Student : Saaruhan Sellappah from CY-Tech, France

Supervisor : Mr Pradeep Singh Shekhawat from BK BIET, India

Secure and Decentralized Identity Management

Issuer, Holder and verifier principle

Ganache TRUFFLE

Smart contract and use of Ganache and truffle

HYPERLEDGER

Use of the hyperledger Aries API

Web Application

The principle of the issuer, holder and verifier in a self-sovereign identity management

Introduction to self-sovereign identity management

- What is self-sovereign identity management (SSI)?

- Benefits of SSI: User control of data, Security, and Privacy.

- Introduction of the three main actors: issuer, holder, verifier.

The principle of the issuer, holder and verifier in a self-sovereign identity management

Role of the Issuer

-Definition of the issuer.

-Function of the issuer: Issuing verifiable credentials (VCs).

-Process: Creation and issuance of VCs.

-Examples of issuers: Government issuing passports, University issuing diplomas.

The principle of the issuer, holder and verifier in a self-sovereign identity management

Role of the Holder

-Definition of the holder.

-Function of the holder: Receiving, storing, and managing verifiable credentials.

-Use of a digital wallet for managing VCs.

-Importance of consent and data control by the holder.

-Example scenarios: Using a VC to prove identity or qualifications.

The principle of the issuer, holder and verifier in a self-sovereign identity management

Role of the verifier

-Definition of the verifier.

-Function of the verifier: Verifying the authenticity of the credentials presented by the holder.

-Verification process: Validation of the issuer's digital signature.

-Example scenarios: Verifying a diploma for employment, verifying a driver's license.

-Benefits for the verifier: Increased security, reduction in fraud and document forgery.

Using Truffle, Ganache, and Creating Smart Contracts, and Deploying on Ethereum IDE

Introduction to Truffle, Ganache, and Ethereum IDE

- Overview of tools: Truffle, Ganache, and Ethereum IDE. (quick demo)

-Purpose of each tool: Development, testing, and deployment of smart contracts..

-Benefits of using these tools: Simplified development workflow, robust testing environment, and streamlined deployment process.

Using Truffle, Ganache, and Creating Smart Contracts, and Deploying on Ethereum IDE

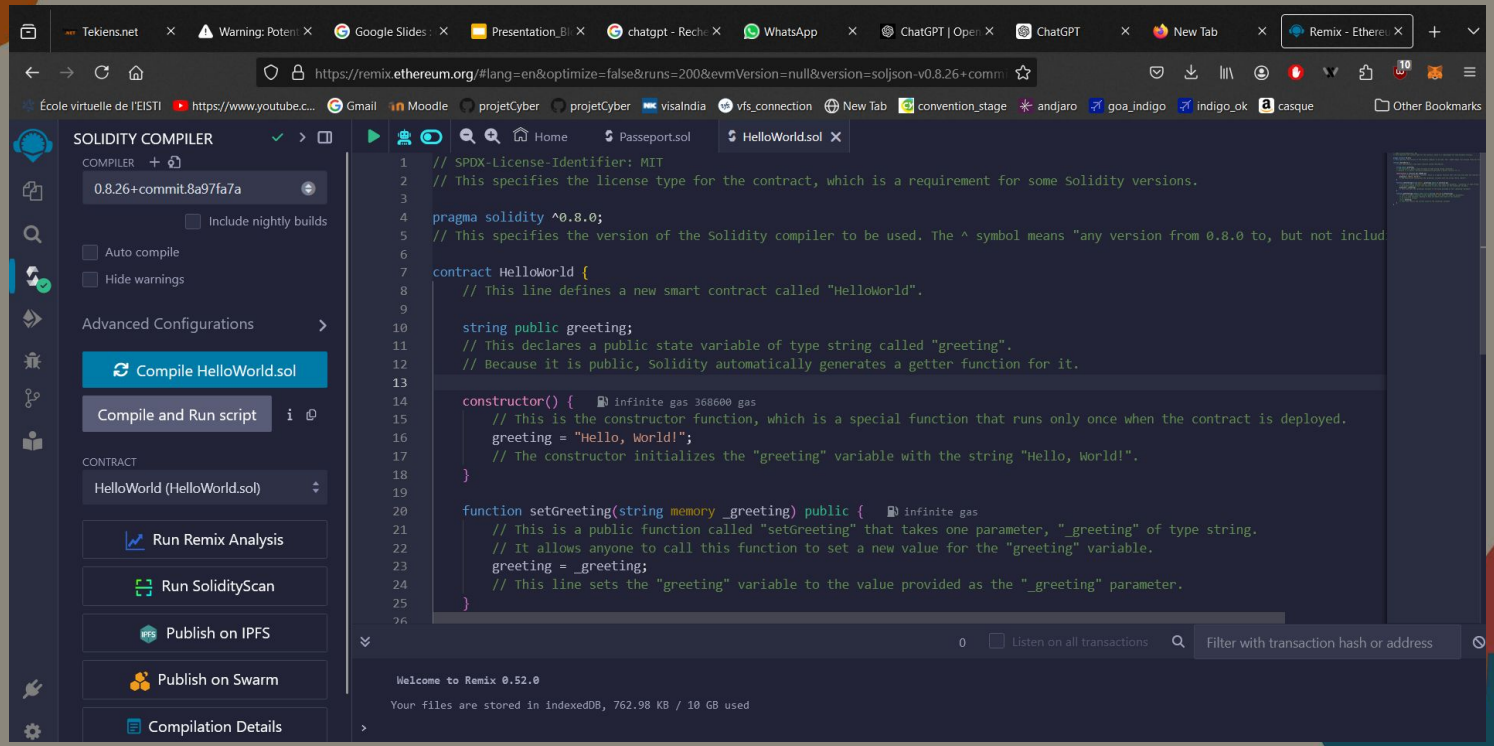
Developing Smart Contracts with Ethereum IDE

-Overview of Ethereum IDE (such as Remix): An online editor for writing and deploying smart contracts.
Key features: Code editor, compiler, and deployment tools.
Steps to develop a contract:

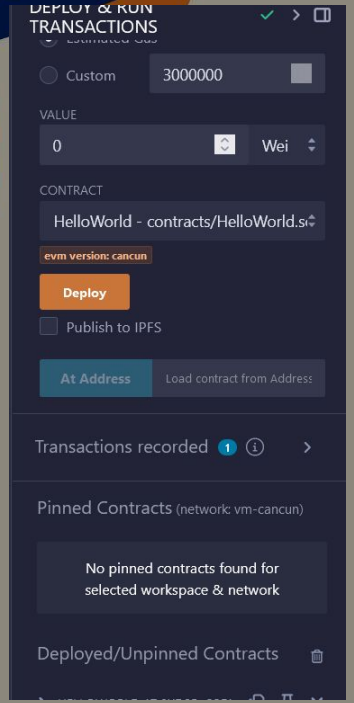
- Writing the smart contract code in the IDE.
- Compiling the contract using the built-in compiler. 

Example: Writing and compiling a simple "Hello World" smart contract :

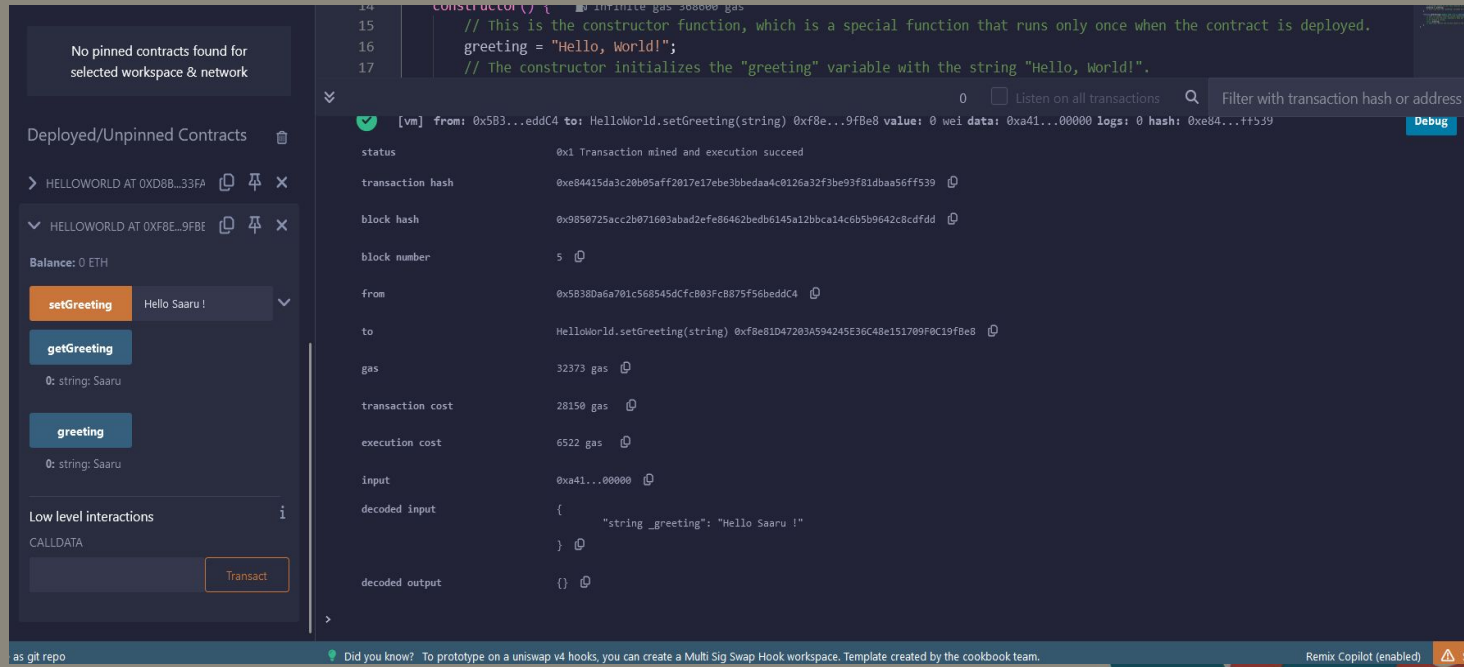
Using Truffle, Ganache, and Creating Smart Contracts, and Deploying on Ethereum IDE



Using Truffle, Ganache, and Creating Smart Contracts, and Deploying on Ethereum IDE



After Compiling



After deploy

Using Truffle, Ganache, and Creating Smart Contracts, and Deploying on Ethereum IDE

Transitioning to Truffle for Advanced Development

TRUFFLE

-What is Truffle? A development framework for Ethereum that provides a project structure and automation tools.
Key Features:

- Simplifies contract development.
- Provides automated testing and migration scripts.

-Steps:Install Truffle: npm install -g truffle.

- Initialize a Truffle project with truffle init.
- Import the smart contract from Ethereum IDE.

Using Truffle, Ganache, and Creating Smart Contracts, and Deploying on Ethereum IDE

Writing and Testing Smart Contracts with Truffle

-Writing Smart Contracts:

- Write smart contracts in Solidity within the Truffle project.

-Compiling Contracts: Compile contracts using truffle compile.

-Testing Contracts: Write and run tests using truffle test.

Using Truffle, Ganache, and Creating Smart Contracts, and Deploying on Ethereum IDE

Setting Up and Using Ganache

-What is Ganache? A personal blockchain for Ethereum development that allows for faster and easier testing.
Key Features:

Provides a local blockchain for development.

Allows for transaction logging and inspection.

-Steps:

Install Ganache (via Ganache CLI or Ganache GUI).

Configure and start the local blockchain.

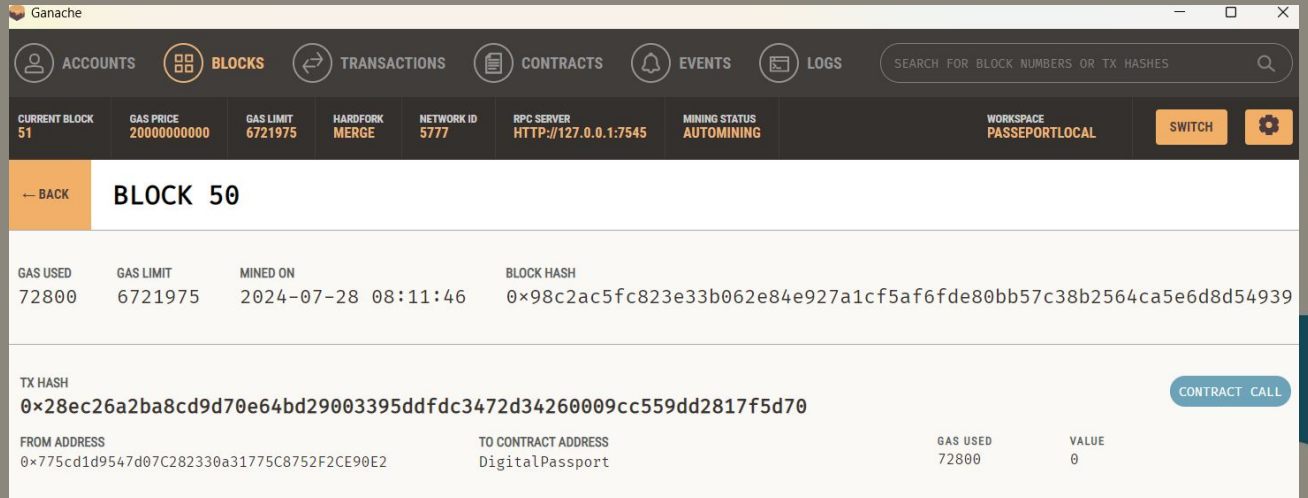
Connect Truffle to Ganache with truffle-config.js.

Using Truffle, Ganache, and Creating Smart Contracts, and Deploying on Ethereum IDE

```
//
development: {
  host: "127.0.0.1",
  port: 7545,
  network_id: "*",
},
```

Quick Demo

```
  }
}
truffle(development)> let passport = await instance.getPassport(accounts[0]);
undefined
truffle(development)> console.log("Passport:", passport);
Passport: Result {
  '0': 'John Doe',
  '1': 'American',
  '2': '123456789',
  '3': '1990-01-01',
  '4': 'New York',
  '5': BN {
    negative: 0,
    words: [ 14643264, 24, <1 empty item> ],
    length: 2,
    red: null
  },
  '6': BN {
    negative: 0,
    words: [ 10606400, 25, <1 empty item> ],
    length: 2,
    red: null
  }
}
undefined
truffle(development)> await instance.updatePassport("John Smith", "American", "123456789", "1990-01-01", "Los Angeles", 1625256000, 1688320000, { from:
 accounts[0] });
{
  tx: '0x28ec26a2ba8cd9d70e64bd29003395ddfdc3472d34260009cc559dd2817f5d70',
  receipt: {
    transactionHash: '0x28ec26a2ba8cd9d70e64bd29003395ddfdc3472d34260009cc559dd2817f5d70',
    transactionIndex: 0,
    blockNumber: 50,
    blockHash: '0x98c2ac5fc823e33b062e84e927a1cf5af6fde80bb57c38b2564ca5e6d8d54939',
    from: '0x775cd1d9547d07c282330a31775c8752f2ce90e2',
    to: '0xd1b5a6b0659863c2732d7fa9da9f65b7dde7',
    cumulativeGasUsed: 72800,
```



Using Hyperledger Aries and Creating a React Platform to Interact with the Blockchain

Introduction to Hyperledger Aries

-**Overview**: Hyperledger Aries is a project within the Hyperledger ecosystem aimed at building interoperable, decentralized identity solutions.

-**Purpose**: It facilitates secure peer-to-peer interactions, identity management, and the issuance and verification of verifiable credentials.

-**Ecosystem**: Aries integrates with other Hyperledger projects such as Hyperledger Indy for decentralized identity and Hyperledger Ursa for cryptographic functions.

Using Hyperledger Aries and Creating a React Platform to Interact with the Blockchain

Core Components of Hyperledger Aries

-**Aries Framework**: Provides a set of tools and libraries for developers to create identity solutions.

-**Agents**: Software entities that manage identities, credentials, and interactions on behalf of individuals or organizations.

-**Protocols**: Define standard methods for secure communication and credential management between agents.

-**Wallets**: Secure storage solutions for keys and credentials, ensuring the privacy and integrity of identity data.

Using Hyperledger Aries and Creating a React Platform to Interact with the Blockchain

Verifiable Credentials and Proofs

-Verifiable Credentials: Digital representations of credentials that can be issued, held, and verified.

-Issuing and Holding: The process of creating, issuing, and storing credentials using Aries.

-Presenting Proofs: Mechanisms for holders to prove possession of credentials without revealing sensitive data, ensuring privacy.

Using Hyperledger Aries and Creating a React Platform to Interact with the Blockchain

Building with Hyperledger Aries

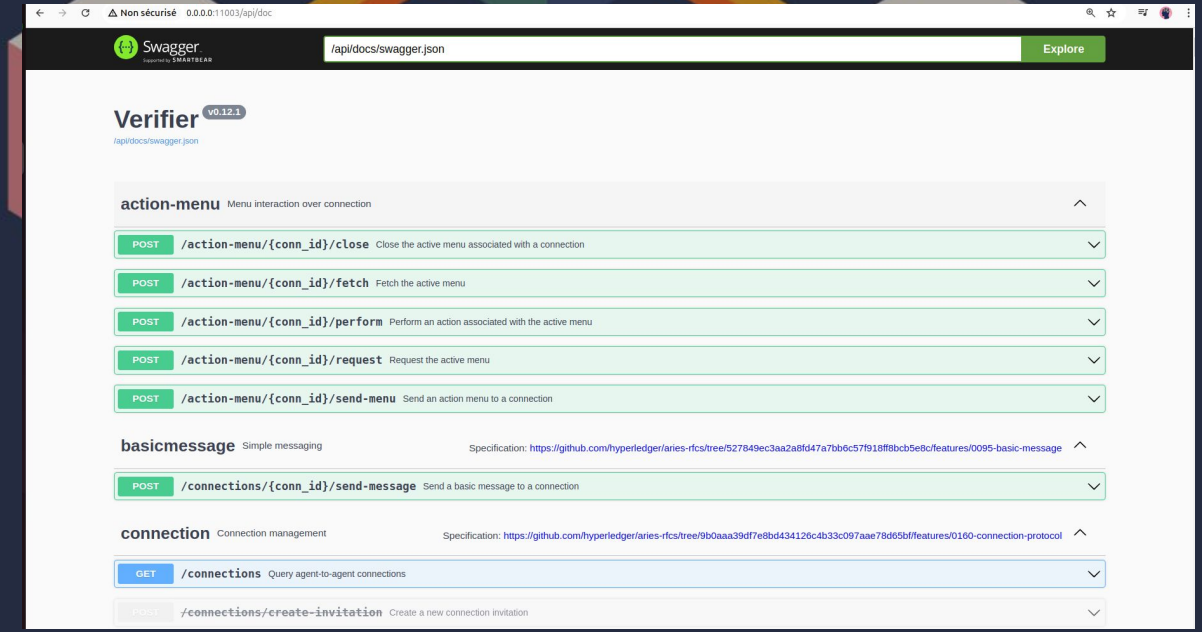
-**Development Tools**: Overview of the SDKs and APIs available for building applications with Aries.

-**Integration**: Methods for integrating Aries with other blockchain solutions, such as Hyperledger Indy for decentralized identity management.

-**Example Use Cases**: Applications in various industries, including identity verification, supply chain management, and financial services.

Using Hyperledger Aries and Creating a React Platform to Interact with the Blockchain

Demo

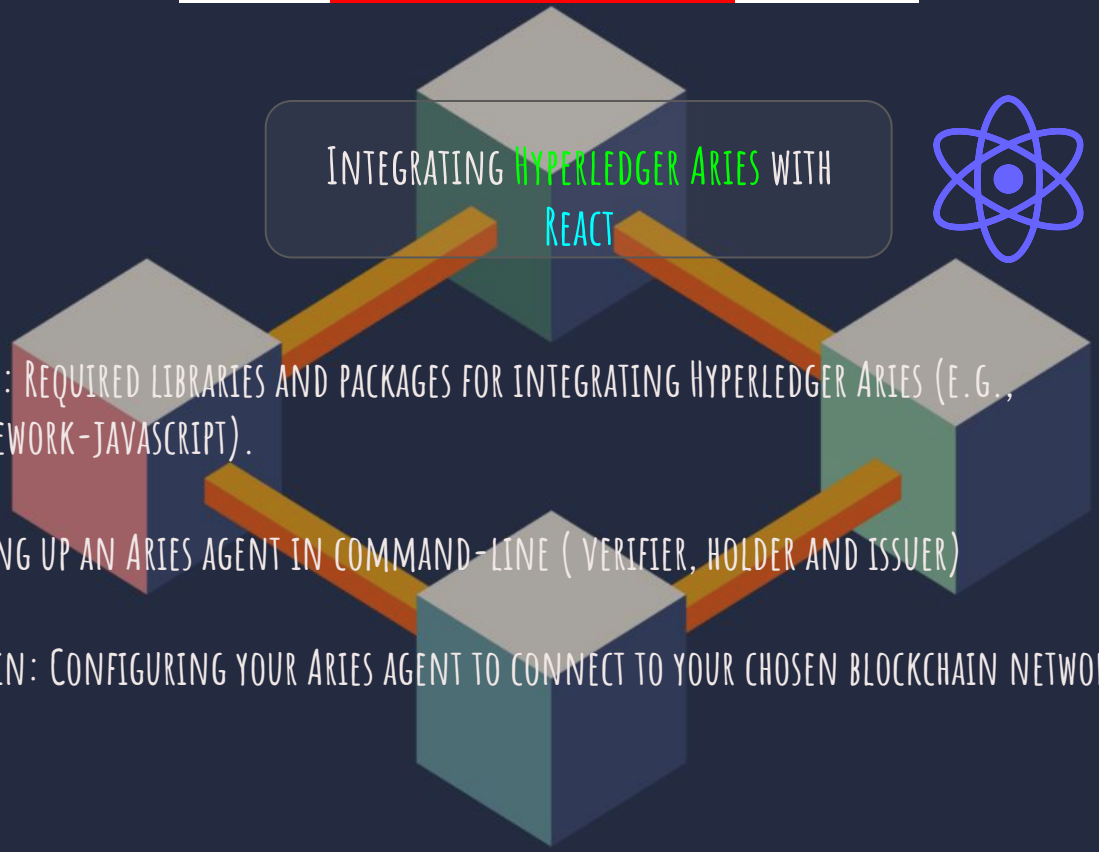


Using Hyperledger Aries in React

Setting Up a React Project

- Introduction to React as a powerful frontend framework for building interactive user interfaces.
- Creating a new React project using Create React App.
- Organizing your React project to facilitate integration with Hyperledger Aries.

Using Hyperledger Aries in React



Integrating Hyperledger Aries with React

-Installing Dependencies: Required libraries and packages for integrating Hyperledger Aries (e.g., @hyperledger/aries-framework-javascript).

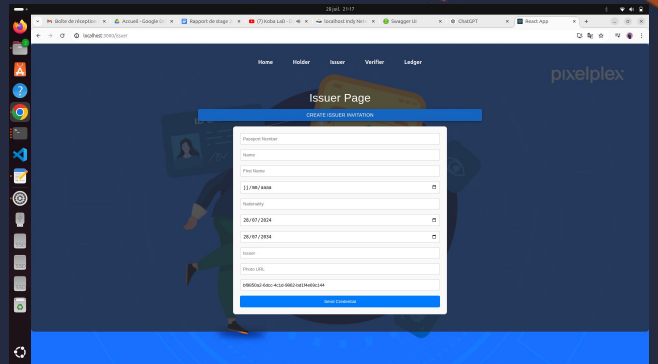
-Initializing Aries: Setting up an Aries agent in command-line (verifier, holder and issuer)

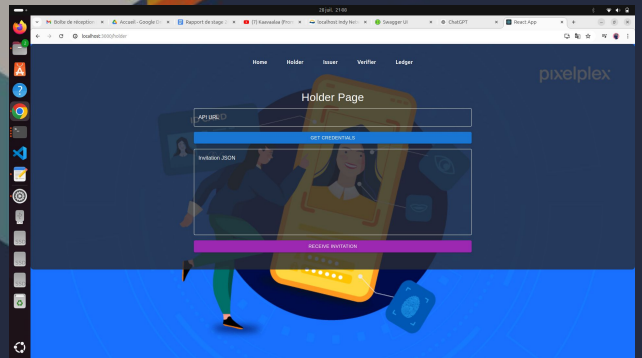
-Connecting to Blockchain: Configuring your Aries agent to connect to your chosen blockchain network for identity management.

Using Hyperledger Aries in React

Developing the User Interface

- **Creating Components:** Building React components to handle user interactions and display data.
- **Managing State:** Utilizing React state management solutions (e.g., Redux, Context API) to manage Aries-related data.
- **UI Elements:** Designing forms and interfaces for credential issuance, display, and proof presentation.





Using Hyperledger Aries in React