Digital Identity Revolution Begins

Decentralized identity for THAIs

-- Phuttimate Niyakit -- Plinphon Patthayawij -- Supakavadee Rodklang -- Tommaso Amorosi

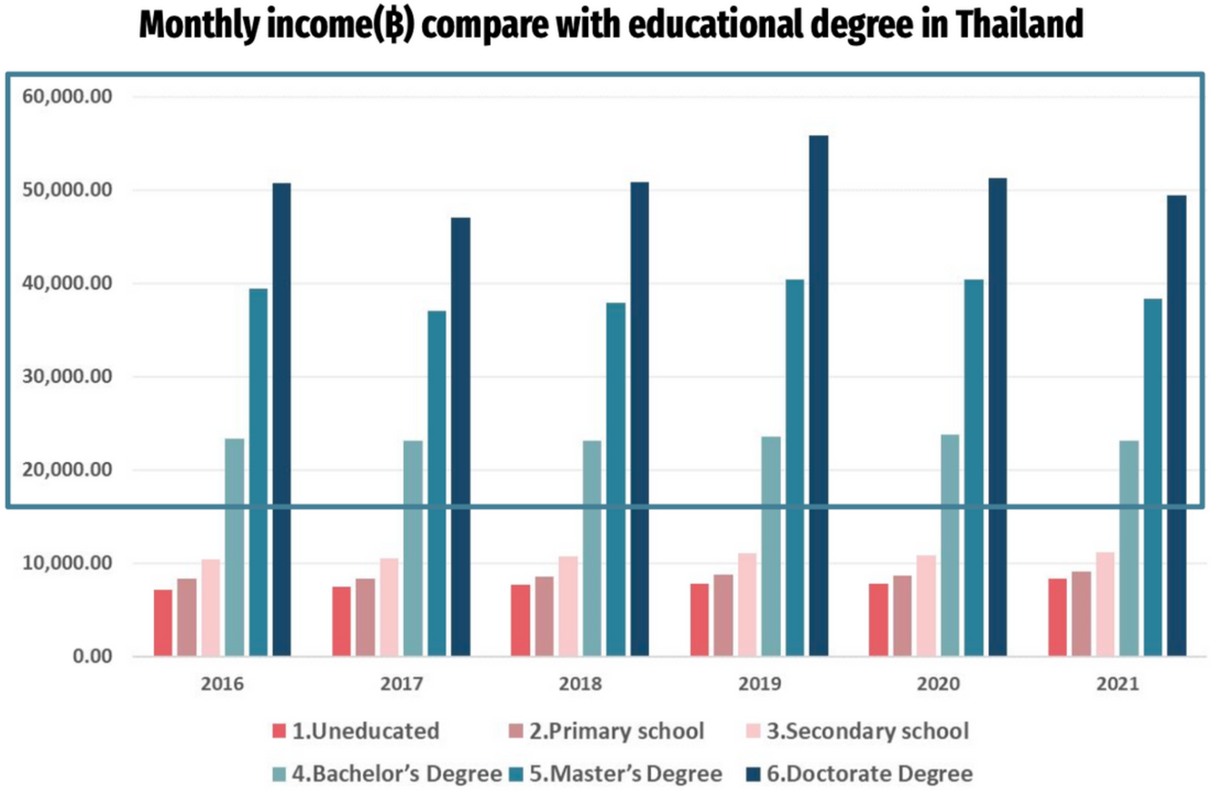
In this Presentation

Here's what we'll cover:

- Problem Statement / Motivation
- Architecture Overview
- Data Flow Diagram
- Algorithm
- DEMO
- QA Process
- Difficulties
- Future Extensions

Why Education Credentials Matter?

People with degrees or certifications consistently earn more and access better job opportunities than those without.



why education
MATTERS

- Education is a **foundation of trust** in society.
- Degrees represent **skills, knowledge, and integrity**.
- Employers, governments, and the public rely on **highly educated people to make decisions**.

What’s the problem with Degree?

In Thailand, cases of **fake academic degrees** used by public officials such as Senator Keskamol CENSORED and Minister Thaman CENSORED h**ave exposed critical gaps in credential verification.**

- Both claimed degrees from unaccredited institutions.
- Investigations revealed questionable legitimacy.
- Yet, there was no digital mechanism to verify authenticity instantly.
- This undermines trust in government and public institutions.

What is..

Decentralized identity

Decentralized Identity (DID)?

DiD is a framework that allows individuals and organizations to create, own, and manage their digital identities without relying on a central authority. Instead of having a single entity (like a government or corporation) control identity data, DIDs leverage blockchain technology to enable self-sovereign identities, where users have full control over their personal information.

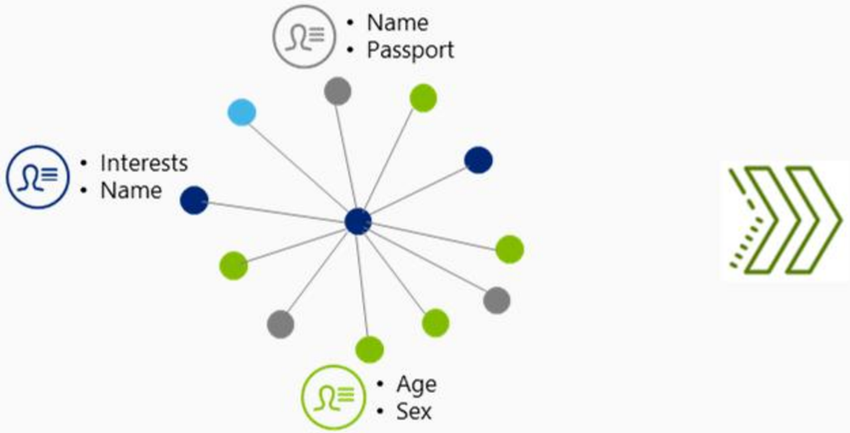


Figure 1. Traditional identity

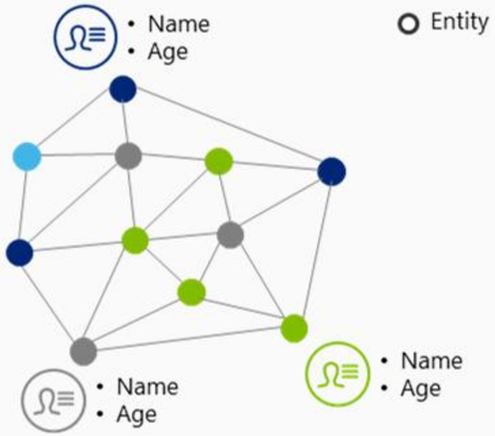


Figure 2. Blockchain-based identity

Key Components of DID in Blockchain

Decentralized Identifiers (DIDs)

Unique identifiers that are created and managed by the user. These identifiers are stored on a blockchain, ensuring they are immutable and verifiable.

Verifiable Credentials (VCs)

Digital statements issued by trusted entities (like universities or government agencies) that attest to certain attributes of the DID holder, such as educational qualifications or professional roles.

Blockchain Ledger

Serves as a decentralized and tamper-proof registry for DIDs and the associated metadata, ensuring transparency and trust in the system.

Decentralized identity (DiD)

a way to give people full control over their digital identity

Self-owned

You create and manage your identity, not a third party.

Private

You choose what information to share and with whom.

Verifiable

Anyone can check if your identity or certificate is real, using cryptographic proofs.

Without DiD

- Anyone can **fake a degree** and it’s hard to check.
- You must **call or email** to verify someone’s certificate.
- We can’t prove if someone really **approved** the degree.

With DiD

- Degrees are **cryptographically** verifiable and tamper-proof.
- Institutions and roles are instantly **validated**
- No need to “trust” the document, j**ust verify** it.

User Flow

User

login or sign up

already signed up?

no

create account

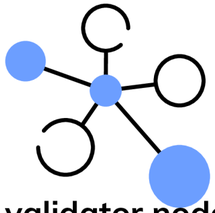
Get DID

yes

claim certificate

end

get proof/hash verification



validator node
POA

receive TX

verify signature

Broadcasts via IPv8

verify vote signature

propose block

create & broadcast vote

finalize block

add block to chain

stores the cert TX / hash for later verification

user manually sent public_key to uni

University

after user finish degree/course

Register

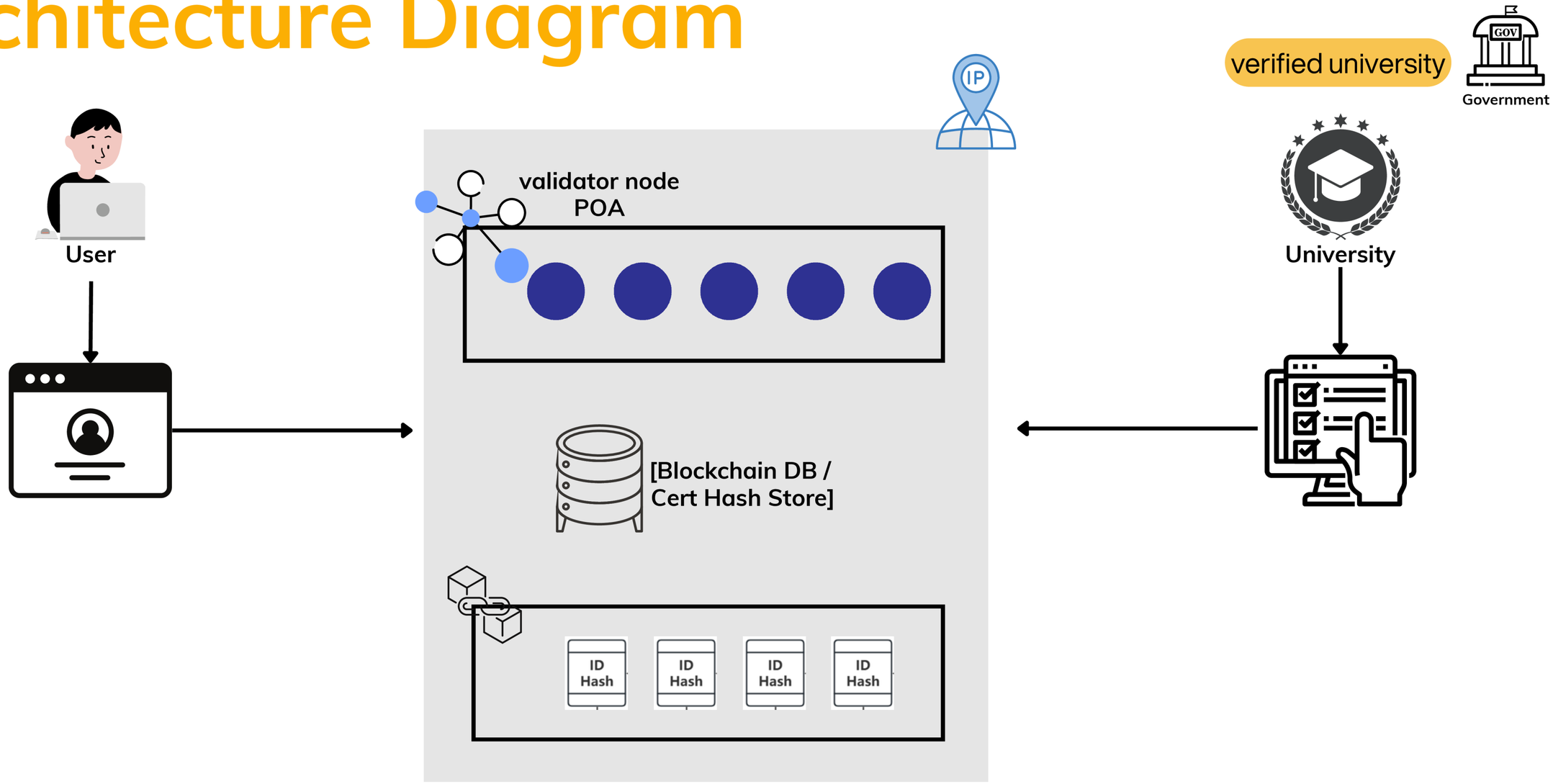
creates the transaction (TX)

Signs the TX with their private key

issue certificate

certificate hash for verification

Architecture Diagram



Algorithmic components

Components:

- **Double voting failsafe**
- **Sybil detection failsafe**
- **Message replay failsafe**
- **Byzantine node failsafe**

Double voting failsafe

This is designed to block users from have more than one vote. The mechanism is the user make one vote and its stored in memory of the node. If that user tries to vote again it will check if they already voted so it actively blocks them.

Sybil attack failsafe

This attack occurs when you split off your node into multiple nodes to increase your voting power. The source IP address would be identical in the split copies.

Message replay failsafe

This one in particular is problematic. This issue comes from duplicate messages. The original message was cleared so it will assume the next are cleared.
The solution is identical to the double voting failsafe. It keeps track of messages and blocks duplicates.

Byzantine node failsafe

This one didn’t work as intended. It was working weird. The premise of it was to check the validator nodes to see if they were in sync with the decision.

It didn’t work so testing how it was affecting the process was not possible.


QUALITY
ASSURANCE
Product
Pants
Shoes
11/2018
2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

What we measured?

TX Verification

Verified that invalid signatures are rejected by the system

Block Finalization Timing

Each finalized block includes TX hashes, making verification auditable and tamper-proof.

Vote Collection

Measured how many valid votes are required (≥3) to finalize a block

Double Voting Prevention

Simulated and verified system behavior under attack attempts

Assumptions

correcness of node

We assume validator nodes are not malicious or partitioned

correctness of display

User interface (e.g. claim certificate) is trusted to display the correct certificate hash

vote

Practical Byzantine Fault Tolerance (PBFT) requires >⅔ honest nodes.

DEMO

Difficulties & Future Extensions

Difficulties

voting and consensus mechanism

- Designing a quorum-based decision rule (e.g. ≥3 accepts to finalize a block)
- Preventing double voting, replay attacks, byzantine attacks, and Sybil behaviour

Future Extensions

Dynamic Validator Set / Rotating Authority

Rotating proposers has been shown to improve fairness and maintain liveness in HotStuff-based BFT protocols (e.g., Sync HotStuff), and can even lead to significant latency improvements in comparison to static leader models.

Shrestha, Nibesh & Abraham, Ittai & Nayak, Kartik. (2021). Optimal Good-case Latency for Rotating Leader Synchronous BFT. 10.4230/LIPIcs.OPODIS.2021.27.

State Persistence Layer

To ensure that all finalized blocks, certificate hashes, and voting results are not lost when a node restarts.