

Making ACA-Py (Almost) Ledger Agnostic: DID resolution over DIDComm (instead of HTTP) to a Universal Resolver.

Thursday 24F

Convener: Victor Martinez Jurado, Markus Sabadello, Daniel Bluhm

Notes-taker(s): N/A

Tags for the session - technology discussed/ideas considered:

ACA-Py, Agents, DIDs, DIDComm, Universal Resolver

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

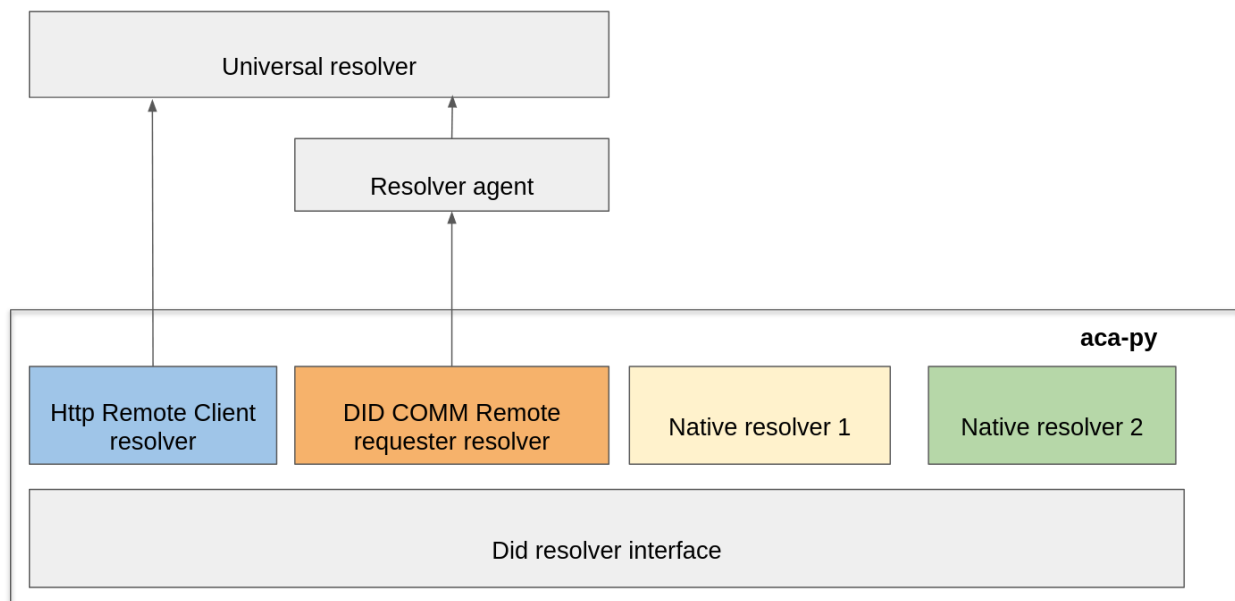
Link to slides:

https://docs.google.com/presentation/d/1oHOr5dZV5fUg3Prbx9VNDHuIM_IcDbgGablYvTN6Ps/edit?usp=sharing

The Vision: Resolve as many did methods as possible during verification

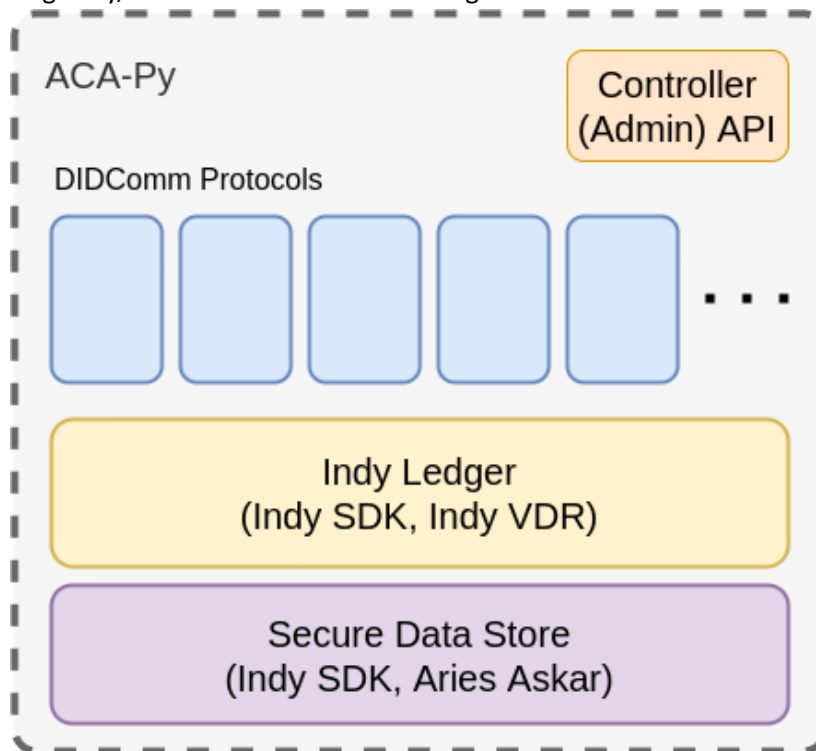
- DIDs are everywhere: the number of DID methods is constantly growing, also we don't want be locked-in to any single DID method.
- We want to leverage in ACA-Py the addition of JSON-LD credentials (plain and BBS+)

High level architecture

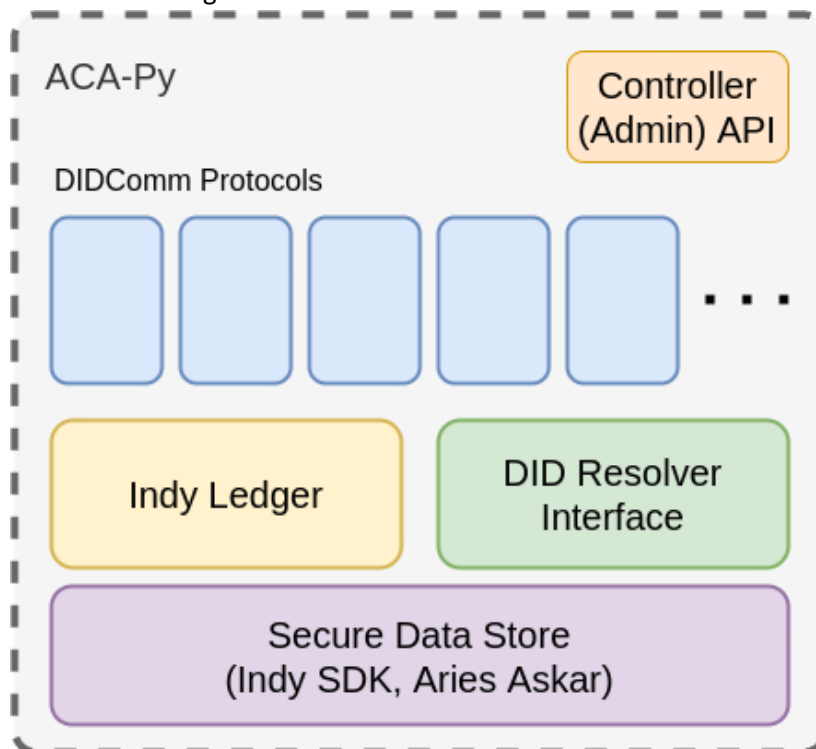


Changes made in Aries Cloud Agent - Python

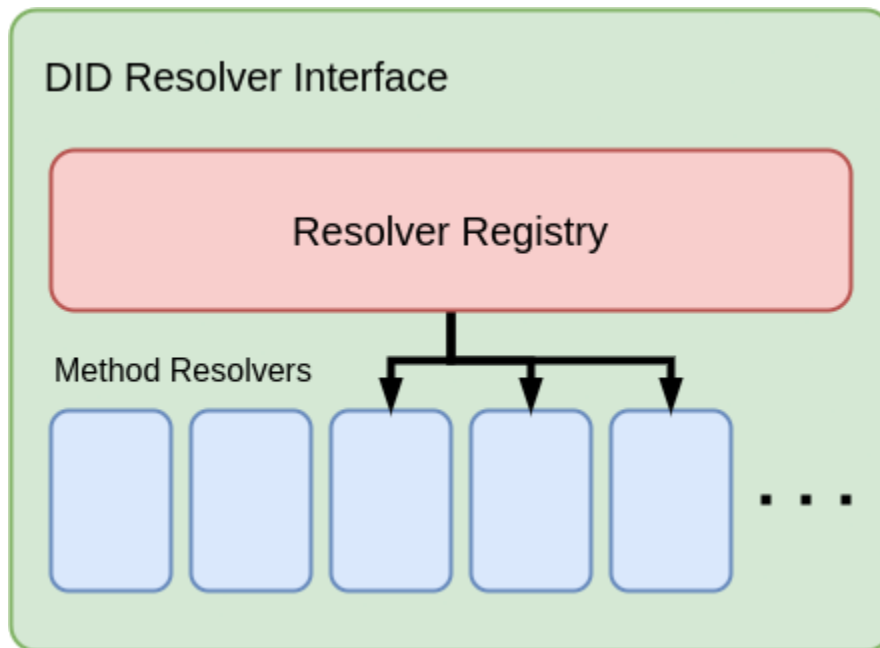
Originally, we started with the following architecture



And worked things into:



Creating an interface for pluggable DID resolvers in ACA-Py. Resolvers are matched to DIDs through use of a Resolver Registry, giving priority to natively implemented resolvers over remote resolvers.



You can run the demo yourself by following the instructions at:

<https://github.com/sicpa-dlab/aries-acapy-plugin-didcomm-resolver/tree/demo/iiw/demo>

Next Steps:

- Technical Items
 - Tighter integration with ACA-Py?
 - DID to Resolver matching via Regex (PR pending)
 - Resolution metadata included in result (resolved via native vs. non-native resolver, etc.)
 - Publish resolver plugins
 - DID Document parsing (PyDID)
- [DID Resolution Protocol](#) Improvements (reporting failures and errors)
- DID Registration?

Resources:

- <https://hackmd.io/@dblhm/uniresolver-acapy>
- <https://github.com/hyperledger/aries-rfcs/blob/master/features/0124-did-resolution-protocol/README.md>
- <https://github.com/sicpa-dlab/aries-acapy-plugin-didcomm-resolver>
- <https://github.com/sicpa-dlab/aries-acapy-plugin-http-uniresolver>