**INSTALLATION AND USER INSTRUCTIONS**

# ChorSSI

# A model-driven approach for modeling and executing self-sovereign identity systems on Blockchain

A guide on executing the ChorSSI framework from a new user point of view.

Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente

Camerino, January 2024, University of Camerino.

Summary

[1 PREFACE 3](#_Toc156397436)

[1.1 Description of the user 3](#_Toc156397437)

[1.2 Obtaining documentation and information 3](#_Toc156397438)

[2 Description of the software 4](#_Toc156397439)

[2.1 Purpose of the software 4](#_Toc156397440)

[2.2 Software Compliance 4](#_Toc156397441)

[2.3 Software architecture 5](#_Toc156397442)

[2.3.1 Hyperledger Aries (Aries Cloud-Agent Python – ACApy) 5](#_Toc156397443)

[2.3.2 Hyperledger Indy 5](#_Toc156397444)

[2.3.3 Von-network 5](#_Toc156397445)

[2.3.4 Indy-SDK 5](#_Toc156397446)

[2.4 Requirements 6](#_Toc156397447)

[2.4.1 If using codespace 7](#_Toc156397448)

[2.5 How to use the software 7](#_Toc156397449)

[2.5.1 Manual operations if using a virtual machine 8](#_Toc156397450)

[2.5.2 Initialize and start the tool 9](#_Toc156397451)

[2.5.3 Modelling 10](#_Toc156397452)

[3 TROUBLESHOOTING AND REPAIR 14](#_Toc156397453)

[3.1 How to Identify and solve problems 14](#_Toc156397454)

[3.1.1 Docker engine 14](#_Toc156397455)

[3.1.2 Runtime errors 15](#_Toc156397456)

# PREFACE

This is a guide on how to use this framework from the beginning.

## Description of the user

The user for which the user manual is intended is a user who wants to easily manage a digital identity system. This is not intended to be used by more than one person.

The person who may manage this system should have some basic knowledge about digital identity systems, digital wallets, digital credentials and blockchain.

## Obtaining documentation and information

The latest version of the documentation is available at the following address: <https://github.com/LorenzoLuziL/ProgettoSystem>

Documentation, user instructions and technical information can be obtained by pulling the original repository from GitHub.

# Description of the software

## Purpose of the software

As we increasingly rely on digital platforms for communication, commerce, and  
other activities, the need for secure and reliable methods of identifying ourselves has  
become more important than ever. Traditional identity management systems, which  
are often centralized and controlled by third parties, have been criticized for being  
insecure and vulnerable to data breaches, fraud, and other forms of abuse. In fact,  
the storage of user identity data in multiple centralized data repositories with varying  
implementations of security present a tempting target for hackers, leading to increased  
security breaches and identity fraud. In addition, centralized data repositories present a  
lack of appropriate data management standards, resulting in potential privacy troubles.

Self-sovereign identity (SSI) is an emerging concept in the field of digital identity management that aims to empower individuals by providing them with more control and autonomy over their personal data and online identities.

The SSI paradigm shifts away from the traditional, centralized systems of identity management, where individuals are forced to rely on third-party providers to create, manage, and authenticate their digital identities. Instead, SSI proposes a decentralized and distributed architecture where individuals can create, own, and control their digital identities using cryptographic technologies such as blockchain.

This software is a model-driven framework that can replicate the behavior and reproduce all the typical operations of self-sovereign identity systems.

## Software Compliance

This software was originally built using the version 18.04 of Ubuntu <https://releases.ubuntu.com/18.04/> and version 14.17.5 of Node.

If you are using different versions of these, you could maybe meet some unexpected problems or errors.

## Software architecture

The software architecture is composed by these main elements, the main repository already includes the repositories that are explained here:

TODO foto software architecture

### Hyperledger Aries (Aries Cloud-Agent Python – ACApy)

This is an open-source toolkit within the Hyperledger ecosystem that provides a set of protocols and tools for creating, transmitting, and storing **verifiable digital credentials**. It enables individuals, organizations, and devices to exchange secure, private, and tamper-evident information with each other without relying on a central authority.

### Hyperledger Indy

Hyperledger Indy is a decentralized, open source **blockchain** platform that provides the infrastructure for building and using decentralized identity (**DID**) solutions. It is specifically designed to address the unique requirements of decentralized identity systems, such as privacy, security, and interoperability.

### Von-network

A von-network is a pre-packaged Indy network built by the Government of British Columbia’s Digital Identity and Trust team. It provides an easy way to run a local sandbox **Indy network using Docker containers** with minimal effort. The network is designed to enable digital identity solutions that allow people, organizations, and things to prove their identities to each other in a secure and decentralized manner.

TODO ci sono passaggi necessari da fare per la von? Magari riprovare da capo come fossi un utente inesperto… Considerando già il docker, python e il resto come inclusi…

### Indy-SDK

**This SDK has to be installed manually by the user** ( TODO siamo sicuri? step by step procedure described here: <https://yunxi-zhang-75627.medium.com/hyperledger-aries-aca-py-agents-setup-and-running-tutorials-part-i-i-i-dev-environment-setup-20ab5a32457e> ).

## Requirements

There are three mandatory software components to be included to let the software works properly.

If not installed yet (obviously) install Python & PIP, NVM and the Docker engine:

* **PYTHON & PIP**

sudo apt update

sudo apt install software-properties-common

sudo add-apt-repository ppa:deadsnakes/ppa

sudo apt update

sudo apt install python3.8

python --version

sudo apt update

sudo apt install python3-pip

pip3 --version

* **NVM ( Node 14.17.5 )**

sudo apt install curl

curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash

source ~/.profile

nvm install 14.17.5

* **DOCKER ENGINE**

sudo apt install install docker.io

Installing these applications could be usefull:

* GitHub Desktop, online download, or tipying:

sudo wget https://github.com/shiftkey/desktop/releases/download/release-3.1.1-linux1/GitHubDesktop-linux-3.1.1-linux1.deb

sudo apt-get install gdebi-core

sudo gdebi GitHubDesktop-linux-3.1.1-linux1.deb

* Visual Studio Code, online download, or:

sudo apt update

sudo apt install gnupg2 software-properties-common apt-transport-https wget

wget -q https://packages.microsoft.com/keys/microsoft.asc -O- | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main"

sudo apt update

sudo apt install code

### If using codespace

TODO

## How to use the software

Once this repository has been cloned into the local machine the tool is ready to be initialized.

### Manual operations if using a virtual machine

There are still some raw-level operations that the user need to do to use ChorSSI on a local Virtual Machine (use Codespace to avoid this ):

* In *APIUtils.js* file (./src/components/util/APIUtils.js):

Replace each line of this form

*url: "https://friendly-couscous-r444p94p66qg354v4-"+port+".app.github.dev/connections/create-invitation?auto\_accept=true&multi\_use=true",*

with:

*url: "http://localhost:"+port+"/connections/create-invitation?auto\_accept=true&multi\_use=true",*

* In *Server.js* file:

Replace each IP address in the file of this form 172.16.5.4 with *localhost*

* In *bpmn.modeler.component.jsx* file (./src/components/bpmn/bpmn.modeler.component.jsx):

replace *url:`https://friendly-couscous-r444p94p66qg354v4-${port}.app.github.dev`*

with *url:`https://localhost:${port} `*

* In *SpellProps.js* file (./src/lib/property-panel/provider/magic/parts/SpellProps.js):

Replace this line of code

*var url = "https://friendly-couscous-r444p94p66qg354v4-" + getPortByAgentName(parsedName) + ".app.github.dev";*

with

*var url = "https://localhost:" + getPortByAgentName(parsedName);*

### Initialize and start the tool

To set up the ChorSSI framework, the networks must be built up and the agents have to be created.

* The first step is building the von-network:

Starting from the main repository type

./von-network/manage build

This command builds docker images for the von-network.

* Then, open a bash terminal and inside the project’s folder run:

node Server.js

This one starts the server which starts the von-network and indy-tails servers.

* Then open another bash terminal to install the project’s dependencies and start the tool with:

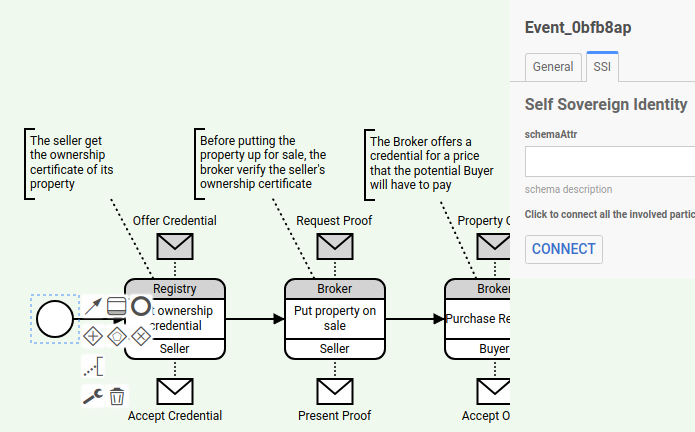
npm i

to install the dependencies and

npm run start

to start the tool.

* In the end, we have to click the “*CONNECT*” button. This feature set up the networks, create and connect the agents that are in the model (should be done after modelling phase, once the model is ultimated). It is placed in the *start* button in the *SSI* section as shown in the figure below:

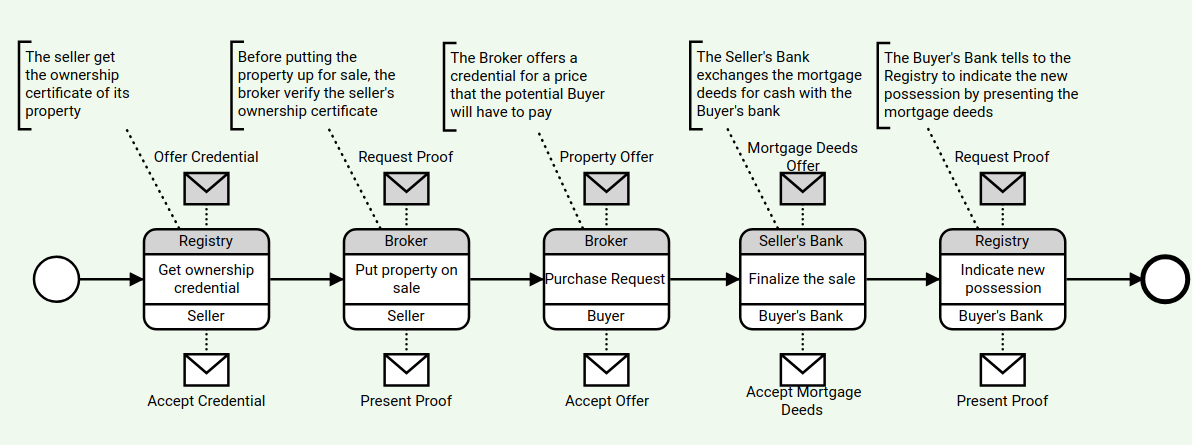
*”CONNECT” button in start event*

### Modelling

The technical complexity of self-sovereign identity technology is a barrier to its adoption, and a low-code approach would make it easier for users to understand and use the technology. In addition, it also suggests that by focusing on usage scenarios and organizational embedding, it will become more accessible for users.

For this reason, this tool has been implemented and includes these features to develop a real use-case model.

* **Modeler**

 *”Chromaway”, a real SSI use case.*

A user interface component that allows users to create, view, and edit **Choreography diagrams**. It provides a graphical user interface that enables users to interactively create and modify Choreography tasks and other elements.

Immagine che contiene schermata, diagramma, design

Descrizione generata automaticamente

* **Palette**

Refers to the collection of elements that can be added to a Choreography diagram from a side panel. The default palette in chor-js includes all the Choreography elements, such as tasks, events, gateways, and flows.

*Palette*

* Immagine che contiene testo, schermata, Carattere, numero

  Descrizione generata automaticamente**Property panel**

Allows users to view and edit the properties of BPMN elements in a BPMN diagram. It provides a way to customize the attributes of elements in the diagram and allows users to set or modify properties such as name, documentation, and implementation details.

*Property panel*

* **Diagram selector**

*Diagram selector*

**Immagine che contiene testo, schermata, Rettangolo, Carattere

Descrizione generata automaticamente**

it’s a simple form allowing users to manage the created diagrams.2.5 Monitoring the credentials

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamenteOn the Profile page (in the top-right panel), we can find a status bar containing all the agents that we have created from the model by clicking the *CONNECT* button.

From this status bar, we can select the agents correlated to the participant of the diagram of which we want to deeply observe some information details; the default agent whose additional information will be shown will be the first in the status bar itself. The shown information will depend on the object that the selected agent has generated:

Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamente*Profile section to handle the credentials*

• if the selected agent is an issuer we will for sure find the created credential definition, the linked schema object, and a reference to the issued credential with the possibility to revoke it.

• if the agent is a holder we will surely find the issued credential stored in their personal wallet.

• if the agent is a verifier we will find the objects regarding the verification of the proof presented by the holders. Note that an agent could act as both a verifier and an issuer and a holder without constraints, it depends on the specific use case model.

Todo magari continuare un po’

# TROUBLESHOOTING AND REPAIR

## How to Identify and solve problems

List of possible solutions to the problems that have been noticed for now.

### Docker engine

|  |  |  |
| --- | --- | --- |
| Error | Cause | Solution |
| connect: permission denied | User is not root | From root user:  sudo usermod -aG docker $USER |
| docker-compose: command not found | docker-compose is not installed | sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  # Make the file executable  sudo chmod +x /usr/local/bin/docker-compose  chmod +x /usr/local/bin/docker-compose |

*DOCKER ENGINE errors*

### Runtime errors

|  |  |  |
| --- | --- | --- |
| Error | Cause | Solution |
| ENOSPC: System limit for number of file watchers reached, watch '<path>/public' | Too many file watchers | *echo fs.inotify.max\_user\_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p*  # check that the new value was applied  *cat /proc/sys/fs/inotify/max\_user\_watches*  # config variable name (not runnable)  *fs.inotify.max\_user\_watches=524288* |

*Runtime errors*