CARDANO2VN.IO
BREAK THE BLOCKS

Atala PRISM Referent cases design

1. Document version

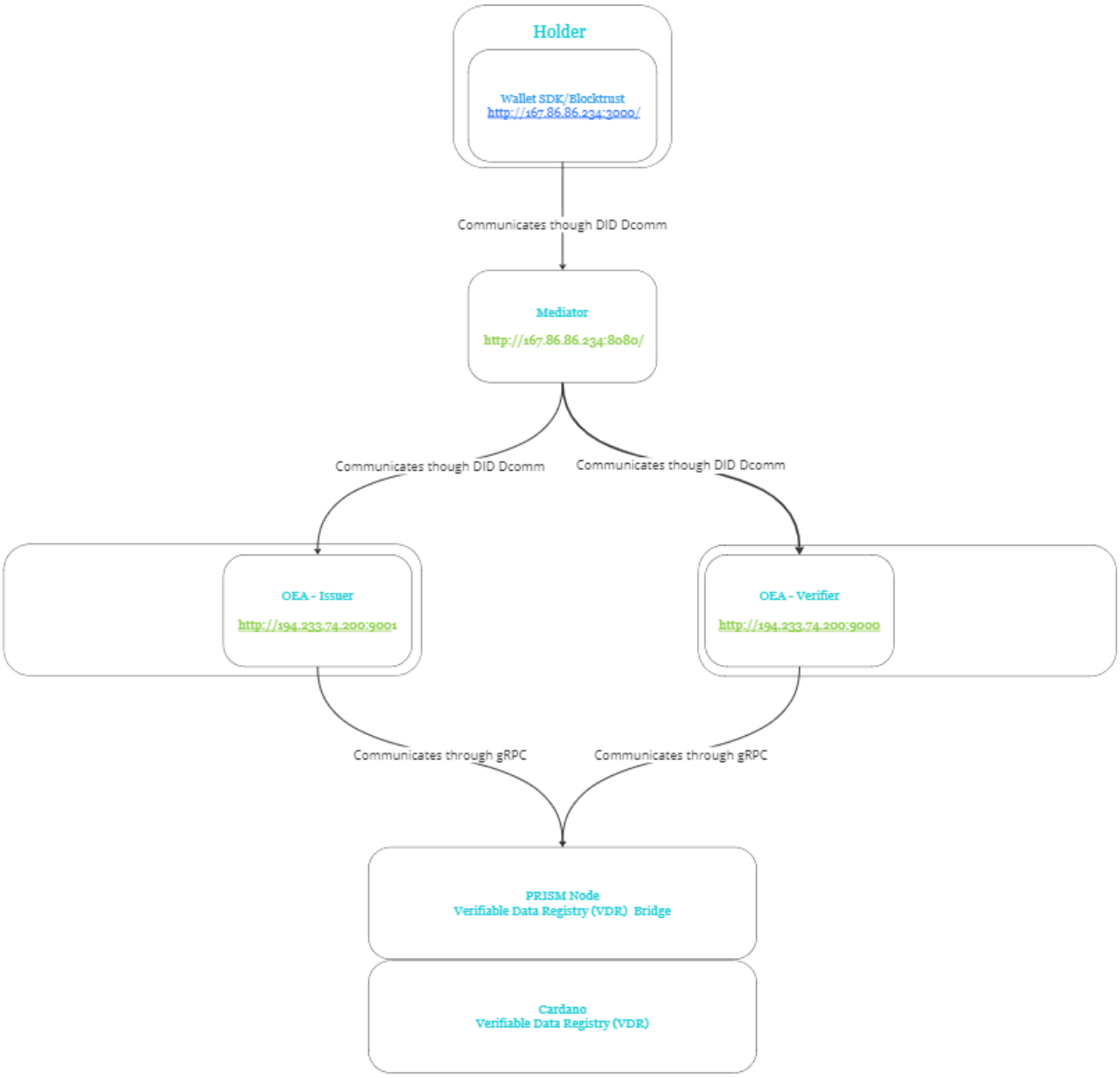
Version	Date	Author	Rationale
0.1	30 March 2023	Tien Nguyen Anh	First draft
0.2	1 April 2024	Tien Nguyen Anh	Review by Architect

2. Overview Architecture:

1.1 Purpose of design:

- This blueprint is for internal use by students taking the Decentralized Identity course - using Atala Prism.
- The design only mentions the basic functions required of a decentralized identification system.
- The purpose of the design is that when students practice through each step of the Referent case, they will gain basic experience in building a decentralized identification system.
- Lab parameters only work within the internal network. We do not provide services outside the classroom

1.2 Overview Architecture:



3. Core Services design:

3.1 Deploy Atala PRISM Wallet SDK

- Mediation server: 167.86.86.234
- OS: Ubuntu
- Link : http://167.86.86.234:3000
- Deployed as: Dockers
- Version: v5.0.0

3.2 Deploy Mediation

- Mediation server: 167.86.86.234
- OS: Ubuntu
- Link : http://167.86.86.234:8080
- Deployed as: Dockers
- Version: v0.14.1

3.3 Deploy OEA - Issuer Agent

- API_KEY_ENABLED=false
- PRISM_AGENT_VERSION=1.31.0
- PRISM_NODE_VERSION=2.2.1
- PORT=9001
- Link : http://194.233.74.200:9001
- Deployed as: Dockers
- Mediation server: 167.86.86.234
- OS: Ubuntu

3.4 Deploy OEA - Verifier Agent

- API_KEY_ENABLED=false

- PRISM_AGENT_VERSION=1.31.0
- PRISM_NODE_VERSION=2.2.1
- PORT=9000
- Link : http://194.233.74.200:9000
- Deployed as: Dockers
- Mediation server: 167.86.86.234
- OS: Ubuntu

4. Core functions design

4.1 Establish Mediation

Mediation is the process that ensures messages get routed and stored correctly between Issuers, Verifiers and Holders, even if they are offline. The mediator offers a service that is always running and can securely store messages and deliver them to the associated DIDs using DIDComm. This enables use-cases where connectivity to a (mobile) wallet cannot be guaranteed

4.2 Establish Holder connection

To connect the Holder to both Cloud Agent instances, you must run this in both Issuer and Verifier endpoints.

4.2.1 Establish a connection Holder - Issuer Agent side

A connection must be established between the Holder and PRISM Cloud Agents to correctly deliver the Issuance + Verification Messages to the Holder.

4.2.2 Establish connection on the Holder - Verifier Cloud Agent

4.2.3 Establish a connection - Holder side

When the holder has the invitation, it's time for the Holder to accept it.

4.3 Issue a Credential from the Issuer to the holder

In this scenario, lab will covers:

- Create a Credential Offer Issuer Agent
- Create CredentialRequest from CredentialOffer Holder
- Store the Issued Credential [Holder]

4.4 Request a verification from the Verifier Cloud Agent to the Holder:

In this topic, we will show demo of these topics:

- Verifier Agent
- Holder: Receives the Presentation proof request
- Verifier: Will then check on the API if the Presentation Request has been completed or not

5. Referent documents:

Blocktrust Identiy Wallet:

Proofspace Identity Wallet:

Atala Prisme Quick Start Guide