Tools for Building Your Identity Application

Nemanja Patrnogic

evernym

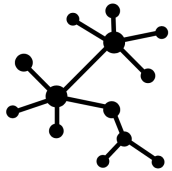
Solid foundation…

Solid foundation...

- **indy-plenum**: Ledger Impl & *Consensus Protocol (RBFT)*
- **indy-node**: *Identity Specific Transaction*
- **The Sovrin Network**: *Public-Permissioned Network*

Ledger Scanner

★ https://indyscan.io



VON

Verifiable Organizations Network

VON Browser

★ https://sovrin-mainnet-browser.vonx.io

evernym

And still...

And still...

- **indy-crypto**: *Indy Crypto Library (ZKP, BLS...)*
- **indy-sdk**: *Low Level SDK*
- **libvcx**: *High Level SDK (Conns, Creds, Wallet Backup...)*
- **indy-cli**: *Admin Interface*

connect.me®

Credential Exchange Demo

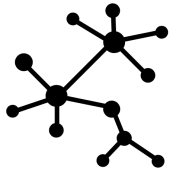
★ https://try.connect.me

evernym

And new...

And new...

- **URSA**: *Hyperledger Shared Crypto Library*
- **ACA-Py**: *Ready-to-use extensible Agent*
- **Aries Framework**: *Framework for writing Agents*
- **Aries Toolbox**: *Tools for working with Agents*
- **OSMA**: *Open-Source Mobile Agent*



VON

Verifiable Organizations Network

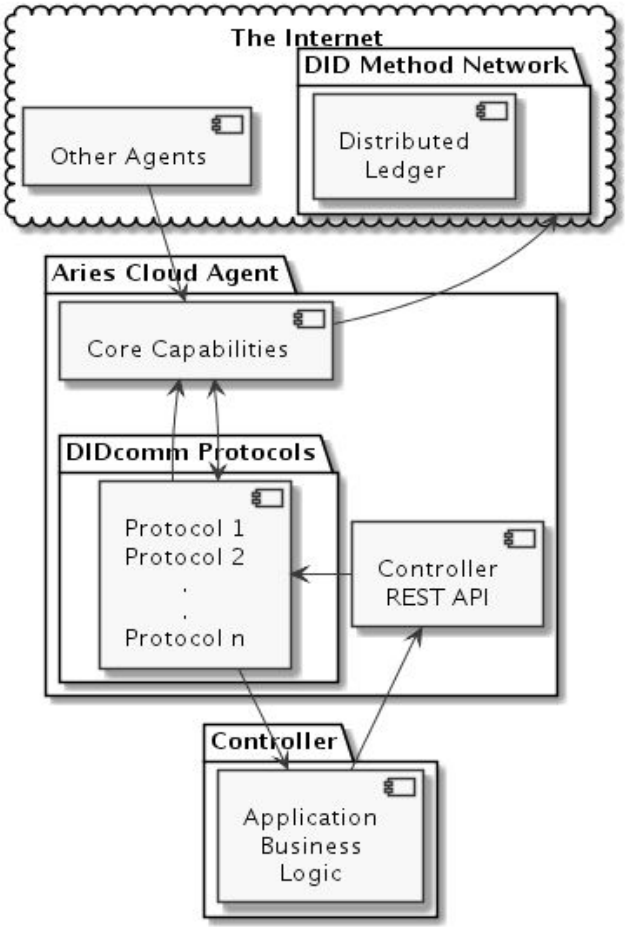
ACA-Py

★ https://github.com/hyperledger/aries-cloudagent-python

evernym

Overall Architecture

- The agent - built from PyPi and deployed “as is”
  - Configured via command line parameters
  - Interacts with other agents via pluggable transports
  - Manages storage, ledger with pluggable implementations
  - Manages messages and protocol state
  - Invokes protocols (configurable set)
  - Driven by a controller
    - Sends events to controller
    - Exposes an HTTP JSON administrative API to controller
- The controller - business logic for instance of agent instance
  - Receives events from agent
  - Sends requests to agent using HTTP JSON administrative API

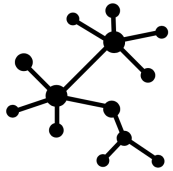


Controller: Events and Administrative API

- Listen for HTTP requests on an webhook URL
  - Process event
  - Optionally pass to external system (person, legacy system, etc.) to get “next step” decision
  - Optionally respond with an HTTP request to the Agent’s Administrative API
- Take inputs from other sources
  - Process input
  - Optionally initiate agent protocols with an HTTP request to the Agent’s Administrative API
- Simple example: Alice/Faber command line API
  - ***Simple***: Both agents know about each other
  - Deploys agent as a sub-process, initializes and waits at command line for user input
  - Process user input to initiate protocol invocations
  - Listen for, receive, process and respond to protocol events
    - Report event received and automatically process (based on command line parameters)

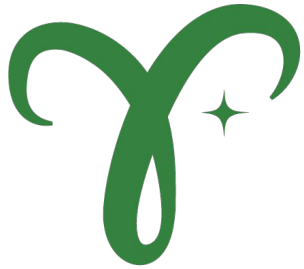
Pluggable Agent Components

- Reason
  - You want to be able to add support for other implementations for your use cases
- Pluggable implementations for:
  - Inbound, Outbound Transports: currently HTTP and WebSockets
  - Wallets - Indy entities (DIDs, schema, cred defs, etc.)
    - Basic - in memory
    - Indy-SDK - SQLite, Postgres
  - Storage - non-secrets (anything else)
    - Basic - in memory
    - Indy-SDK - SQLite, Postgre
  - Future(?): Ledger Integration



VON

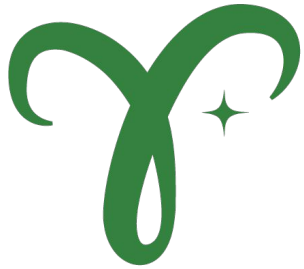
Verifiable Organizations Network



ACA.py Example

★ https://github.com/hyperledger/aries-cloudagent-python/tree/master/demo#the-alicefaber-python-demo

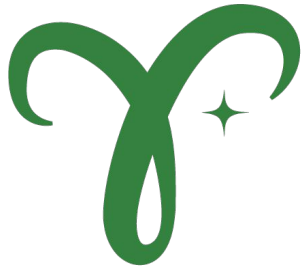
evernym



Aries Agent Toolbox

★ https://github.com/sovrin-foundation/aries-toolbox

Aries Agent Framework (.Net)



- A .NET Core library for building Sovrin interoperable agent services
- Abstraction on top of Indy SDK
- Runs .NET Standard (2.0+), including ASP.NET Core and Xamarin
- Mobile Agents (OSMA, StreedcredID)
- Web Agents
- https://github.com/hyperledger/aries-framework-dotnet

Mobile Agents...

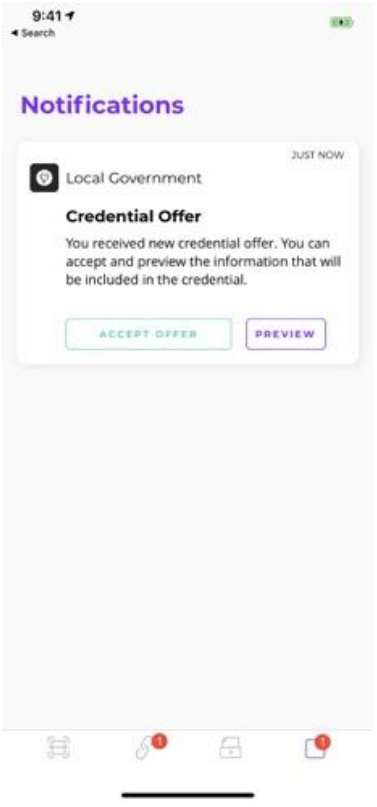
OSMA

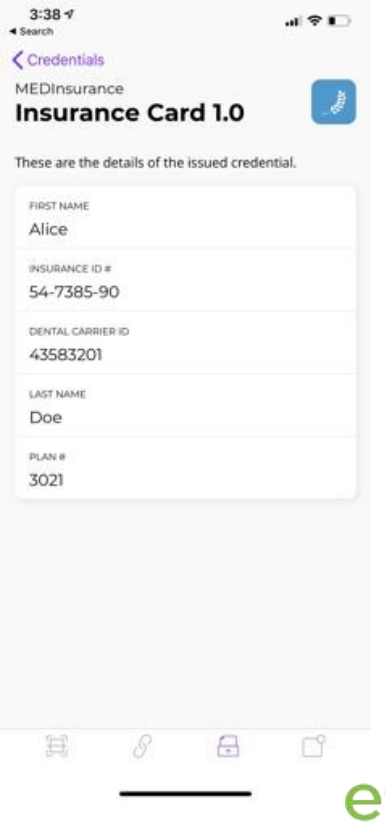
MATTR™ OpenSource

- An open source mobile agent for achieving self sovereign identity
- Provide a common project to progress emerging community standards around mobile agents.
- Cross platform mobile app (iOS/Android) built using .Net Aries Agent Framework
- Contributed by MATTR Global
- https://github.com/mattrglobal/osma

evernym

Streetcred ID





Questions?