

# Internship report 2023/2024



Laboratory Internship Tutor: Mr. Pradeep Singh Shekhawat  
School internship tutor: Mr. Mohamed Haddache

**Subject: Secure and Decentralized  
Identity Management (Blockchain)**

	Internship report 2023/2024	Saaruhan SELLAPPAH
---	--------------------------------	-----------------------

## **Summary**

Acknowledgments.....	3
Presentation of BK BIET.....	4
Introduction.....	5
Part 1.....	6
I.A).....	6
I.B).....	7
I.C).....	8
I.D).....	10
Part 2.....	11
II.A).....	11
II.B).....	12
II.C).....	12
II.D).....	15
Conclusion.....	18

	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

## Aknowledgements,

Mr Pradeep Singh Shekhawat, Mr Mohamed Haddache,

I would like to express my deepest gratitude for the opportunity you gave me to complete this internship on Secure and Decentralized Identity Management. Your expertise, support and wise advice were essential to the success of this project. Mr. Shekhawat, your laboratory supervision has been invaluable, providing me with a practical perspective and technical skills that I will cherish throughout my career. Mr. Haddache, your educational support allowed me to better understand the theoretical and methodological issues of this innovative field. Thanks to you, I was able to deepen my knowledge of blockchain and develop concrete solutions for real problems. I sincerely thank you for your commitment and kindness throughout this enriching experience.

With all my gratitude,

Saruhan Sellappah

	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

## Presentation of BK BIET :

B.K. Birla Institute of Engineering & Technology (BKBIET) is a prestigious private engineering college located in Pilani, Rajasthan, India. Established in 2007 by the Krishnarpan Charity Trust, it was chaired by the late industrialist Basant Kumar Birla. The institute offers undergraduate and postgraduate programs in various engineering disciplines, including Computer Science, Electronics, and Mechanical Engineering. BKBIET is affiliated with Bikaner Technical University and is approved by the All India Council for Technical Education (AICTE). The campus is equipped with modern facilities, including state-of-the-art laboratories, a central library, and advanced computing resources. It also emphasizes research and innovation, providing opportunities for students to engage in projects and internships with industry leaders. The institute has a strong placement record, with many graduates securing positions in top companies globally. BKBIET fosters a vibrant campus life with various extracurricular activities, clubs, and sports facilities, promoting holistic development. The college is committed to providing quality education and developing skilled engineers who contribute to society's technological advancement.

At BKBIET, under the mentorship of Mr. Pradeep Singh Shekhawat, I worked on the project titled Secure and Decentralized Identity Management. I started by understanding the fundamentals of blockchain and the concept of self-sovereign identity (SSI). Then, I designed a framework to create SSI on a blockchain platform, incorporating verifiable credentials (VCs) from trusted entities like governments and universities. I implemented user-controlled private attributes to ensure data privacy and security. Collaborating with peers, I utilized the college's advanced computing resources and labs for development and testing. Regular consultations with Mr. Shekhawat provided guidance and progress reviews. Finally, I documented the entire process and findings, presenting my work to the department.

	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

## **Introduction**

Blockchain is an innovative technology that is radically changing the way we manage and secure digital data. It is based on a distributed and immutable ledger, decentralized on a network of nodes. Each transaction or data added to the blockchain is cryptographically secured and validated by consensus, ensuring its transparency, integrity and fraud resistance. Since its emergence with Bitcoin in 2008, blockchain has evolved to find applications in various areas, including decentralized identity management.

Decentralized Identity Management is based on the principle that individuals should have full control of their identity information, without relying on a central authority. In traditional systems, identities are managed by trusted third parties such as governments, businesses or financial institutions, who store and verify the information. However, this centralization poses risks, including cyberattacks, fraud and privacy violations. Blockchain offers a solution by allowing the creation of self-sovereign identities, where each user controls and manages their own identity data in a secure and private manner.

To implement a decentralized identity management system, it is necessary to use blockchain-specific tools and frameworks. My project started with Truffle and Ganache, essential tools in the Ethereum ecosystem. Truffle is a development environment, testing framework, and asset pipeline for Ethereum, facilitating the creation and deployment of smart contracts. Ganache, a customizable local blockchain, is ideal for development and testing. Using these tools, I developed and tested smart contracts to manage digital identities in a secure and transparent manner.

However, decentralized identity management requires more than just development infrastructure. It also requires robust protocols to manage verifiable credentials and trust relationships between different parties. This is where Hyperledger Aries comes in. Hyperledger Aries provides development tools for self-sovereign identity management, facilitating interoperability between different blockchains and decentralized identity systems. By integrating Hyperledger Aries into my project, I was able to create a complete ecosystem for secure identity management, enabling confidential and reliable verification and sharing of information.

In conclusion, blockchain, as a distributed ledger technology, offers an innovative way to manage identities in a secure and decentralized manner. By combining the development capabilities of Truffle and Ganache with the identity management protocols of Hyperledger Aries, it is possible to create identity systems that respect user privacy while providing optimal security. My project on Secure and Decentralized Identity Management shows how these technologies can be integrated to transform the management of identity information in the digital age.

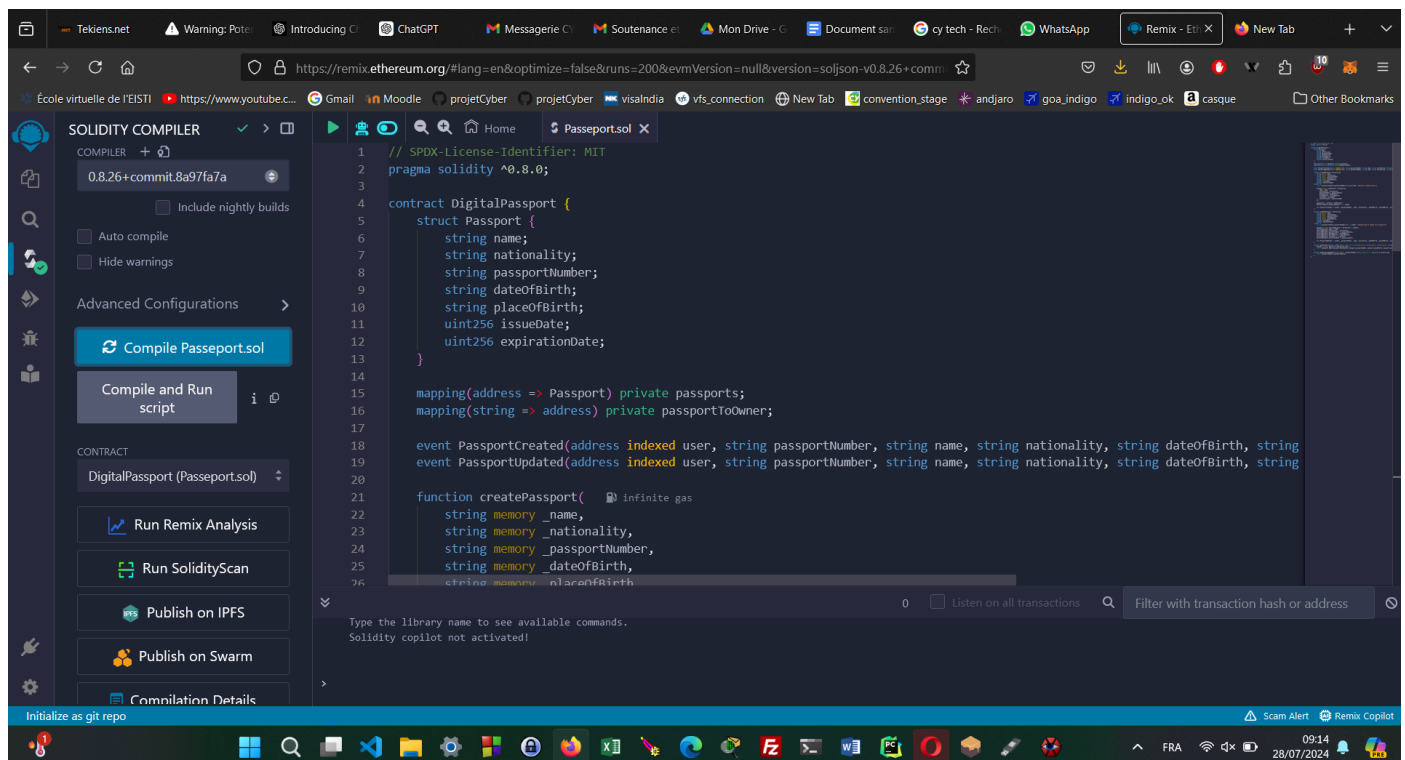
	Internship report 2023/2024	Saaruhan SELLAPPAH
---	--------------------------------	-----------------------

## Part 1: Using Truffle, Ganache and creating Smart Contract and deploying on Ethereum IDE:

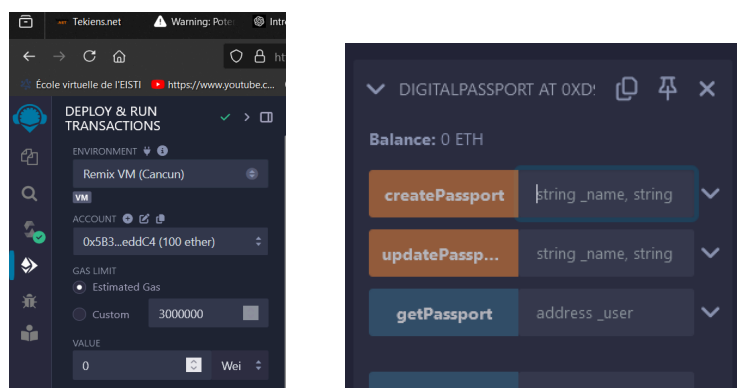
link to the github of the first project: <https://github.com/RolexSir30/passeportLocalBlockchain>

### A) Testing smart contracts on Remix IDE.

A smart contract is a computer program that runs automatically when certain predefined conditions are met. Deployed on a blockchain, it makes it possible to facilitate, verify and apply the terms of an agreement without requiring an intermediary. Smart contracts are immutable and transparent, ensuring that transactions are secure and reliable. I created a smart contract to simulate creating and updating a passport using Remix IDE. This contract allows you to save the initial information of a passport and modify it later. The Remix IDE tool was used to develop and test this smart contract efficiently:



Screenshots of the Remix IDE platform



	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

This contract allows you to save the initial information of a passport and modify it later. The last two photos I sent show the process of deployment and testing of the functions implemented in the smart contract. Using Remix IDE, I was able to deploy the contract on a local blockchain and perform its various functions. This allowed me to verify that the passport creation and update operations are working correctly. The Remix IDE user interface facilitates this testing by providing immediate and detailed feedback. This step is crucial to ensure that the smart contract meets expectations and is error-free.

However, Remix IDE has its limitations. I cannot deploy the smart contract locally, and deploying on the main network requires paying fees. I tried to get free ether using MetaMask, but it proved difficult. Faucets, which offer small amounts of ether for free, are often saturated or limited. This constraint complicates deployment and testing on the main network. Finding solutions to obtain ether remains a challenge.

So I was forced to go through a local blockchain network, Ganache.

## B) Compilation and deployment on Ganache.

Ganache is a local blockchain used primarily for the development and testing of smart contracts. It allows developers to simulate an Ethereum blockchain environment on their own machine. With Ganache, it is possible to deploy, execute and test smart contracts quickly without requiring gas fees or a connection to a main network. It also provides tools for viewing and managing transactions and contract statuses. Ganache thus facilitates development by providing a controlled and repeatable environment to detect and correct errors before deployment on a public blockchain. The deployment of smart contracts is free.

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK51

GAS PRICE2000000000

GAS LIMIT6721975

HARDFORKMERGE

NETWORK ID5777

RPC SERVERHTTP://127.0.0.1:7545

MINING STATUSAUTOMINING

WORKSPACEPASSEPORTLOCAL

SWITCH

MNEMONIC

hand entire aunt april victory pretty fiscal universe endorse firm squirrel hen

HD PATH

m44'60'0'0account\_index

ADDRESS

0x775cd1d9547d07C282330a31775C8752F2CE90E2

BALANCE

99.97 ETH

TX COUNT

48

INDEX

0

ADDRESS

0x3B5dE3Cde58Ce09aB9f537b07f0795174e55A7e0

BALANCE

100.00 ETH

TX COUNT

3

INDEX

1

ADDRESS

0xf8776cCDaF1c46B4F33902953c61f418800D6C9a

BALANCE

100.00 ETH

TX COUNT

0

INDEX

2

ADDRESS

0x2Df97a518Edf20b02F04AF53d54653B1d4516845

BALANCE

100.00 ETH

TX COUNT

0

INDEX

3

ADDRESS

0xe71046090a7B1CBCaC19b05C225Eb3c9f65052d2

BALANCE

100.00 ETH

TX COUNT

0

INDEX

4

ADDRESS

0x289c23A8838281b72EC7a754Dd00e1cDd80d8323

BALANCE

100.00 ETH

TX COUNT

0

INDEX

5

ADDRESS

0xFAC29d85F11d1CfD6c662343e566424AE4Dc5A41

BALANCE

100.00 ETH

TX COUNT

0

INDEX

6

	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

*Screenshot of the different Ethereum accounts that can be used for the deployment of smart contracts.*

On the Ganache blockchain, Ethereum accounts are automatically generated to facilitate development and testing. Each account is associated with a unique address and has an initial amount of simulated ether, eliminating gas fees. These accounts allow developers to deploy smart contracts and simulate transactions without real cost. Ganache offers an interface to manage these accounts, view transactions and monitor contract status. This simplifies the development process by providing a secure and controlled environment for testing decentralized applications.

In order to interact with this blockchain I use a development framework called truffle. Truffle is a development framework for the Ethereum blockchain that simplifies the process of creating and managing smart contracts. It offers an integrated toolset to compile, deploy and test smart contracts efficiently. Truffle also includes an automated test suite, a migration manager to deploy contracts to different networks, and an interactive console to interact with deployed contracts. In summary, Truffle facilitates the development, management and updating of smart contracts, making the blockchain development process more accessible and organized.

### C) Using Truffle.

Installing and launching the Truffle project is described in the github project Read.me file ( <https://github.com/RoloxSir30/passeportLocalBlockChain> ).

Attached are the steps taken to test the basic functions of the smart contract on the Ganache platform in:

Project compilation:

```

PS H:\Desktop\passeportLocal\client> truffle compile

=====
> Compiling .\contracts\Passeport.sol
> Artifacts written to H:\Desktop\passeportLocal\client\src\contracts
> Compiled successfully using:
   - solc: 0.8.0+commit.c7dfd78e.Emscripten.clang
PS H:\Desktop\passeportLocal\client>
PS H:\Desktop\passeportLocal\client>

```

Deployment on the platform Ganache :



	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

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

These configuration lines in the file `truffle-config.js` specify deployment settings for the local development network:

`host: "127.0.0.1"` sets the IP address of the local development server.

`port: 7545` indicates the port on which Ganache

`network_id: "*"`  allows Truffle to connect to any network, making it easy to connect to Ganache or other test networks.

Attached is the result of the migration.

```
PS H:\Desktop\passeportLocal\client> truffle migrate

Compiling your contracts...
=====
> Compiling .\contracts\Passeport.sol
> Artifacts written to H:\Desktop\passeportLocal\client\src\contracts
> Compiled successfully using:
   - solc: 0.8.0+commit.c7dfd78e.Emscripten.clang

Starting migrations...
=====
> Network name:    'development'
> Network id:     5777
> Block gas limit: 6721975 (0x6691b7)

2_deploy_contracts.js
=====

Replacing 'DigitalPassport'
-----
> transaction hash: 0x4a8d01c01e938c40a9433893f943bee22f1499fce630
3d7da1619461425b5a87
> Blocks: 0        Seconds: 0
> contract address: 0xD1054A060659863cA732d7F7A5b4E9f654b7DDe7
> block number:    48
> block timestamp: 1722145402
```



	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

to this platform, I decided to move towards a framework that has already integrated all of these features: Hyperledger Aries. Secure and complete.

## **Part 2: Using Hyperledger Aries and creating the React platform to interact with the Blockchain platform.**

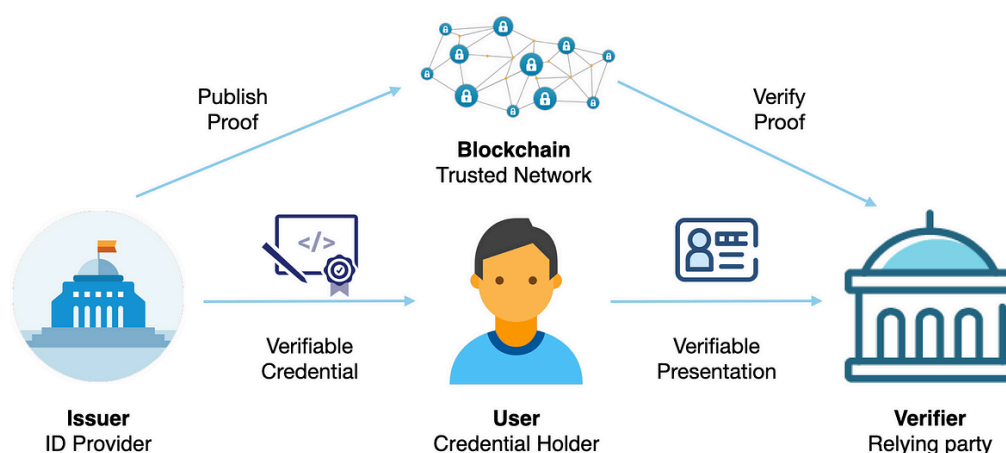
Here is the link to the final project: <https://github.com/RolexSir30/ProjetInde/tree/master>

### **A) The 3 roles of a decentralized identity management system.**

In a Decentralized Identity Management system, three main roles play a crucial role: the holder, the issuer and the verifier. Each of these roles has distinct responsibilities in the identity management and verification process.

The holder is the individual or entity that owns and controls its own decentralized identity wallet. This wallet contains Verifiable Credentials received from issuers. The holder is responsible for managing and storing this information securely, usually using technologies like digital wallets. When it is necessary to prove their identity or qualifications, the holder may choose to share some of their identity information with verifiers, ensuring that such sharing respects their privacy and preferences.

The issuer, or issuer, is a trusted entity that creates and issues verifiable credentials to holders. This information may include diplomas, birth certificates, professional licenses



	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

## B) Setting up and installing Hyperledger Aries.

You can find the steps for installing hyperledger aries and aries cloud agent python on the project's github in the Read.me section.

## C) Creation of agents, verifier and issuer and integrated platform.

In order to launch the application, the first step is to launch the ledger using the “./manage start” command. Attached are the commands to launch the agents:

### Issuer :

```
aca-py start --label Bob -it http 0.0.0.0 8001 -ot http --admin 0.0.0.0 11001
--admin-insecure-mode --genesis-url http://localhost:9000/genesis --seed
Issuer00000000000000000000000000000000 --endpoint http://localhost:8001/ --debug-connections
--auto-provision --wallet-name ISSUER --wallet-key secret --wallet-type askar
--replace-public-did --recreate-wallet --tails-server-base-url http://localhost:8080
--auto-respond-credential-proposal --auto-respond-credential-offer
```

Explanation of each option:

**Here is an explanation of the options used in the `aca-py start` command:**

1. **`--label Bob`** : Defines the agent label. Here the agent will be called "Bob".
2. **`-it http 0.0.0.0 8001`** : Configures the agent to use HTTP transport. The agent will listen on address `0.0.0.0` and port `8001`.
3. **`-ot http`** : Specifies that the agent's outbound transport will also be over HTTP.
4. **`--admin 0.0.0.0 11001`** : Activates the agent administration interface, accessible via HTTP on the address `0.0.0.0` and port `11001`.
5. **`--admin-insecure-mode`** : Allows access to the administration interface without authentication. This is only used for testing and development as this is not secure.
6. **`--genesis-url http://localhost:9000/genesis`** : Specifies the URL of the blockchain genesis file to use to connect to the network. The genesis file contains information necessary to establish the initial connection to the blockchain network.
7. **`--seed Issuer00000000000000000000000000000000`** : Uses the provided seed to generate a key for the agent's decentralized identity (DID). This is often used for testing or to configure known identities.

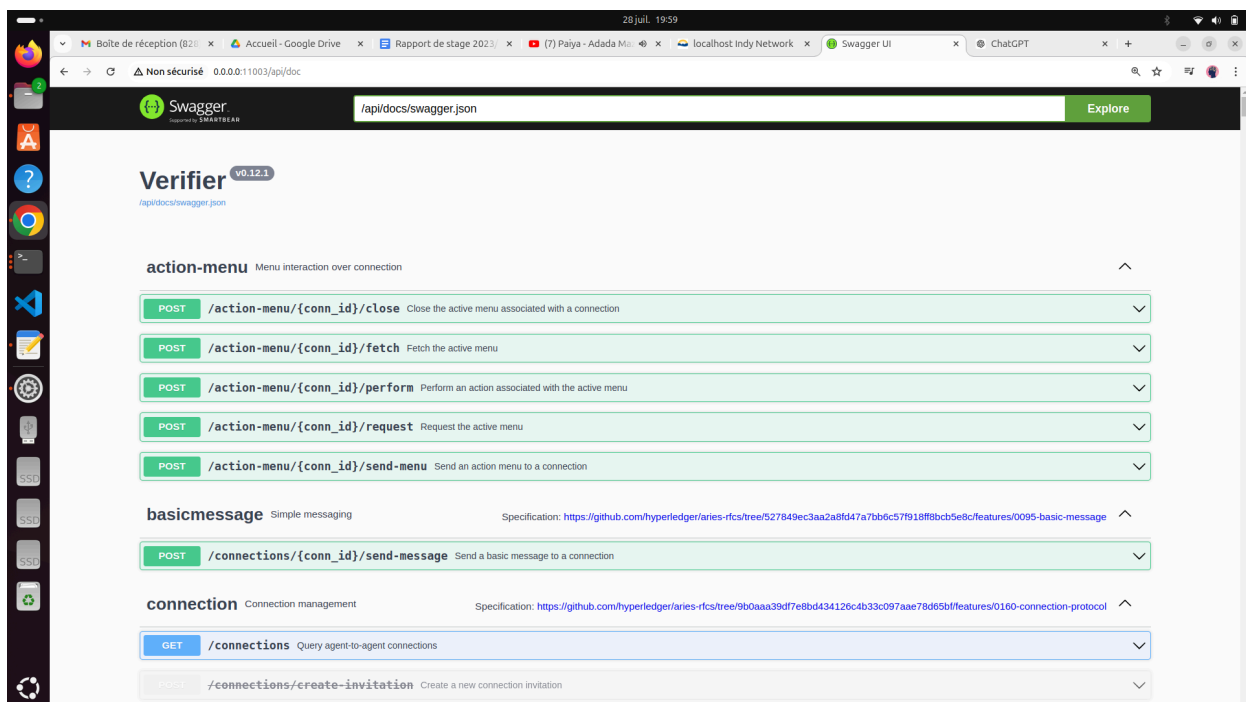


	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

New options:

**--auto-store-credential** : Configures the agent to automatically store received credentials. This option was not present in previous configurations.

Graphical interface delivered by the Aries hyperledger framework:



When I access the URL **http://0.0.0.0:11003/api/doc** for an Aries Cloud Agent - Python (ACA-Py)(the port differs depending on the agent), I obtain interactive documentation of the available APIs (see photo above). This documentation is typically provided by Swagger (or a similar tool), and it presents a list of all REST API functions that the agent exposes. Here is a general description of what you can expect from this documentation.

Core functionality available through the Aries Cloud Agent - Python (ACA-Py) API, including managing connections, issuing credentials, querying evidence, and creating schemas and credential definitions.

From this API I created a web application in React implementing these basic functions.

	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

## D) Web Application in React :

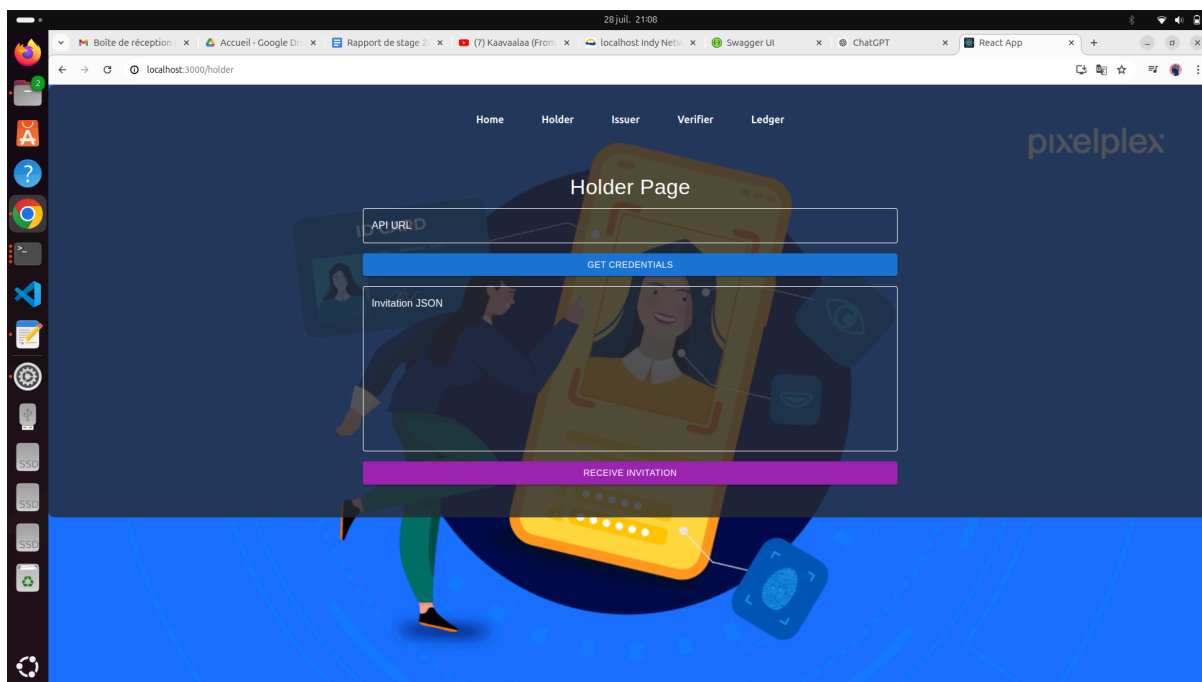
To interact with this API, I needed a visual interface and opted for React. I chose React for its flexibility, ease of managing state with Redux, and large ecosystem of libraries. The goal was to use the basic functions of Holder, Verifier and Issuer as requested in the internship subject. These functions include managing connections, issuing credentials, querying evidence, and creating schemas and credential definitions. React allowed me to create an interactive and responsive user interface, making it easy to interact with the ACA-Py API.

Before creating the buttons and the interface, I first created a schema of the credential on the API provided by Aries.

Voici les attributs du passeport : "date\_of\_issue\_dateint", "photo\_url", "passport\_number", "date\_of\_expiry\_dateint", "issuer", "firstname", "date\_of\_birth", "name", "nationality".

Installing react as well as creating a react project is described in the Github Read.me provided on page 9 just after the title part 2.

### 1) Holder Page

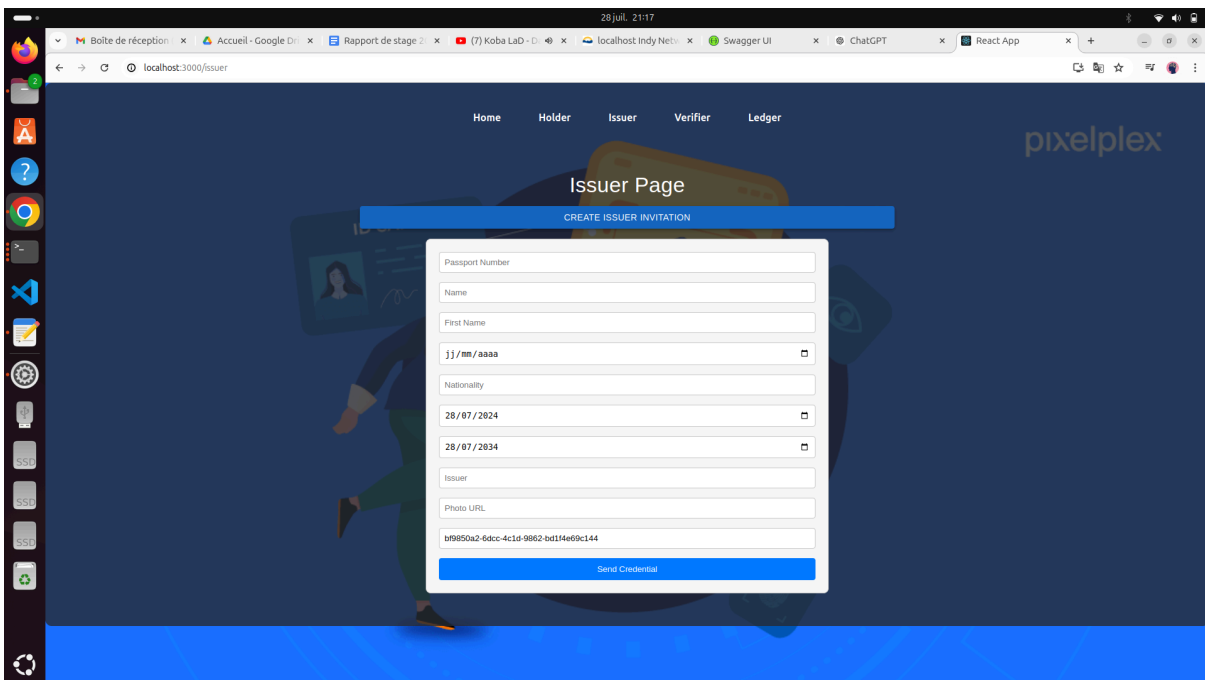


The holder has two main functions in its graphical interface. The first consists of recovering the credentials stored in your wallet. To do this, the user enters their admin API link and clicks on "Get Credential". This function allows the holder to easily view and manage their credentials in complete security. The second function allows you to create a connection with the verifier or issuer. To do this, the user pastes a JSON invitation created by these last two roles and clicks "Accept". This feature simplifies the login process by making interactions

	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

smooth and secure. Thanks to this intuitive interface, the holder can not only manage and retrieve their credentials, but also establish secure connections with other agents, thus facilitating the exchange of data and the necessary verifications. This approach improves the user experience by making decentralized identity management more accessible and efficient.

## 2) Issuer Page

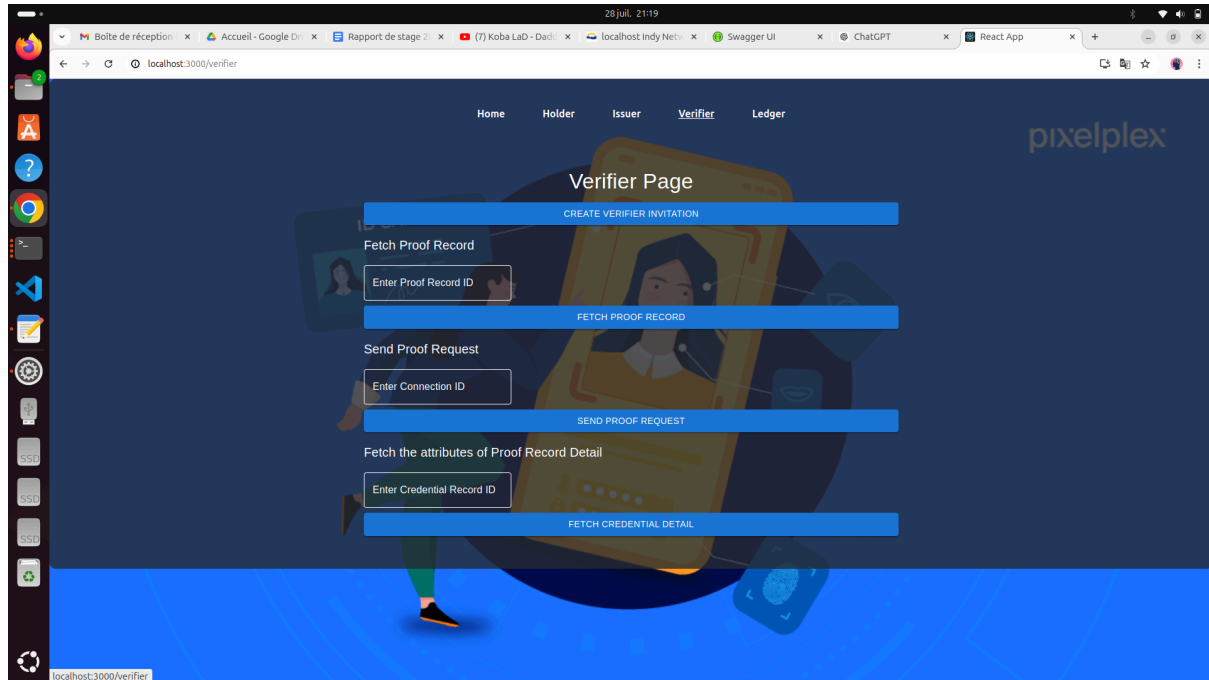


The issuer has two main roles in its GUI. The first is to create a new invitation to establish a secure connection with a holder. By generating this invitation, the issuer facilitates the connection process. The second role consists of sending a credential once the connection is established. This allows the issuer to deliver verifiable information efficiently and securely. Thanks to these two essential functions, the issuer can not only initiate connections with holders, but also issue credentials with ease. This interface makes the process of managing decentralized identities fluid and intuitive.

## 3) Verifier Page



	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------



The verifier has several essential features in its graphical interface. First, it can create an invitation to establish a secure connection with a holder. Then, the verifier can enter a proof request to request verifiable information from the holder. Once the query is sent, he can see the result and check the evidence provided. Finally, the verifier has the possibility of seeing the credentials of the holder from whom he requested proof, thus facilitating the process of verification and authentication of information. This intuitive interface allows the verifier to manage connections and evidence efficiently and securely.

In short, thanks to this web application, I created a graphical interface allowing the three main roles of a decentralized management system to interact in a fluid and efficient manner.

	<p>Internship report</p> <p>2023/2024</p>	<p>Saaruhan SELLAPPAH</p>
---	---	-------------------------------

### **Conclusion :**

In conclusion, I developed a web application using React to create an intuitive graphical interface, facilitating interaction with the Aries API. I first designed a credential scheme for a passport, then integrated essential functionality for the roles of holder, issuer, and verifier. The holder can retrieve and manage their credentials as well as create secure connections. The issuer can generate invitations and issue credentials. The verifier can create invitations, send proof requests, and review the holder's credentials. This interface allows decentralized identity management, improving the efficiency and security of the exchange of verifiable information. Initially, I had started a project with Truffle to develop smart contracts, but decided to abandon it to focus on the integration with Aries, which better met the needs of the project. My work resulted in a coherent and accessible system, meeting user needs in a decentralized identity management environment.

Saaruhan Sellappah

23 ans

CV



Paris



sellappah@cy-  
tech.fr



0781656990

Intérêt



Niveau Moyen

(6 ans de pratique)



[https://github.com/  
RoloxSir30](https://github.com/RoloxSir30)

## Expérience professionnelle et projets

### **Avril 2024 – Août 2024 : Expérience : Stage à BK BIET, Inde :**

Gestion sécurisée et décentralisée de l'identité

Création d'une identité souveraine (SSI) sur une plateforme blockchain, où chaque utilisateur a le contrôle total sur son propre identifiant en ligne, similaire à un passeport numérique. Attribution d'attestations vérifiables (VC) par des entités de confiance telles que gouvernements ou universités, ainsi que la gestion des attributs privés par l'utilisateur, permettant une authentification sécurisée et décentralisée des identités en ligne.

### **Février 2024 – Mars 2024 : Projet de compression sans perte en langage Haskell**

### **Janvier 2024 -Mars 2024 Projet : Application de messagerie sécurisée.**

Outils : Python, Flask.

Hashage de mot de passe, certificats SSL, chiffrement des messages asymétriques.

### **Novembre - Décembre 2023 : Projet : Création d'une application de e-commerce avec JEE et Spring Boot.**

Projet JEE basé sur une couche DAO (JDBC, Hibernate, JPA), une couche business et une couche web

JEE : <https://github.com/RoloxSir30/BazarWeb>

Spring : <https://github.com/Le-7/BazarWeb>

### **Mars 2023 : Projet : Compétition scolaire de création d'un jeu vidéo (3ème place)**

Création d'un jeu de Snake et d'un jeu de Brick Breaker en Java

### **Juillet 2022 – Août 2022 : Expérience : Stage au CLIP (Club informatique pénitentiaire)**

La mission consistait à enseigner des sujets liés à l'informatique, tels que la programmation web, aux détenus désireux de se réintégrer sur le marché du travail.

### **Février 2022 - Avril 2022 : Projet : Développement d'un jeu d'échec en JAVA :**

Conception UML du projet. Développement de l'interface graphique et des lignes de commande.

### **Janvier 2022 – Mai 2022 : Projet : Développement d'un site web d'un zoo.**

### **2021— : Expérience : Animateur périscolaire.**

## Formation

---

2021-2025	Cy-Tech – ex EISTI (Ecole Internationale des sciences du traitement de l’information) Formation d’ingénieur en informatique
2018-2021	Lycée Pierre Gilles de Gennes - ENCPB CPGE (Classes Préparatoires aux Grandes Écoles)

## Compétences

---

Langages : **SQL, Python, C, PHP, CSS, html ,Javascript, Java, JEE**  
OS : **Windows, UNIX**  
Logiciels : **Pack office Microsoft, Latispro, Regressi, Talend, Microsoft Power BI.**  
Base de données : **MySQL, SQLite, Mongo DB**  
Permis : **B ( voiture )**  
Brevet d’aptitude à la formation d’animateur.  
  
Brevet de secourisme : **PSC1.**

## Langues Vivantes

---

Tamoul : **Courant**  
Anglais : **Avancée (CEA Cambridge)**  
Espagnol : **Intermédiaire**  
Français : **Courant**

