CSB/CSE Capstone Project 2023

Oracle Decentralized Identity Solution

Ari Carp, William Han, Adrian Ross, Shun Udea
Advisor: Professor Andrea Smith
Sponsor: Oracle Corporation with Mark Rakhmilevich and Bala Vellanki
Special thanks to Professor Hank Korth

Motivation

How Do You Preserve Privacy While Verifying Identity and Credentials?

Today, we are **issued credentials as paper documents** (e.g., driver's license or passport). When we need to prove claims, such as our name, we hand over the **entire document**.

The paper credential model ideally **proves to the *verifier* :**
- The identity of the credential's ***issuer***.
- The identity of the credential's ***holder***.
- The claims have not been altered.
- The claims meet the request's requirements.

However, the current model presents **risks to privacy, particularly online :**
- Possible forgery and human error.
- Additional personal information included (e.g. place of residence, date of birth) not needed for verification.
- Insecure channels, tampering, and data breaches.

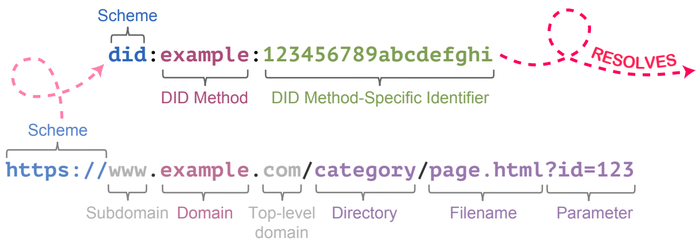
Resulting in an **overall imbalance of trust**.

A SOLUTION: Self-Sovereign Identity supported by a trustless network: a **blockchain** ledger-based verifiable data registry (VDR). Powered by ***Oracle Blockchain Platform*** and the W3C ***Verifiable Credential Model***, a proven infrastructure behind authentic data and decentralized identity. This gives individuals full ownership and control over their digital identities and credentials without relying on multiple siloed identity systems maintained by companies and government agencies.

Decentralized Identities (DIDs)

A Unique URL to Describe You

A decentralized identifier (DID) is a W3C universally unique identifier (UUID) with added power:
- DIDs are ***resolvable*** into a **DID Document (DIDDoc**) that contains at a minimum **public keys** for the controller of the DID that allow them to prove control over the DID.
- A DID is defined by its **DID method**—a specification that defines how to create, resolve (read), update, and delete the DIDs. Resolving a DID is like resolving a URL, but instead returns the DIDs associated DIDDoc, which enables issuance and verification of credentials.



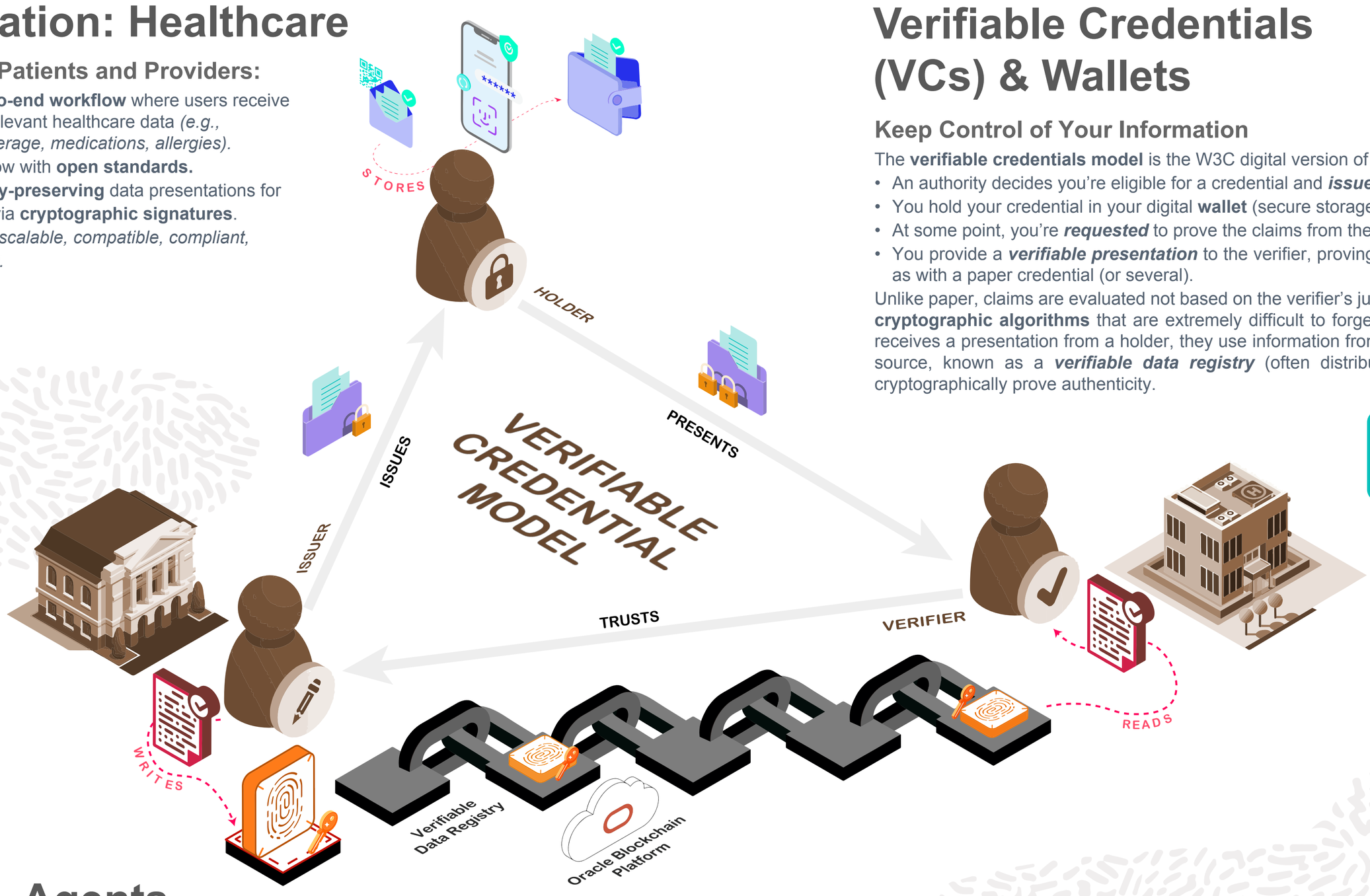
The standard elements of a DIDDoc

1. **DID** (for self-description)
2. **Set of public keys** (for verification)
3. **Set of auth methods** (for authentication)
4. **Set of service endpoints** (for interaction)
5. **Timestamp** (for audit history)
6. **Signature** (for integrity)

Application: Healthcare

Benefits to Patients and Providers:

- Support **end-to-end workflow** where users receive and present relevant healthcare data *(e.g., insurance coverage, medications, allergies).*
- Enable workflow with **open standards.**
- Enable **privacy-preserving** data presentations for specific uses via **cryptographic signatures**.
- *Open-source, scalable, compatible, compliant, W3C standard.*



Verifiable Credentials (VCs) & Wallets

Keep Control of Your Information

The **verifiable credentials model** is the W3C digital version of the paper model:
- An authority decides you're eligible for a credential and ***issues*** you one.
- You hold your credential in your digital **wallet** (secure storage).
- At some point, you're ***requested*** to prove the claims from the credential.
- You provide a ***verifiable presentation*** to the verifier, proving the same things as with a paper credential (or several).

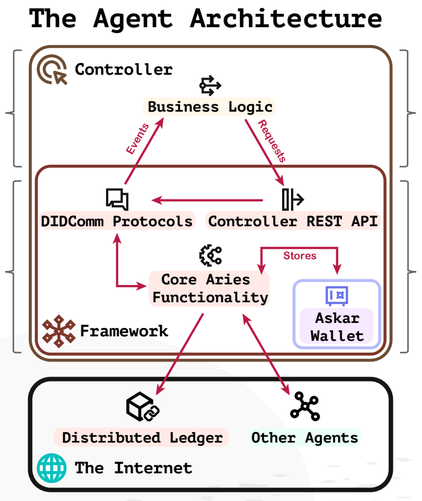
Unlike paper, claims are evaluated not based on the verifier's judgment but using **cryptographic algorithms** that are extremely difficult to forge. When a verifier receives a presentation from a holder, they use information from a decentralized source, known as a ***verifiable data registry*** (often distributed ledgers), to cryptographically prove authenticity.

Agents

Your Representative in the DID Network

An **agent** is the software that represents a user in the decentralized identity network. It consists of components split into two groups: the ***framework*** and the ***controller***. Both are built on the open-source **Hyperledger Aries** toolkit and are part of the greater Hyperledger ecosystem for enterprise-grade blockchain technologies.

The Agent Architecture



The Controller:
1. Handles different events
2. Retrieves info from protocol
3. Initiates appropriate requests

The Aries Agent Framework:
1. Determines connection type
2. Creates protocol state object
3. Stores protocol state object
4. Sends webhook protocol state object in its wallet
5. Interacts with other agents and the Verifiable Data Registry

HYPERLEDGER ARIES

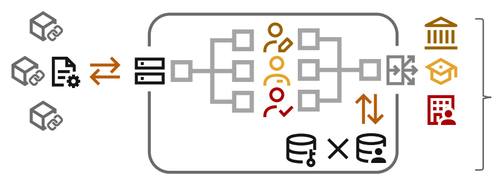
Controller Examples:
for Issuers, Holders, and Verifiers
- Business Integration (External systems or IoT)
- User Software (CLI or GUI Frontend)
- Mediators and User Mobile Wallets or Web Apps

Framework Construction:
- Base framework (Aries JavaScript Framework, ACA-Py, Aries Framework Go, etc.)
- Injectable plugins or modules for extra features
  - DID Resolvers (OracleDidResolver)
  - DID Registrars (OracleDidRegistrar)
  - Network State Objects (OracleLedgerService)
  - Key Types (PemKeyType)
  - Data Compression (DerKeyCompression)

- **DIDComm:** secure peer-to-peer messaging that makes up the pre-defined interaction sequences known as Aries Protocols
- **Cloud vs. Edge:** Cloud agents can communicate directly with the ledger and any other agent in the network. Edge agents like mobile wallets are possible but need a cloud agent to act as a **mediator**.

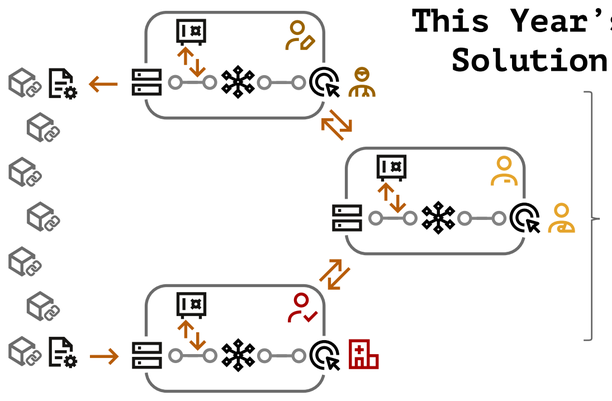
Contributions

Last Year's Solution



Verifiable University Transcripts:
- ✓ Implemented OBP Chaincode methods
- ✓ Key functions and APIs with NodeJS
- ✓ Built React Issuer + Verifier Portal
- ✗ Single Centralized Server (Not Distributed)
- ✗ Shared Insecure Storage (Faux Wallet)
- ✗ Hardcoded Secrets and Configurations
- ✗ Outdated DID and VC Protocols

This Year's Solution



Verifiable Healthcare Records:
- ✓ Updated OBP Chaincode and Protocols
- ✓ Interoperable and Scalable Agent Framework
- ✓ Encrypted Secure True-Wallet Storage
- ✓ Built Upon W3C and Open-Source Aries Standards
- ✓ Distributed P2P Network with Cloud & Edge Nodes
- ✓ Key functions and APIs with NodeJS
- ✓ Built React Patient and Provider Portals
- ✗ No Selective Disclosure
- ✗ Not Mobile Native

Oracle Blockchain Platform (OBP)

Oracle has developed a **permissioned blockchain** platform on **Hyperledger Fabric** to enhance transaction efficiency and streamline agreements between multiple parties through smart contracts, called chaincode in Hyperledger's terminology. This solution utilizes the platform as the Verifiable Data Registry.

Chaincode

OBP **Chaincode** (smart contract) methods are pieces of code that are **deployed on-chain** and define the rules and logic to write and validate transactions on the network. This facilitates the write and read operations agents need, as well as additional capabilities.

Proposed Future Work

- Switch to a credential type that supports selective disclosure
- Integrate zero-knowledge proofs (ZKPs) for predicate claims
- Add a native mobile solution with Aries Bifold and a mobile wallet
- Support for custodial wallet backup
- Integrate Oracle IAM support for Self-Sovereign Identity issuers and roles
- Add highly-scalable onboarding and identity-proofing solutions for issuers to mass distribute credentials
- Use Aries Agent Harness for end-to-end testing