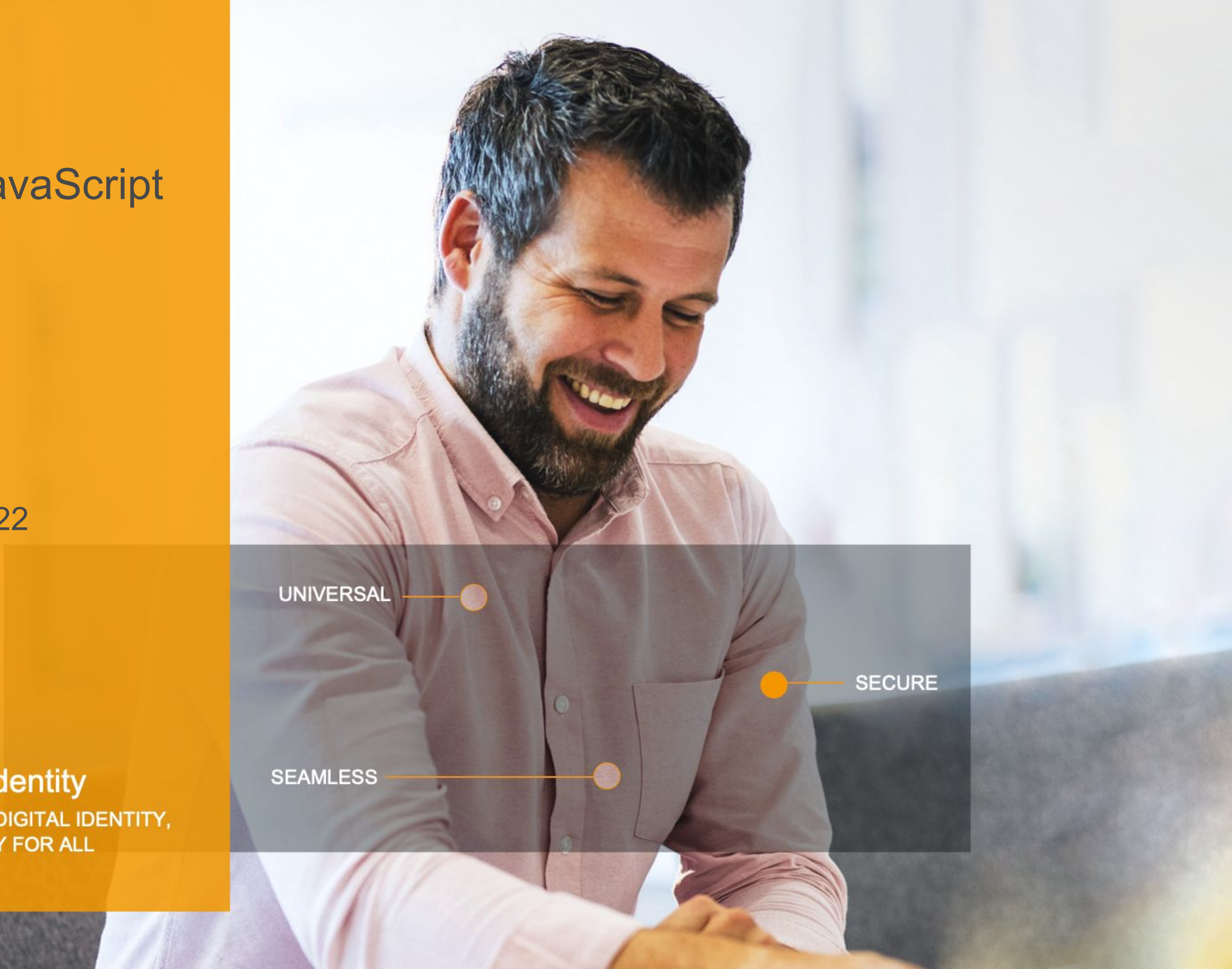
Workshop

Aries Framework JavaScript

Presented by:
- Berend Sliedrecht
- Karim Stekelenburg

Animo Solutions & Sicpa

12, 13 & 14 December 2022



SICPA

Digital Identity

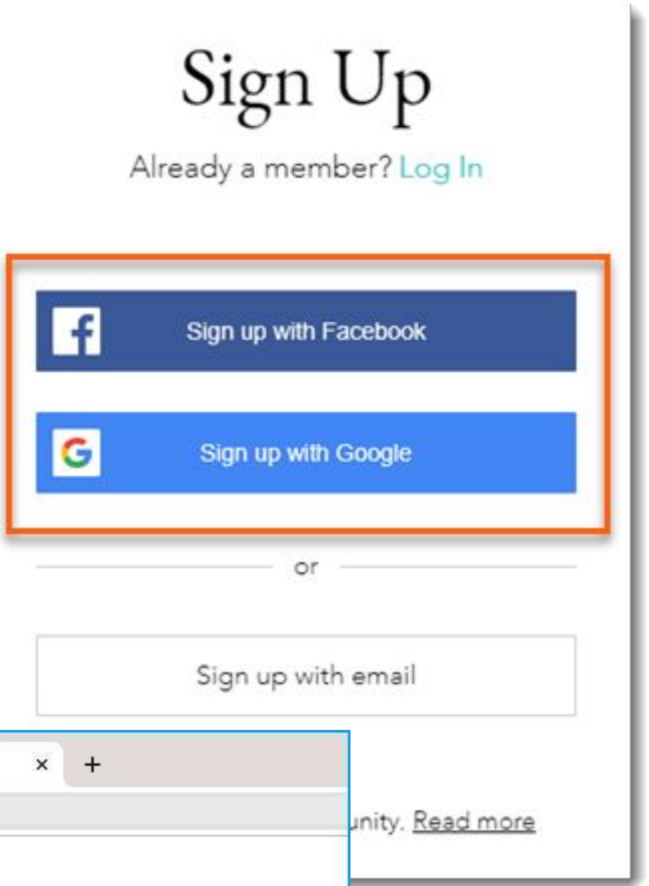
INCLUSIVE DIGITAL IDENTITY, AN IDENTITY FOR ALL

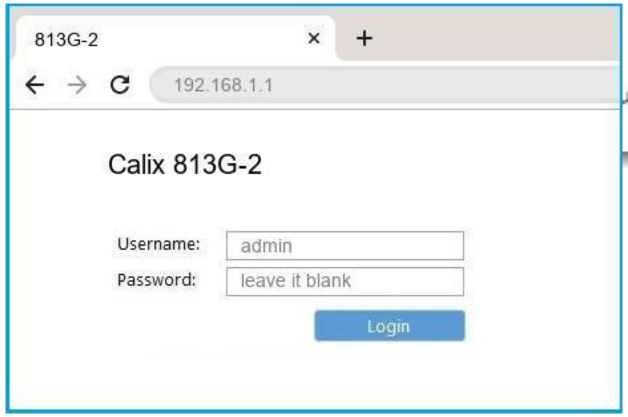
SICPA

Agenda

Introduction

When I think about Digital Identity today….



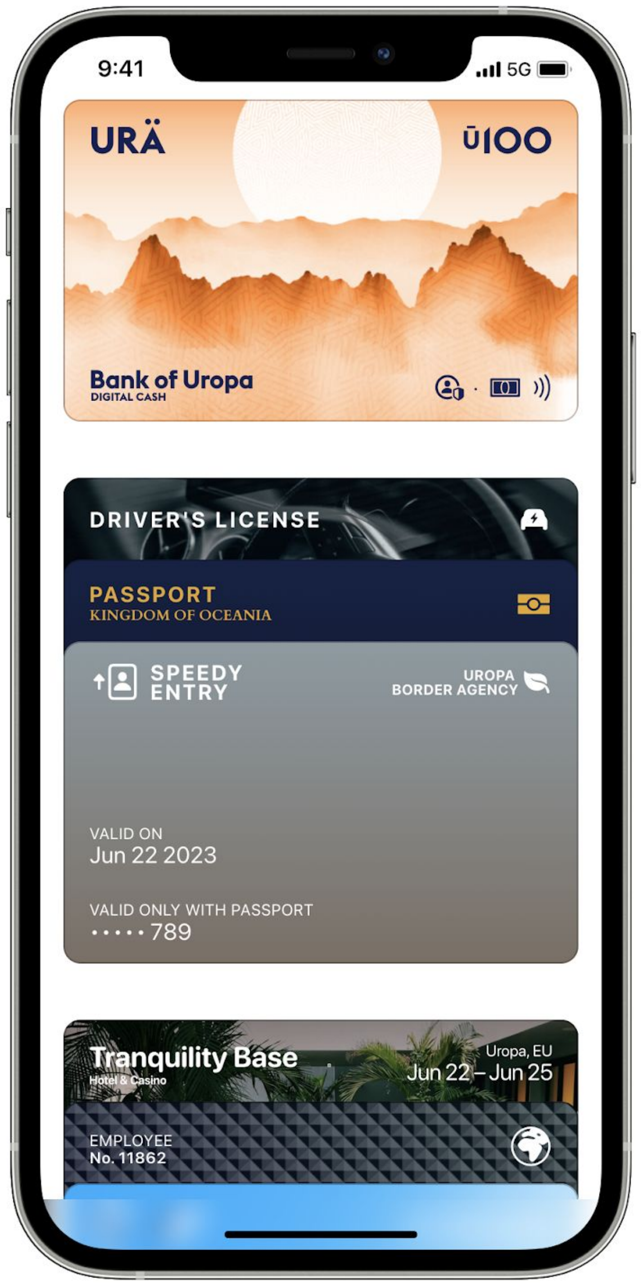


Calix 813G-2

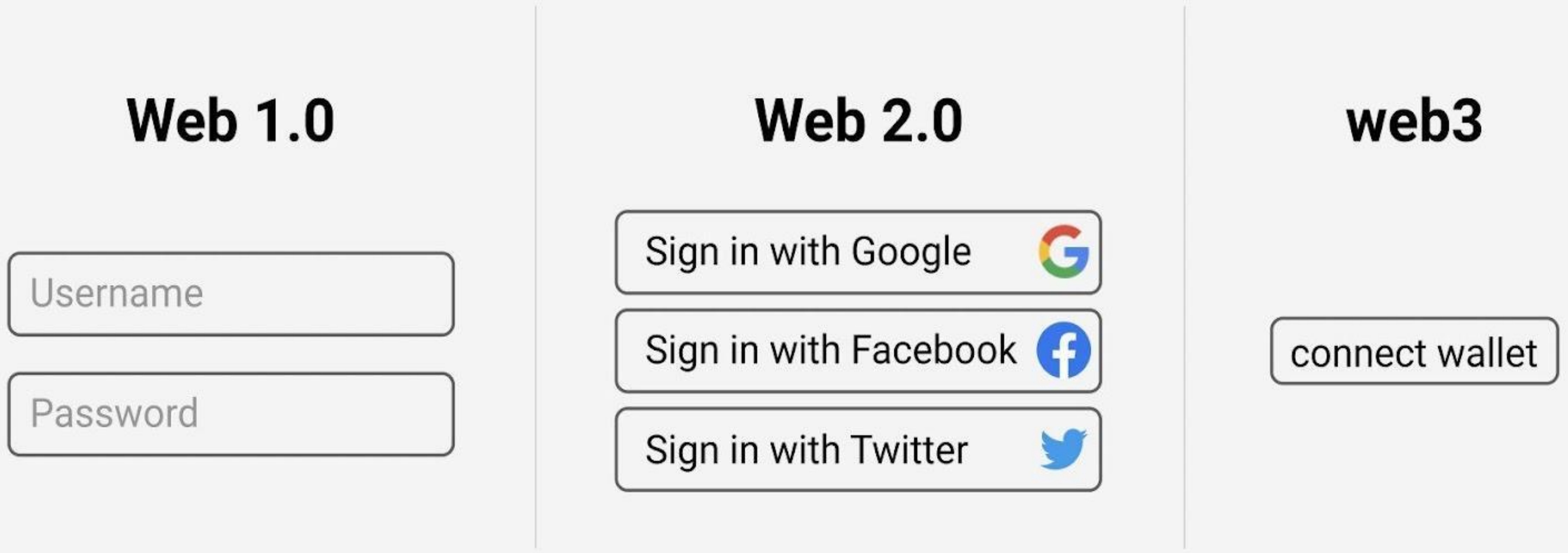
Username: admin

Password: leave it blank

Future is Digital Wallets…



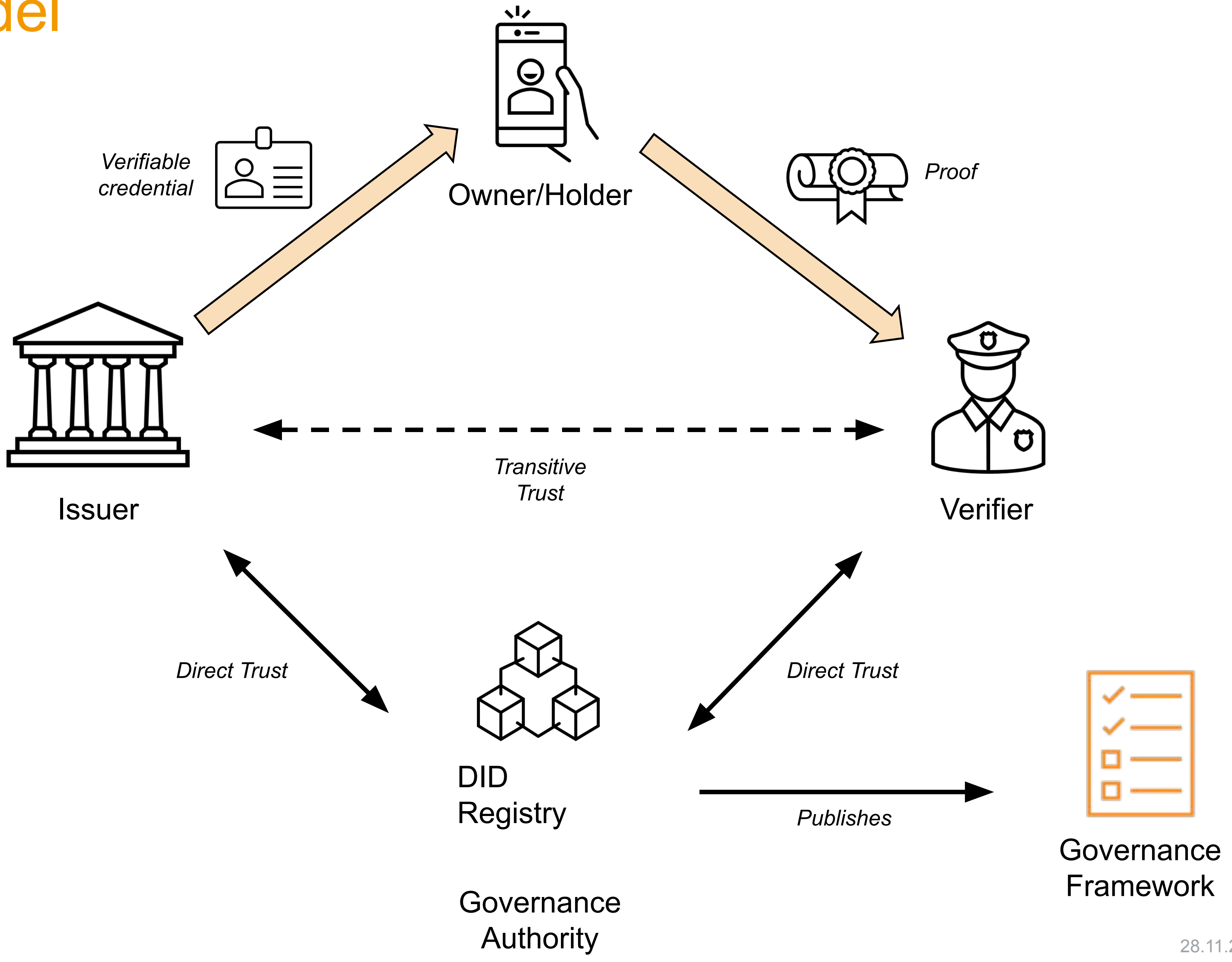
Digital wallets



What is SSI – Self Sovereign Identity?

Self-sovereign identity (SSI) is an approach to digital identity that gives individuals control over the information they use to prove who they are to websites, services, and applications across the web

SSI Trust model



DIDs and VCs

Decentralized identifiers (DIDs) are a new type of identifier that enables verifiable, decentralized digital identity. A DID refers to any subject (e.g., a person, organization, thing, data model, abstract entity, etc.) as determined by the controller of the DID. In contrast to typical, federated identifiers, DIDs have been designed so that they may be decoupled from centralized registries, identity providers, and certificate authorities.

Verifiable credentials (VCs) are an open standard for digital credentials. They can represent information found in physical credentials, such as a passport or license, as well as new things that have no physical equivalent, such as ownership of a bank account. VCs are digitally signed, which makes them tamper-resistant and instantaneously verifiable

HYPERLEDGER ARIES

What is Hyperledger Aries ?

Hyperledger Aries provides **digital agents focused on creating, transmitting and storing verifiable digital credentials**. It is infrastructure for blockchain-rooted, peer-to-peer interactions

What is an Aries Agent ?

- Acts on behalf of an Identity owner
- Stores and utilize cryptographic keys to securely perform operations
- Communicate with other agents via DIDComm protocols

HYPERLEDGER
ARIES

Frameworks

- Aries Cloud Agent Python (aca-py)
- Aries Framework Javascrit (AFJ)
- Aries Framework Go
- Aries Framework Rust
- Aries Framework .Net

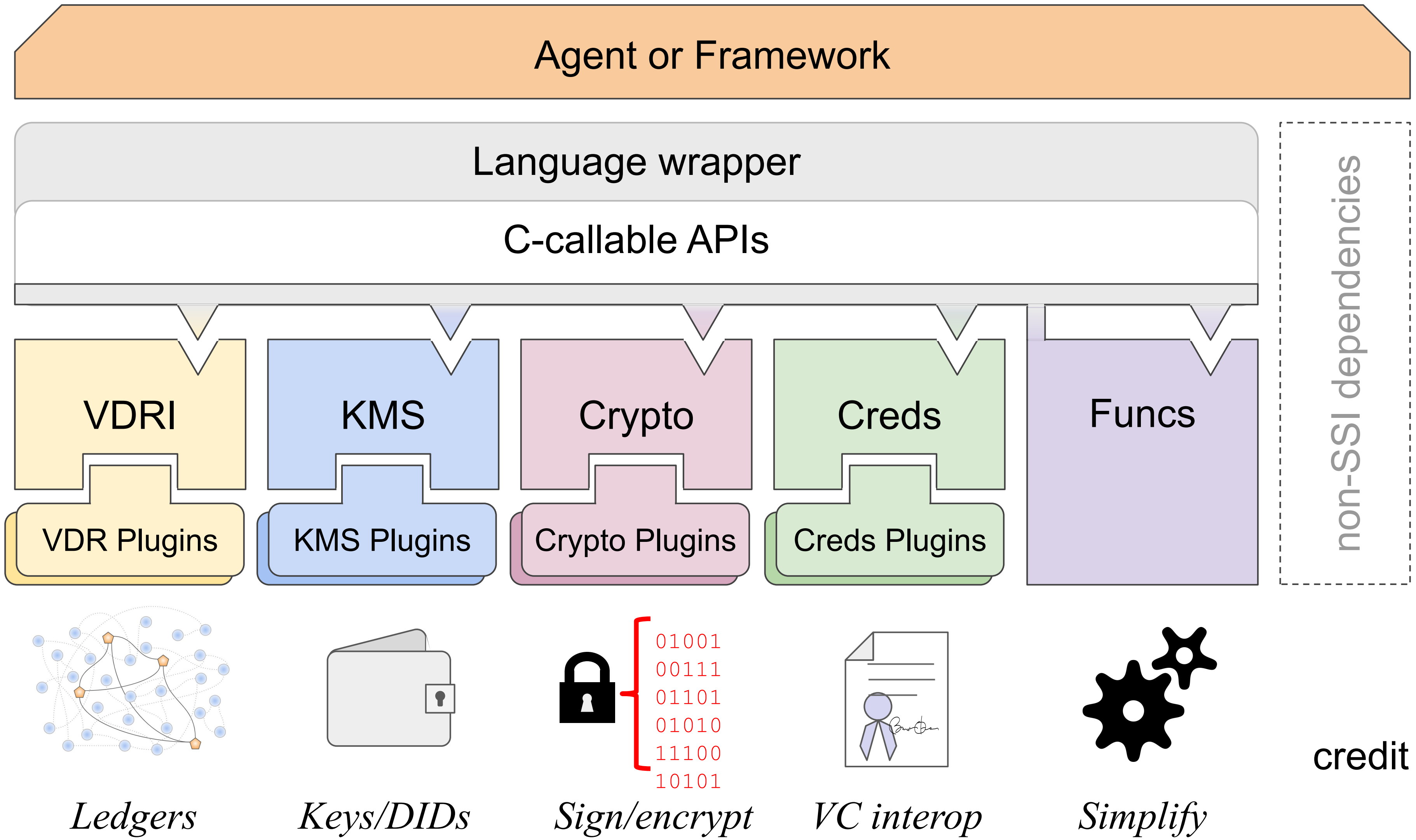
Aries Agent Test Harness

Other services

- Aries Mediators
- Aries Askar
- Aries plugins
- Annoncreds

Architecture

HYPERLEDGER ARIES



credit : Daniel Hardma

General Information

SICPA

General Information

- Repository: https://github.com/sicpa-dlab/afj-react-native-training
  - branch main is incomplete and during this workshop you will make it complete
  - branch working-wallet is complete. This can be used if you are stuck
- The repository contains a README (in wallet/README.md) which explains all the steps in detail and includes a code snippet of the implementation
- The slides can be found in the repository as well
- Documentation: https://aries.js.org

Technological Information

Technological Information

- Aries Framework
  - core (0.2.5)
  - react-native (0.2.5)
  - react-hooks (0.3.1)
- Indy Sdk
  - indy-sdk-react-native (0.2.2)
- Expo
  - @animo-id/indy-sdk-expo-plugin (0.1.0)

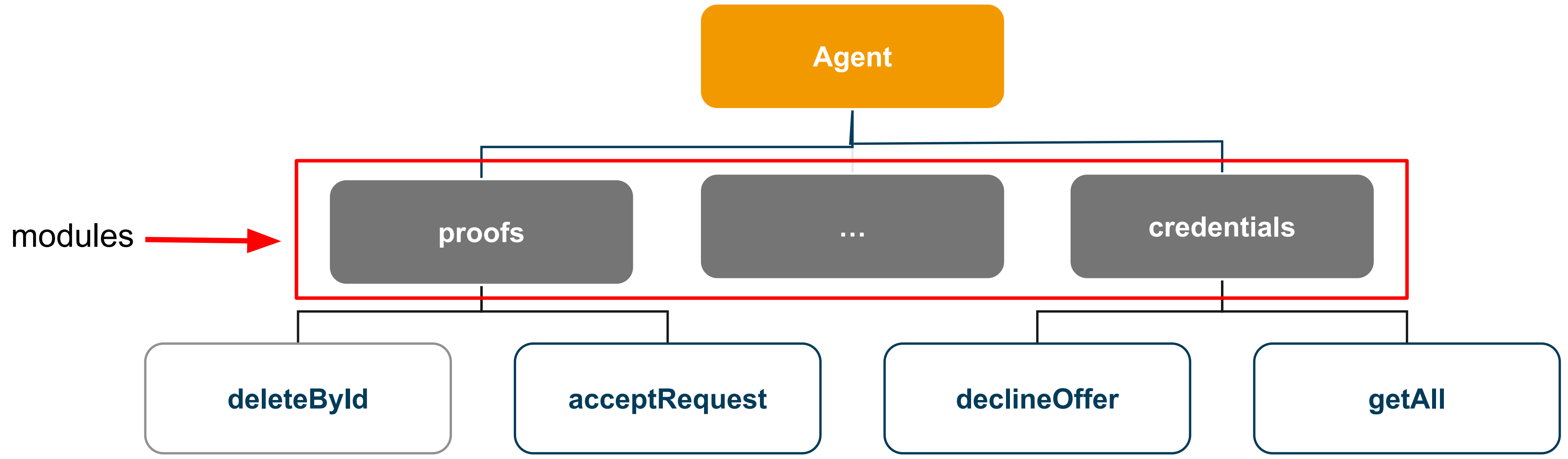
State Management

- React hooks
  - Easier to understand
  - Integrates well with hook state management
- Redux
  - More complex
  - Double state
  - Integrates better when redux is already used

Demo

Agent Structure

Agent Structure



Setting Up

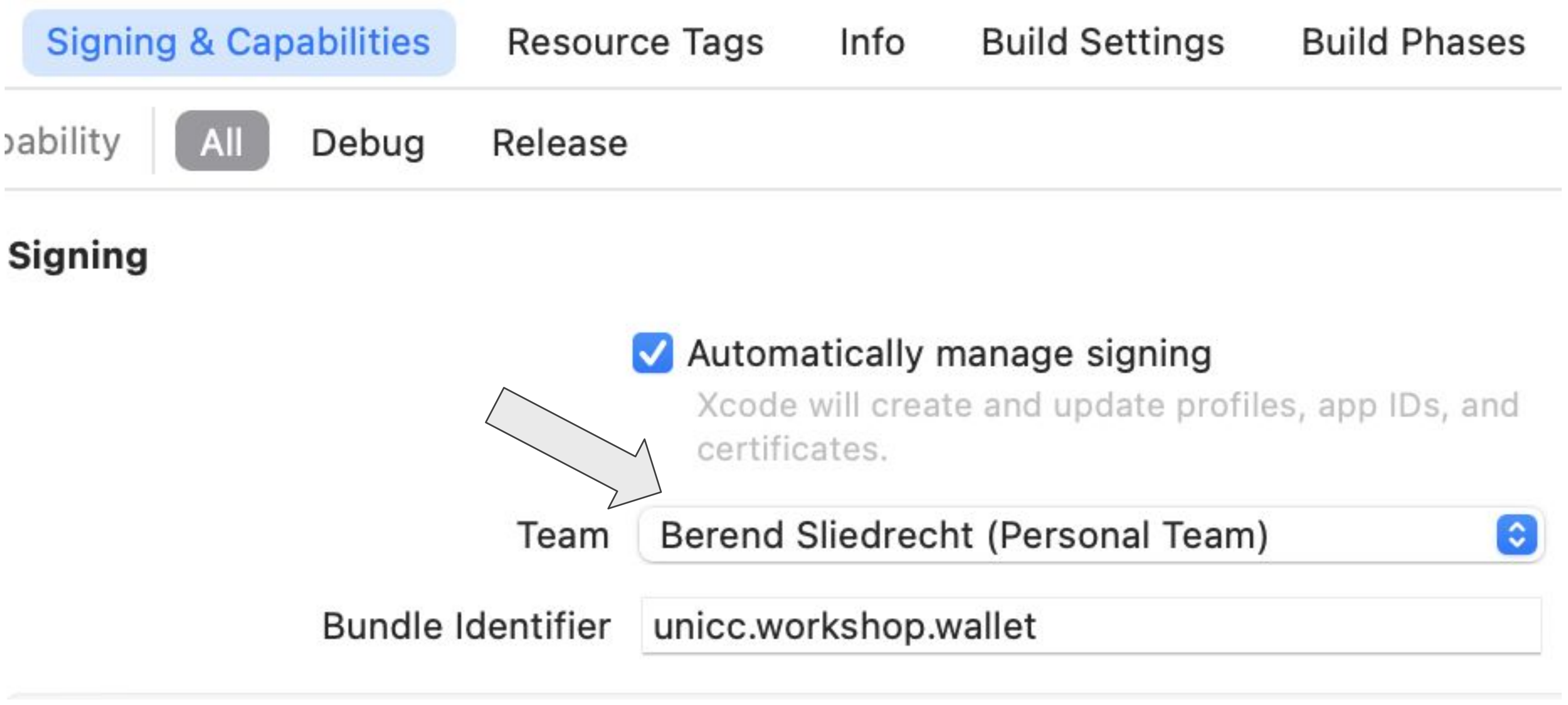
SICPA

Setting Up

- Prerequisites
  - React Native development environment (Android Studio / Xcode)
  - yarn v1
  - node v16
- clone the repository: https://github.com/sicpa-dlab/afj-react-native-training
- Go to the wallet directory
- run `yarn install` to install the required dependencies
- run `yarn prebuild` to prebuild the iOS and Android directories
  - The application does not work for iOS simulators
- run `yarn android` to start it on an Android device
  - Since we have to scan QR codes, a physical device might be better
- run `open ios/UniccWalletWorkshop.xcworkspace/` to launch Xcode for iOS
  - Add a development team (can be personal)
  - Select device as target
  - Click ‘play’
- run `yarn start` to start the expo dev server

Starting The Application - iOS

1. Select your team
2. Change the bundle Identifier (if required)



Signing

☑ Automatically manage signing

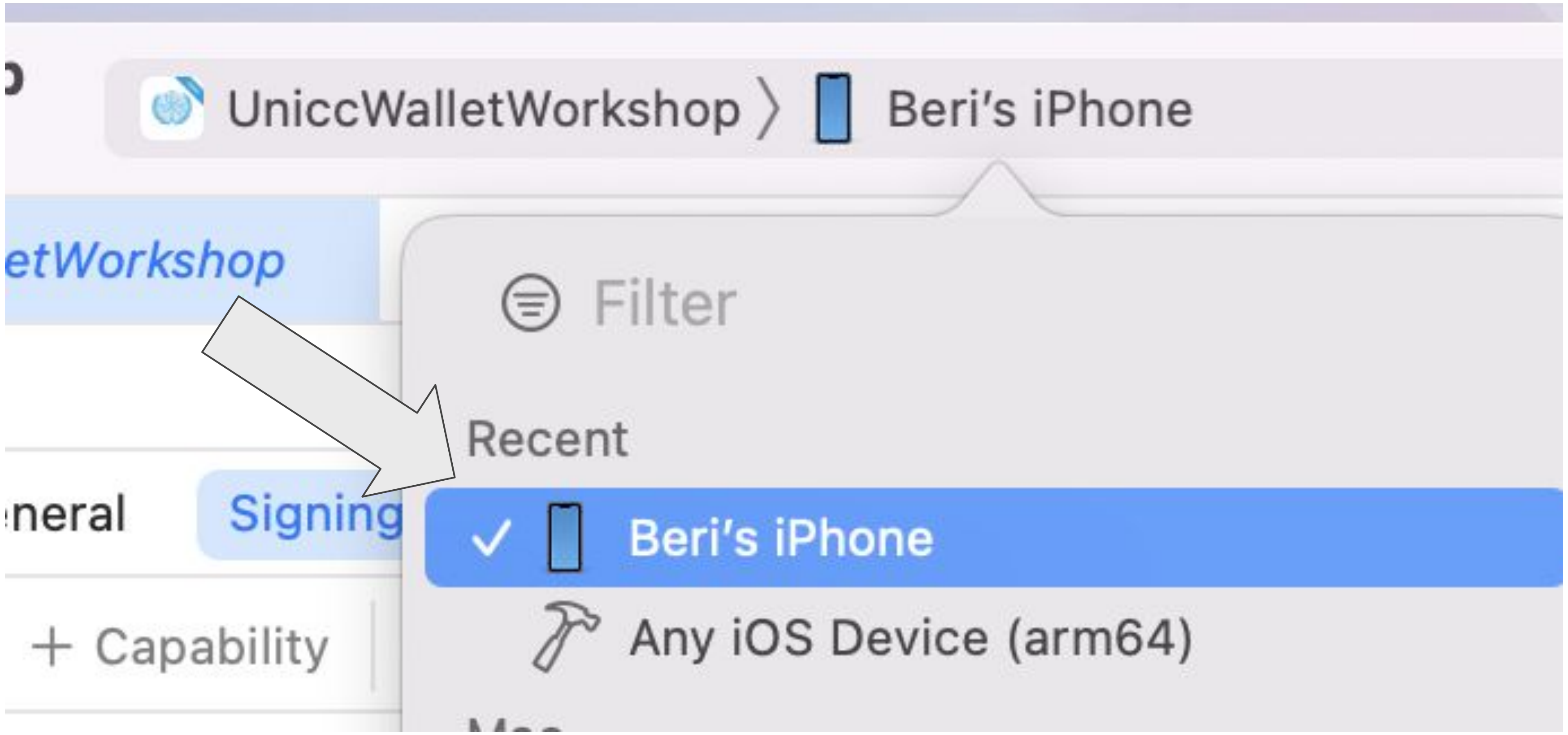
Xcode will create and update profiles, app IDs, and certificates.

Team: Berend Sliedrecht (Personal Team)

Bundle Identifier: unicc.workshop.wallet

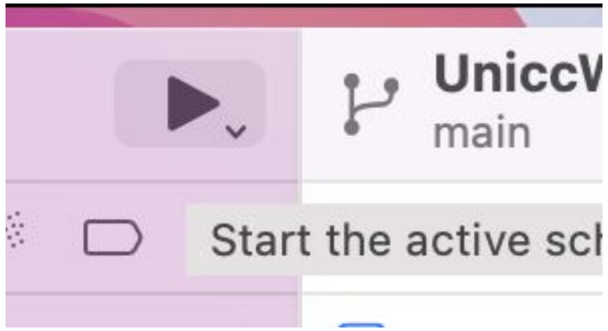
Starting The Application - iOS

1. Select your device



Starting The Application - iOS

1. Click the play button to launch the wallet



Agent Setup

Agent Setup - What can we do here?

- Label
- Wallet configuration
  - Id
  - Key
- Auto acceptance
  - Connections
  - Credentials
  - Proofs
- Ledgers
  - Production vs test ledgers
  - Genesis transactions
- Mediator
  - Invitation
  - Pickup protocol
- Update wallet
- Register transports
  - Inbound Transport
  - Outbound Transport
- Wallet initialization
- Dependency injection
  - Push notifications

Agent Setup - Minimal Working Agent

- Label
- Wallet configuration
- Transport
- Wallet initialization

Creating And Initializing The Agent

```
const logger = __DEV__ ? new ConsoleLogger(LogLevel.debug) : undefined

export const agentConfig: InitConfig = {
 label: 'My Mobile Application',
 logger,
 indyLedgers: [{
   id: "internal-id-test-net",
   isProduction: false,
   genesisTransactions: `{"reqSignature: {}, ..."}`
 }],
 autoAcceptConnections: true,
 autoAcceptCredentials: AutoAcceptCredential.ContentApproved,
 autoAcceptProofs: AutoAcceptProof.ContentApproved,
 connectToIndyLedgersOnStartup: true,
 autoUpdateStorageOnStartup: true,
 mediatorConnectionInvite: "https://mediat...X1dfQ"
 mediatorPickupStrategy: MediatorPickupStrategy.PickUpV2,
}

export async function createAgent(config: InitConfig) {
 const agent = new Agent(config, agentDependencies)
 agent.registerOutboundTransport(new HttpOutboundTransport())
 agent.registerOutboundTransport(new WsOutboundTransport())

 agent.injectionContainer.resolve(PushNotificationsApnsModule)
 await agent.initialize()

 return agent
}
```

SICPA

Thank you
for your attention

—

SICPA

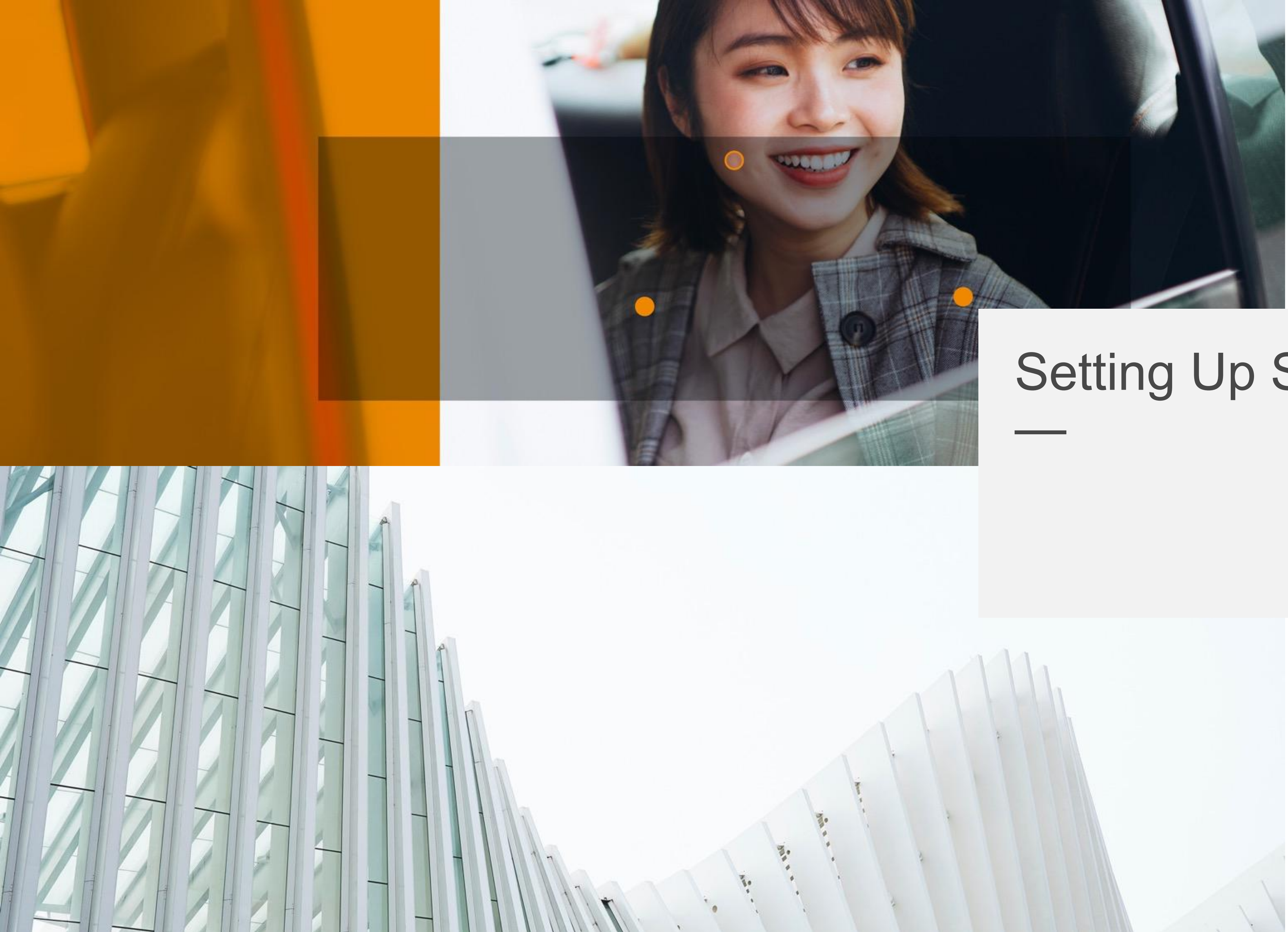
Agenda

- Recap Part I
- Setting up Siera
- Connecting with another agent
- Receiving a credential

Recap Part I

Recap Part I

- General information
- Application running
- Agent configuration



Setting Up Siera

Setting Up Siera

Windows

1. Go to https://siera.animo.id
2. Click on ‘Get Started’ and download the Windows executable

macOS (arm & x86)

1. With brew: brew install siera
2. With cargo: cargo install siera

Linux

1. With cargo: cargo install siera

Arch

1. With AUR: yay -S siera

Ubuntu

1. apt install siera

Setting Up Siera

In a terminal run the following command to setup an environment:

```
siera configuration add -d
```

To test that everything is working correctly you can create a QR invitation:

- The QR is rather big, be careful

```
siera connection invite -q
```

If something goes wrong, you can always add a `-v` for more logs

```
siera -vvv message --connection-id=fake_id --message=”Hello!”
```

To see your configuration

```
siera configuration view
```

Setting Up Siera

Here are some example of what you can do with Siera:

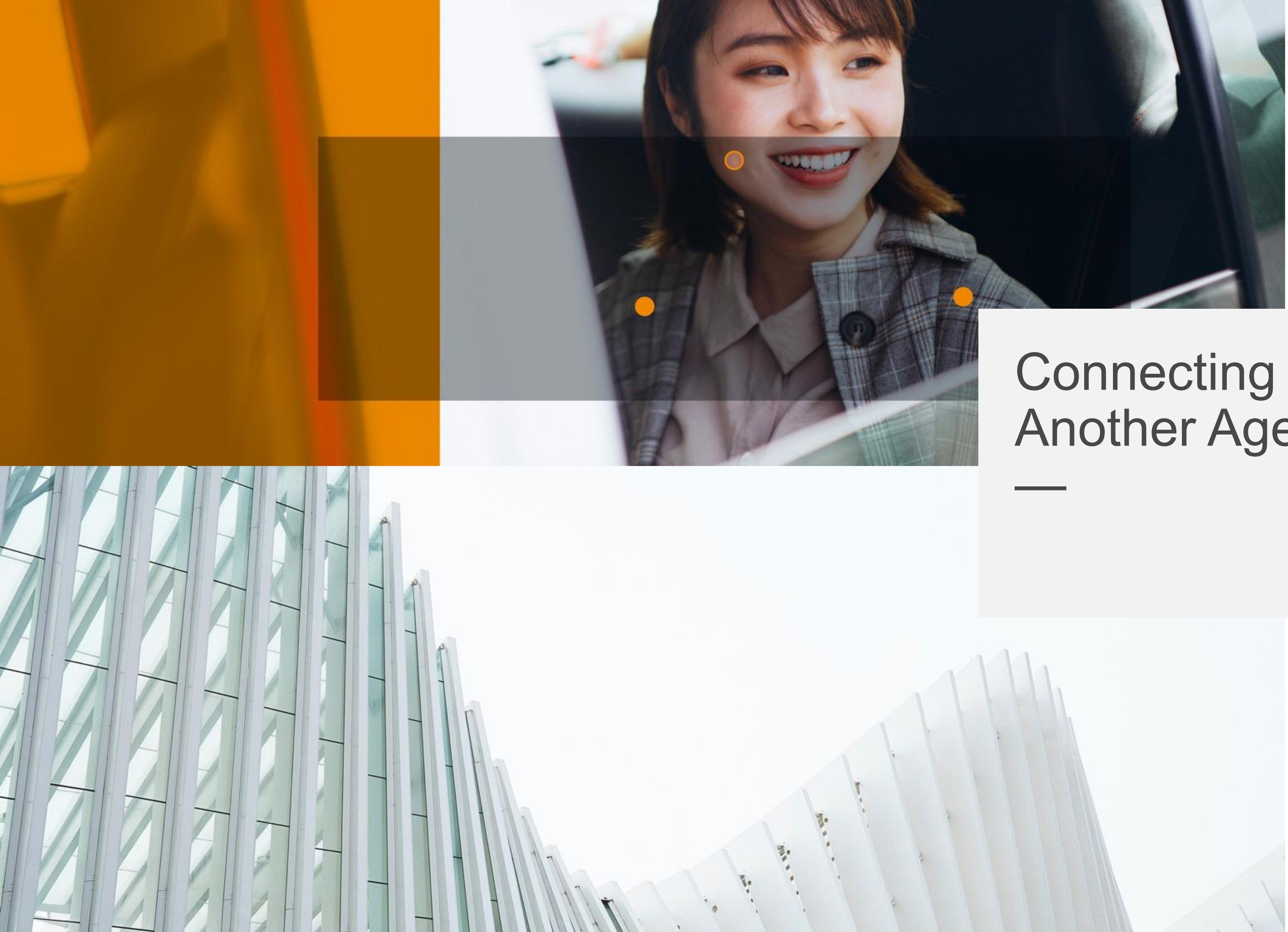
```
# List your active connections
siera connection list -s=active

# Create a schema on the ledger
siera schema create -n=”Test Schema” -a=name -a=age

# Register a credential definition
siera credential-definition create -s=example_schema_id -t=”test cred def”

# Offer a credential
siera credential offer -c=example_cred_def_id -i=connecton_id -k=name -v=john
-k=age -v=30

# Request a proof
siera proof request -c=connection_id -n=”Identity” -a=name -p=”age,>=,23”
```



Connecting With Another Agent

Connections

Why do we need a connection between agents?

- Peer to Peer reusable communication
- Add some metadata to the connection, like an alias

Connecting With Another Agent

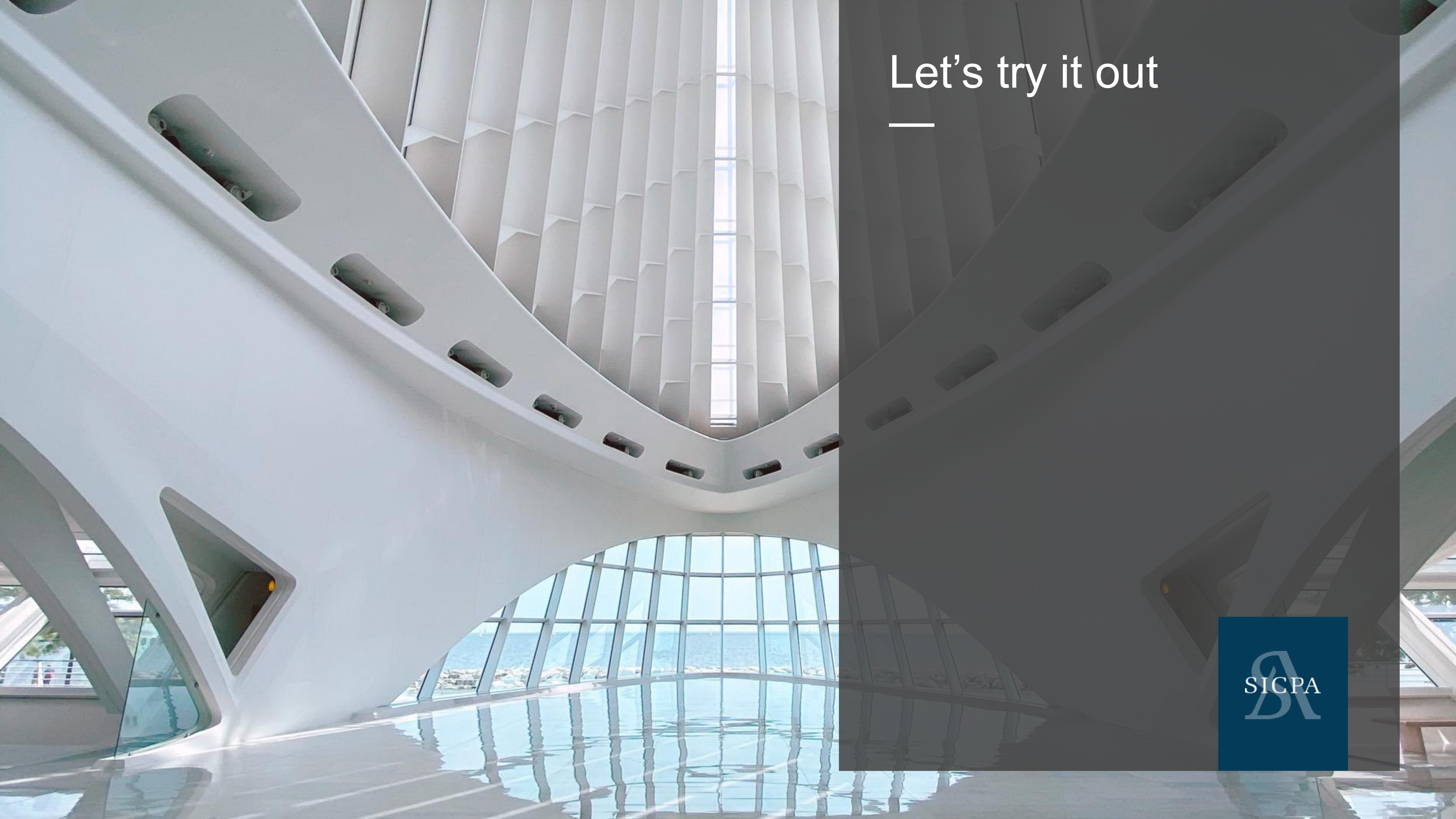
Before we can connect with another agent, we have to add some code to the configuration.

1. We must add a mediator connection invitation
2. We should add some auto acceptance

Connecting With Another Agent

Once the configuration is finished, we have to do the following:

1. Make sure that we can see our connection information
2. Parse the QR code so we can display information about the invitation to the user
3. Receive the invitation on the agent
4. Deleting old connections would also be nice



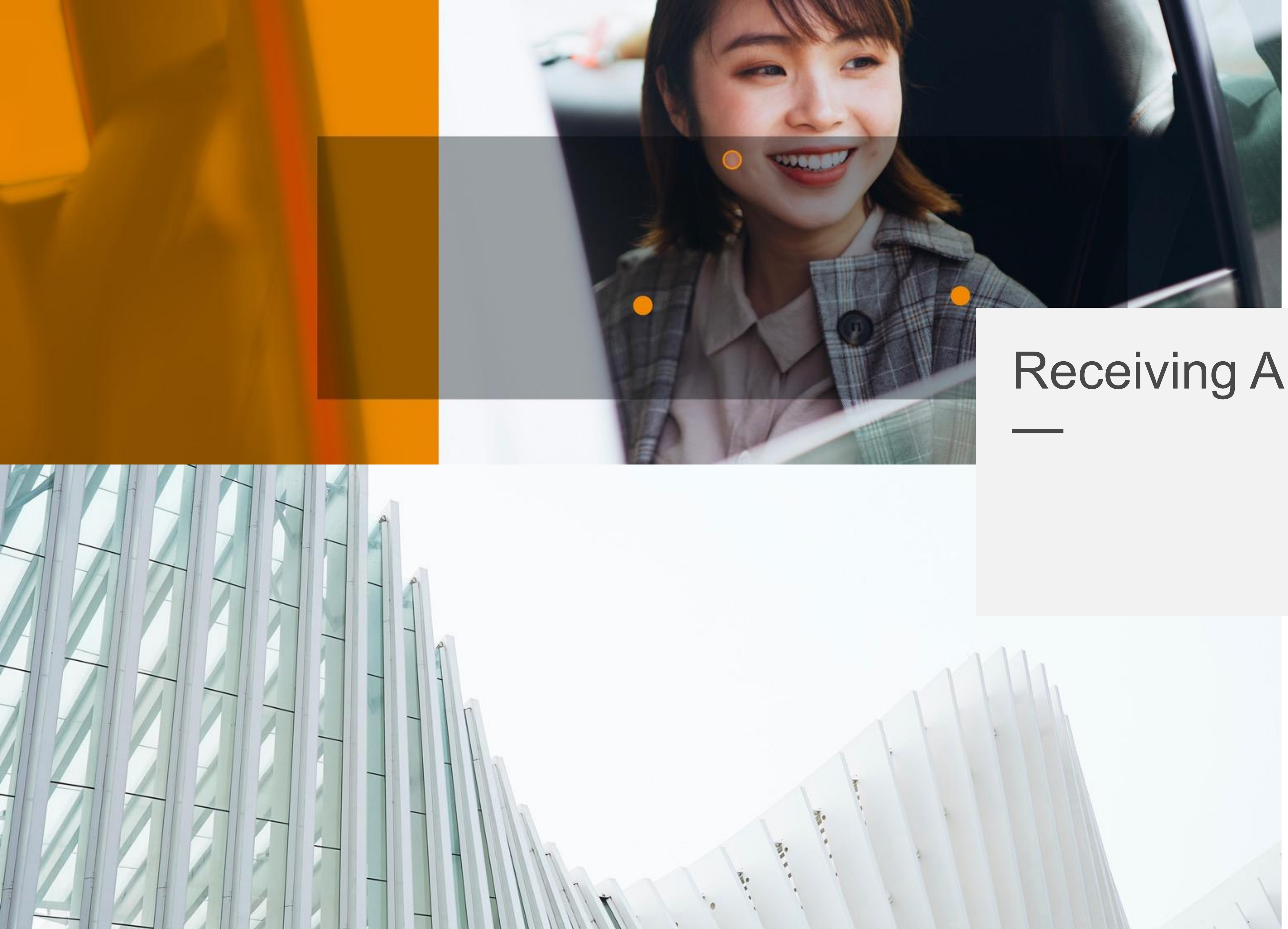
Let’s try it out

Let’s try it out

Siera will create an invitation that we will scan and accept in the wallet

```
siera connection invite --qr
```

After this, there should be a new connection in your wallet
If you tap on it, a new page should appear with connection details



Receiving A Credential

Receiving A Credential

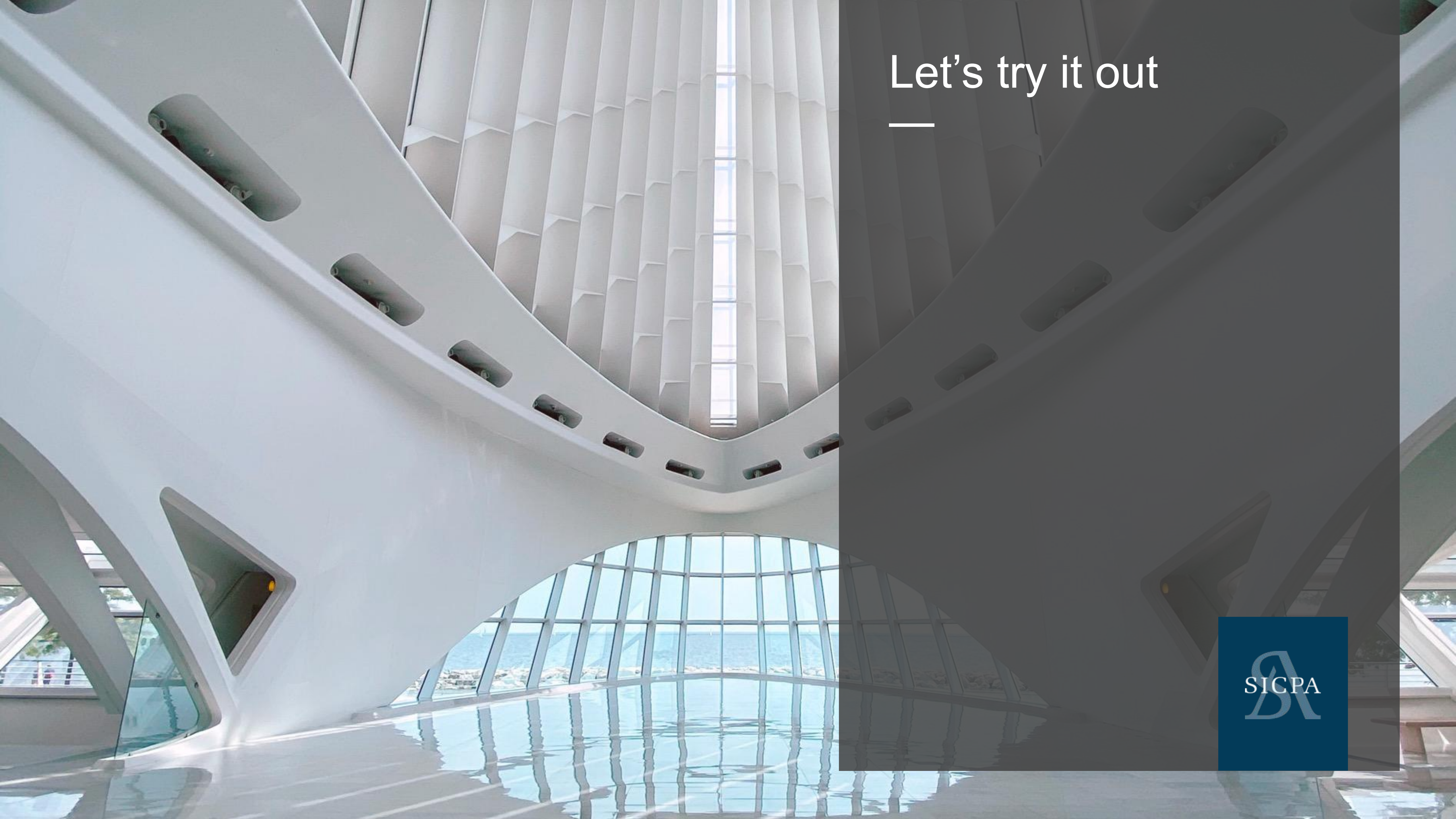
As with the connections, we have to change some configuration on the agent:

1. Add the Test Ledger from http://test.bcovrin.vonx.io/
2. Add auto acceptance for incoming credentials

Receiving A Credential

Before we can receive a credential we have to add some functionality to the wallet

1. Display the credential information
2. Declining a credential
3. Accepting a credential
4. Deleting a credential



Let’s try it out

Let’s try it out

Use the following siera commands

```
# This command returns the connection id
siera connection invite --qr

# This command returns the schema id
siera schema create --name=”My Unique Name” -a=name -a=age

# This command returns the credential definition id
siera credential-definition create --schema-id=SCHEMA_ID

# Let’s offer a credential
siera credential offer -c=CRED_DEF_ID -i=CONNECTION_ID -k=name -v=john
-k=age -v=23
```

SICPA

Thank you
for your attention

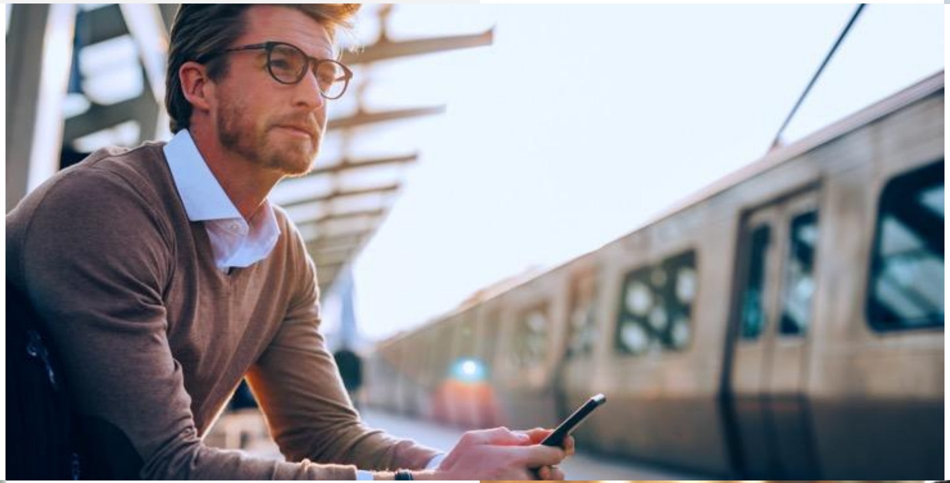
—

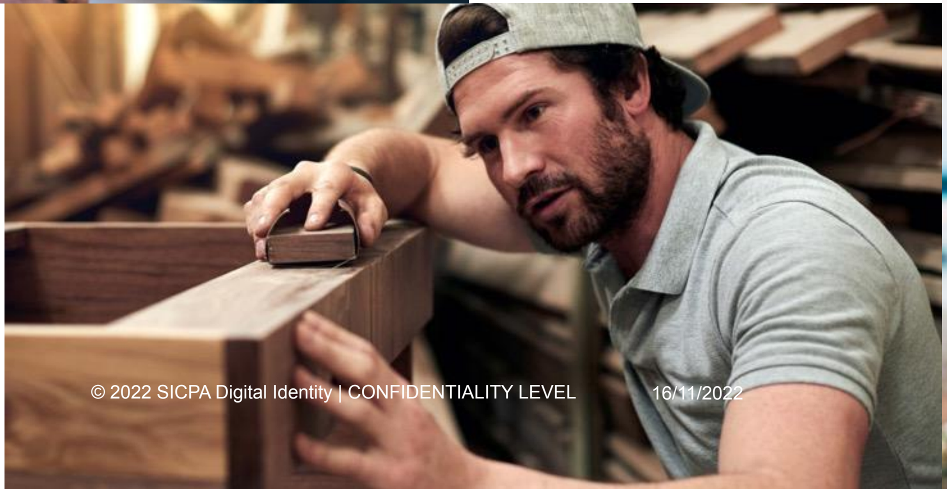
SICPA

Agenda

- Recap Part II
- Sharing a proof
- Walking through the demo
- Finishing up

Recap Part II

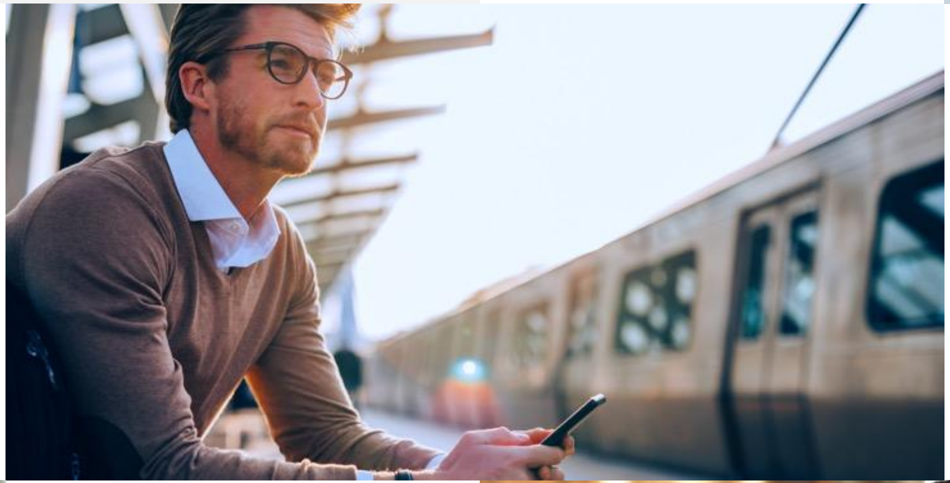




Recap Part II

- Setup siera
- Established a connection
- Received a credential

Sharing A Proof



Sharing A Proof

As we have done before, we first must add something to the agent configuration:

1. Auto acceptance

Sharing A Proof

- Responding to a proof request is quite different from responding to a credential offer.
- Before we even respond to a proof request, we must first get the credentials from our wallet

1. Displaying the proof request information
2. Selecting the correct credentials
3. Accepting a proof
4. Deleting a proof (on two different locations)

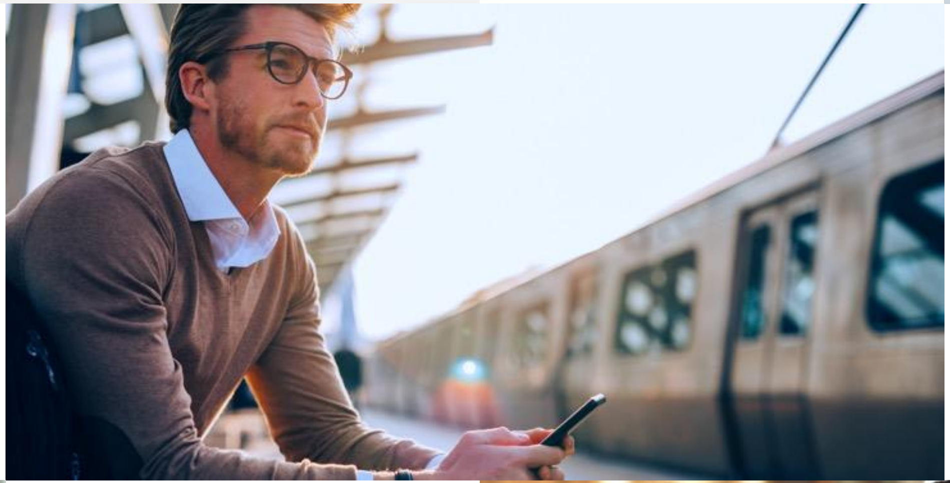
Sharing A Proof

Now we can test this with siera

1. We need a credential in our wallet
2. We can create a proof request (with the attribute of name and predicate of age > 20)
3. We should be able to share this proof

```
siera proof request --connection-id=CONNECTION_ID --name=”Example Proof
Request” -a=name -p=”age,>=,20”
```

Walking through the demo

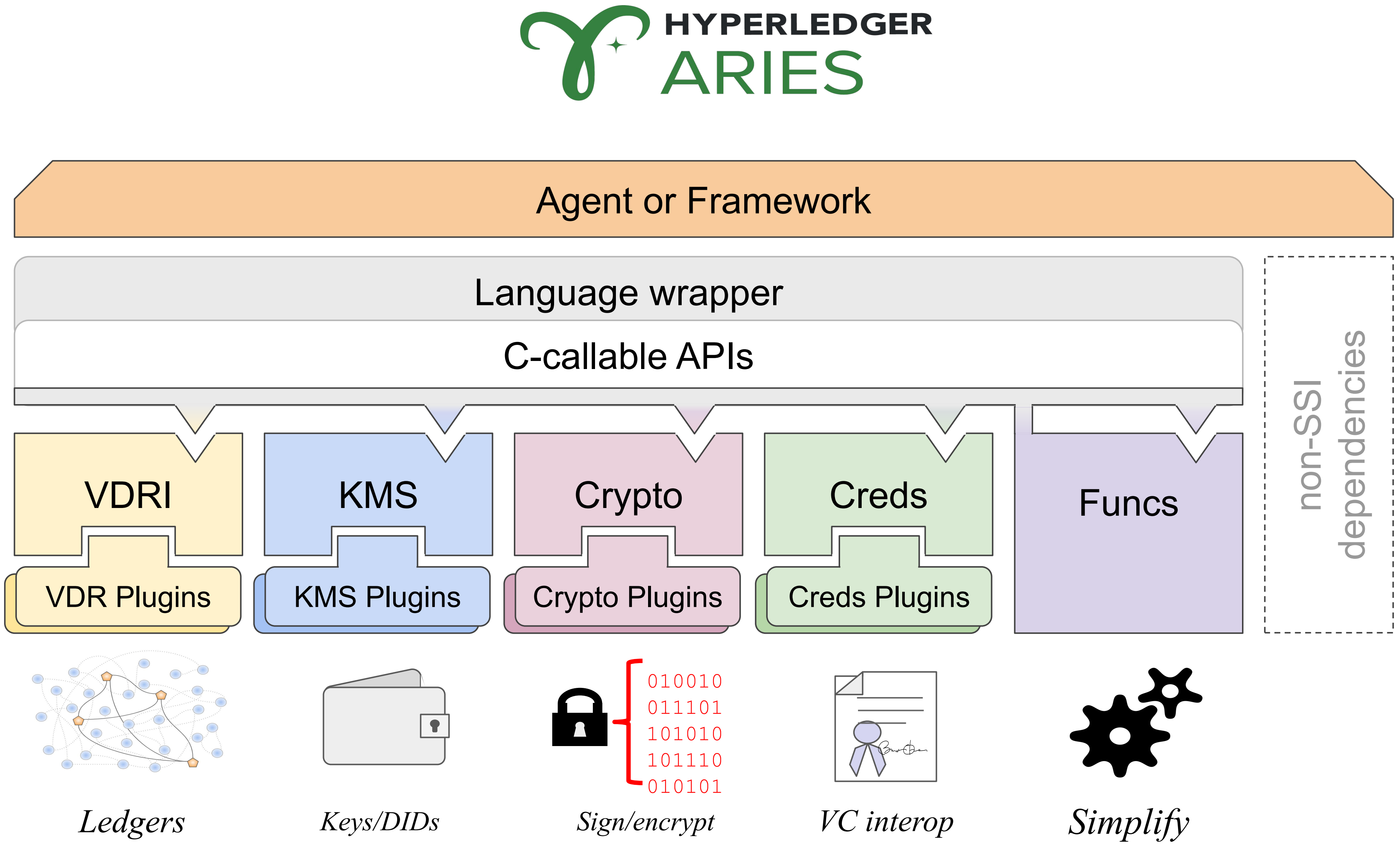


Walking through the demo

The demo is located at: https://demo.animo.id

You should be able to fully complete the use case of ‘Noah’

How Aries is evolving ?



Anoncreds

HYPERLEDGER ANONCREDS

- https://github.com/hyperledger?q=anoncreds

- Anoncreds ledger agnostic initiative
    - https://github.com/orgs/hyperledger/projects/16
    -
- Anoncreds W3C mapping
  https://github.com/andrewwhitehead/anoncreds-w3c-mapping

- Anoncreds 2.0

DIDCOMM v2

- DIDComm v2 is getting adopted in Aries

- DIDComm v2 protocols in DIF
    - https://identity.foundation/didcomm-messaging/spec/

- WACI DIDComm v2
    - https://identity.foundation/waci-didcomm/

- SICPA will donated DiDcomm v2 libraries to Open Wallet Fundation

Aries Architecture is becoming more modular

- Secure storage (Askar or indy wallet)
    - https://github.com/hyperledger/aries-askar
- Indy VDR / Anoncreds VDR
    - https://github.com/hyperledger/indy-vdr
    - https://github.com/hyperledger/anoncreds-methods-registry
- Better support of W3c credential data model
- Crypto support of BBS+, ED25519 and others
    - https://github.com/hyperledger/aries-askar/tree/main/askar-crypto

Miscellaneous

- Open wallet fundation
  - https://github.com/openwallet-foundation
- EUDI Wallet (EU Digitial Identity Wallet)
  - European Digital Identity Architecture and Reference Framework Available
    - MDL standard (attended / unattended)
    - DTC standard
    - w3c credential

- OID for verifiable credentials
  - https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html
  - https://openid.net/specs/openid-4-verifiable-presentations-1_0.html
  - https://openid.net/specs/openid-connect-self-issued-v2-1_0.html

Useful resources

- **Discord**
  - https://discord.com/invite/hyperledger
- **Repositories**
  - https://github.com/hyperledger/aries-framework-javascript
  - https://github.com/hyperledger/aries-framework-javascript-ext
  - https://github.com/hyperledger/aries-javascript-docs
- **Siera CLI tool**
  - Website: https://siera.animo.id/
  - Repository: https://github.com/animo/siera
- **Demo**
  - Website: https://demo.animo.id
  - Repository: https://github.com/animo/animo-demo
- **Various SSI resources:**
  - https://github.com/animo/awesome-self-sovereign-identity
- **Email**:
  - Berend Sliedrecht: berend@animo.id
  - Karim Stekelenburg: karim@animo.id

Useful resources didcomm v2

- **AFJ PR**
  - https://github.com/hyperledger/aries-framework-javascript/pull/1096
- **SICPA DIDComm v2 libraries**
  - https://github.com/sicpa-dlab/peer-did-python
  - https://github.com/sicpa-dlab/didcomm-rust
  - https://github.com/sicpa-dlab/didcomm-python
  - https://github.com/sicpa-dlab/didcomm-jvm
  - https://github.com/sicpa-dlab/didcomm-react-native
  - https://github.com/sicpa-dlab/didcomm-v2-mediator-ts
- **ble-react-native**
  - https://github.com/sicpa-dlab/ble-react-native

Useful resources aca-py

- **Kafka queues**
  - https://github.com/sicpa-dlab/aries-acapy-plugin-kafka-events
- **Redis queues**
  - https://github.com/bcgov/aries-acapy-plugin-redis-events

- **Universal resolver**
  - https://github.com/sicpa-dlab/acapy-resolver-universal

- **Universal registrar interfaces** (already available in aca-py)

SICPA

Thank you for your attention

—