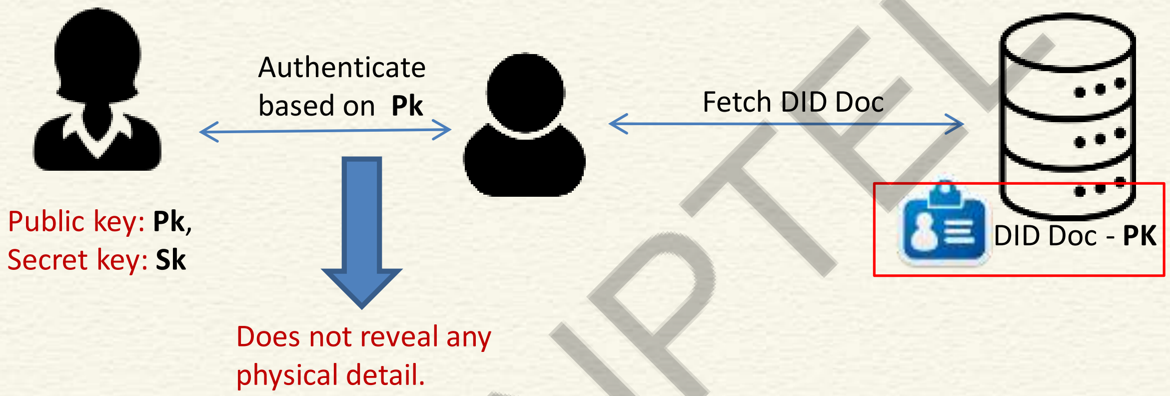


If some physical detail is presented, then that is only self attested by the DID controller, and not any verified information.

Binding DID to Physical Identity

DIDs only allow a DID controller to prove its control over its DID Document.

This is useful to authenticate an entity with respect to its DID



DID are not inherently tied to any physical identity (real world identity).

Verifiable Credentials

- **Verifiable Credentials Data Model** – W3C Recommendation
- Digital Representation of Credentials
  - Driver's licenses - assert that capability of operating a motor vehicle
  - University degrees - assert our level of education
  - Government-issued passports - permit to travel between countries
  - Identity – Birth Certificate, Citizenship Certificate, etc.
- **Decouples Issuer, Holder and Verifier**
- **Cryptographically secure**
- **Privacy respecting**
- **Machine-verifiable**

https://www.w3.org/TR/vc-data-model/

CONCLUSIONS

- **Implementation of DID**
- **Use of blockchain for DID registry implementation**
- **Verifiable credentials and their relationship with DID**

REFERENCES

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


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

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

Lecture 45: Identity Management - III

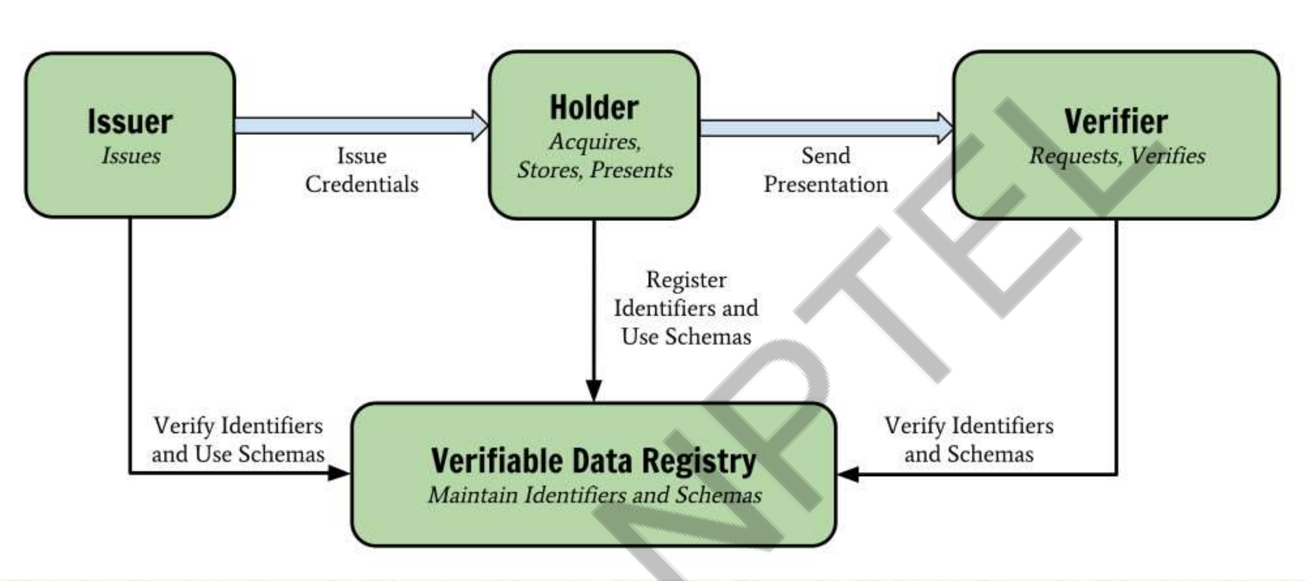
CONCEPTS COVERED

- Working Principle of Verifiable Credentials (VCs)
- VC Issuer, Holder and Verifier
- Use of Decentralized Registry in VC Management
- VC Trust Model
- Combining DID and VC

KEYWORDS

- **Verifiable Credential (VC)**
- **VC Presentation**
- **DID**
- **Hyperledger Aries**

VC Data Model Components

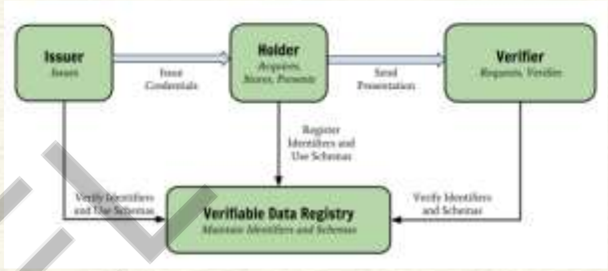


https://www.w3.org/TR/vc-data-model/

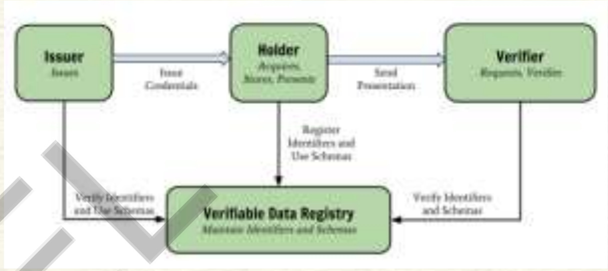
VC Data Model Components

Holder - possesses one or more VC and generating **verifiable presentations** from them. Example holders include students, employees, and customers.

Issuer –Asserts claims (in physical world) about one or more subjects, creating a VC from these claims, and transmitting the VC to a holder. Example issuers include universities, governments, etc.



VC Data Model Components



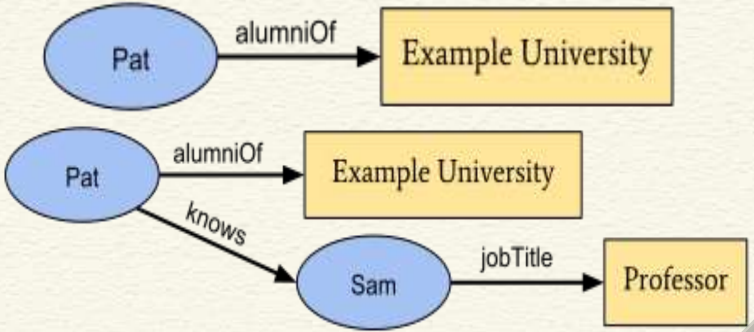
Subject - Entity about which claims are made. Example subjects include human beings, animals, and things.
Holder of a VC might not be the subject - example, a parent (the holder) might hold the verifiable credentials of a child (the subject), or a pet owner (the holder) might hold the verifiable credentials of their pet (the subject).
Note: some credentials might even be self-certified by the subject

Verifier – Receives verifiable presentation to assert claims about subject. Example verifiers include employers, security personnel, and websites.

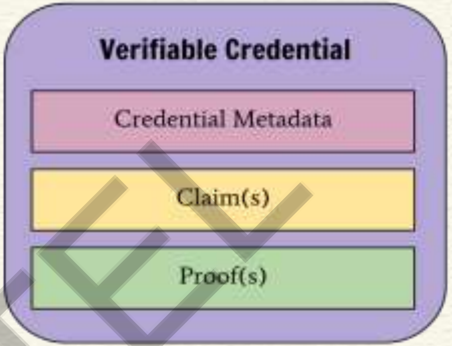
Verifiable data registry - System for creation and verification of DID, keys, and other relevant data, such as VC schemas, revocation registries, issuer public keys, and so on.

VC Data

Claims

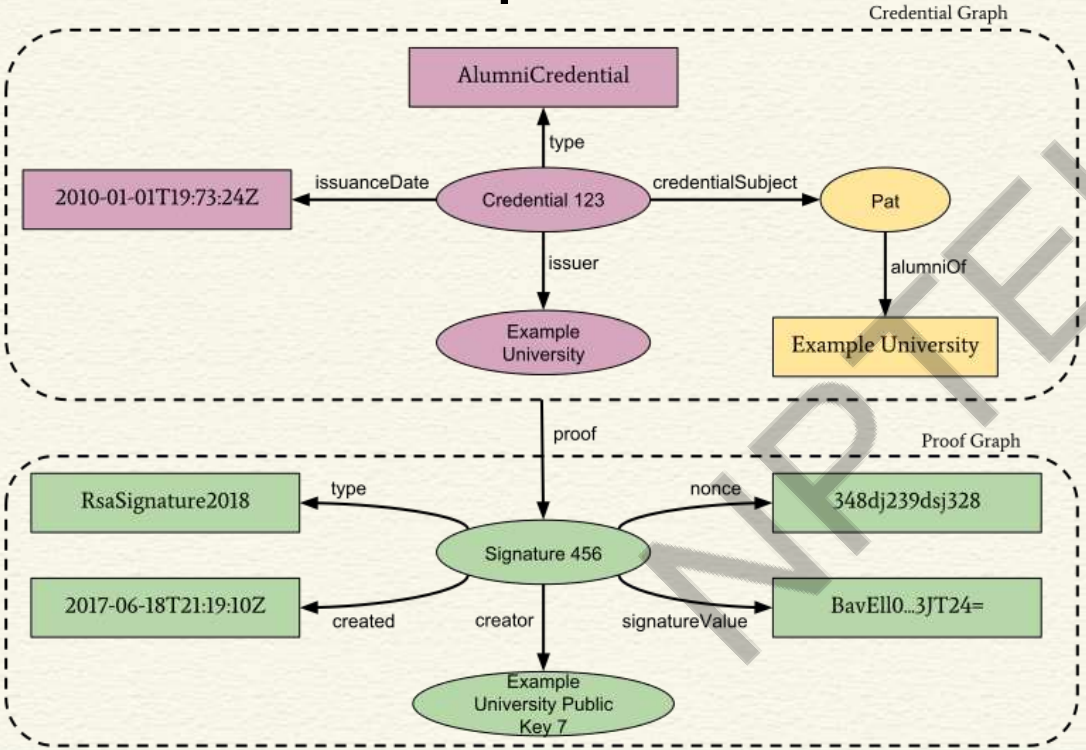


- A claim is a statement about a subject.
- Here Pat and Sam are subjects.



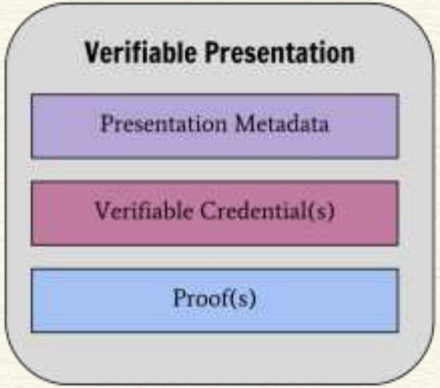
- A credential is a set of one or more claims made by the same entity.
- Proof is usually signature by the issuer

Information Graph of a basic Verifiable Credential



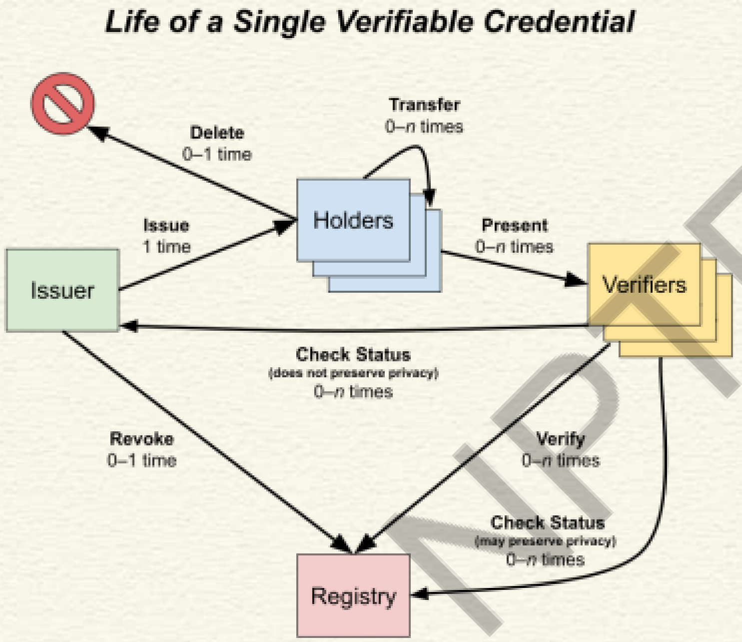
These two together are effectively forming the verifiable credential for Pat

Information Graph of Verifiable Presentation



A verifiable presentation expresses data from one or more VCs, and is packaged in such a way that the authorship of the data is verifiable. Holder has to convince that indeed the VC was issued to him.

Verifiable Credentials Flow

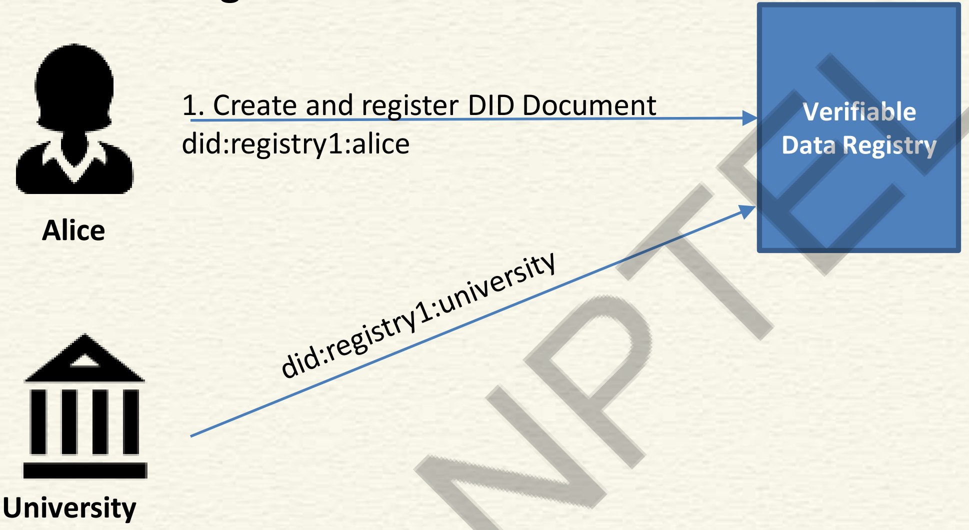


VC Trust Model

- Acting as issuer, holder, or verifier requires neither registration nor approval by any authority, as the trust involved is bilateral between parties.
- Verifier trusts the issuer to issue the VC that it received. To establish this trust, a VC is expected to either:
  - Include a proof establishing that the issuer generated the credential (signature), or
  - VC has been transmitted in a way clearly establishing that the issuer generated VC is not tampered in transit or storage.
- All entities trust the verifiable data registry to be tamper-evident and to be correct. Blockchain can help??
- Holder and verifier trust Issuer to issue true credentials about the subject, and revoke them quickly when appropriate.

Combining DIDs and VCs

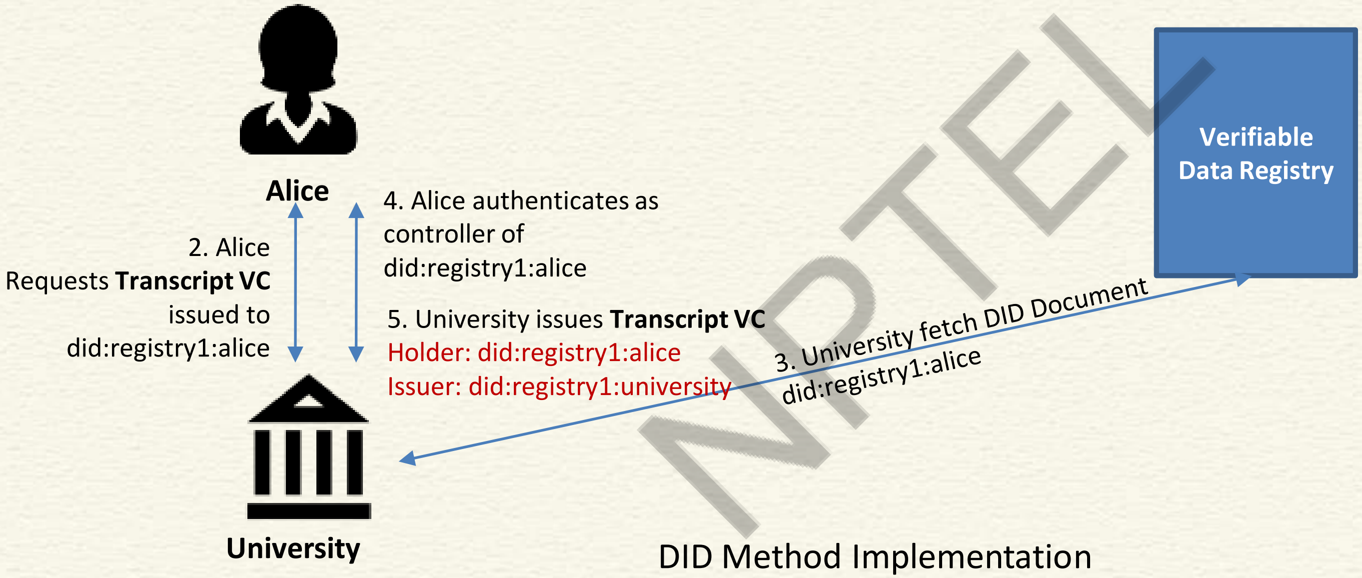
Step 1. Create and register DID



DID Method Implementation

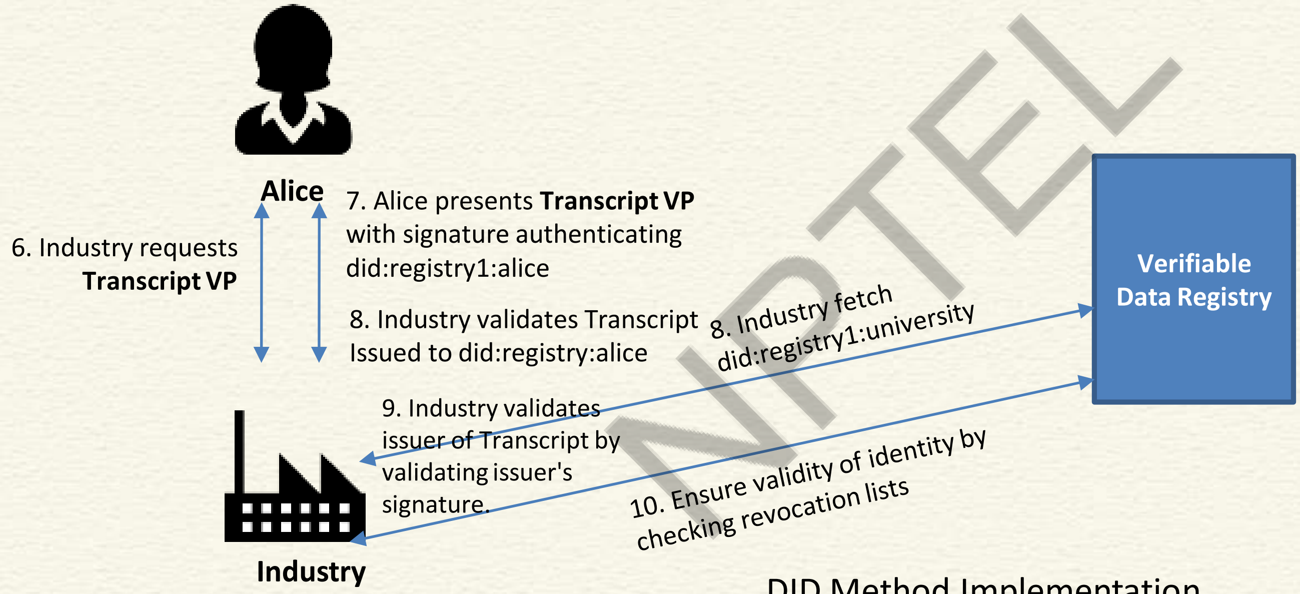
Combining DIDs and VCs

Step 2. Issue Verifiable Credential



Combining DIDs and VCs

Step 3. Verifiable Presentation and Verification



DID Method Implementation

Use of Blockchain for VCs

Hyperledger Aries is meant for creating, transmitting and storing verifiable digital credentials

CONCLUSIONS

- Explained the complete workflow of VCs
- VC trust model
- Combining DID and VC
- Introduced Hyperledger Aries

REFERENCES

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


Thank
you