



## **NPTEL ONLINE CERTIFICATION COURSES**

**Blockchain and its applications**  
**Bishakh Chandra Ghosh**

**Department of Computer Science & Engineering**  
**Indian Institute of Technology Kharagpur**

**Lecture 51: Hyperledger Aries**

## CONCEPTS COVERED

- Aries Overview
- Aries Architecture
- Installation and usage



# KEYWORDS

- Digital Credentials
- Aries





# Hyperledger Aries

Hyperledger Aries provides a **shared, reusable, interoperable tool** kit designed for initiatives and solutions focused on creating, transmitting and storing verifiable **digital credentials**.

It is infrastructure for **blockchain-rooted**, peer-to-peer interactions.

Aries agent frameworks -.

- **Aries Cloud Agent - Python (ACA-Py)**
  - For any non-mobile application. Has production deployments.
- Aries Framework - .NET
- Aries Static Agent - Python



<https://www.hyperledger.org/use/aries>

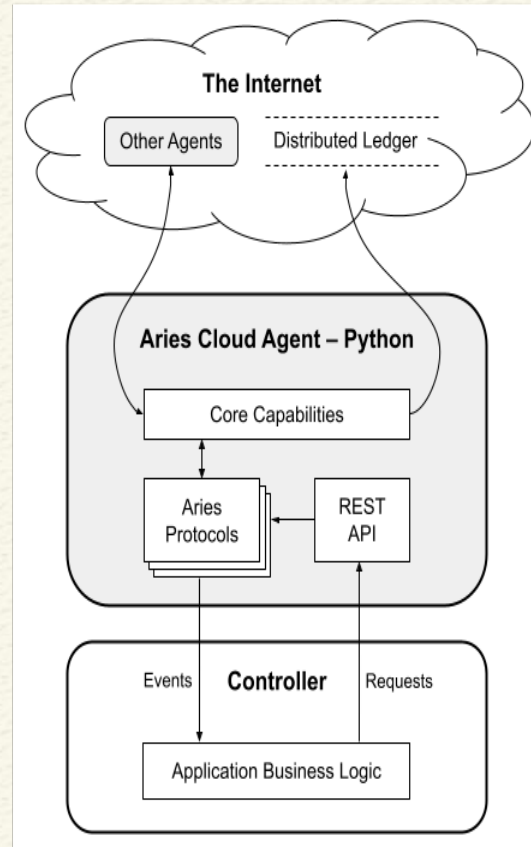


# Aries Architecture

## ACA-Py

- ACA-Py exposes a REST API to access Aries capabilities
- Write a '**Controller**' to implement application business logic.
- Controller sends **HTTP requests** (REST) to execute commands.

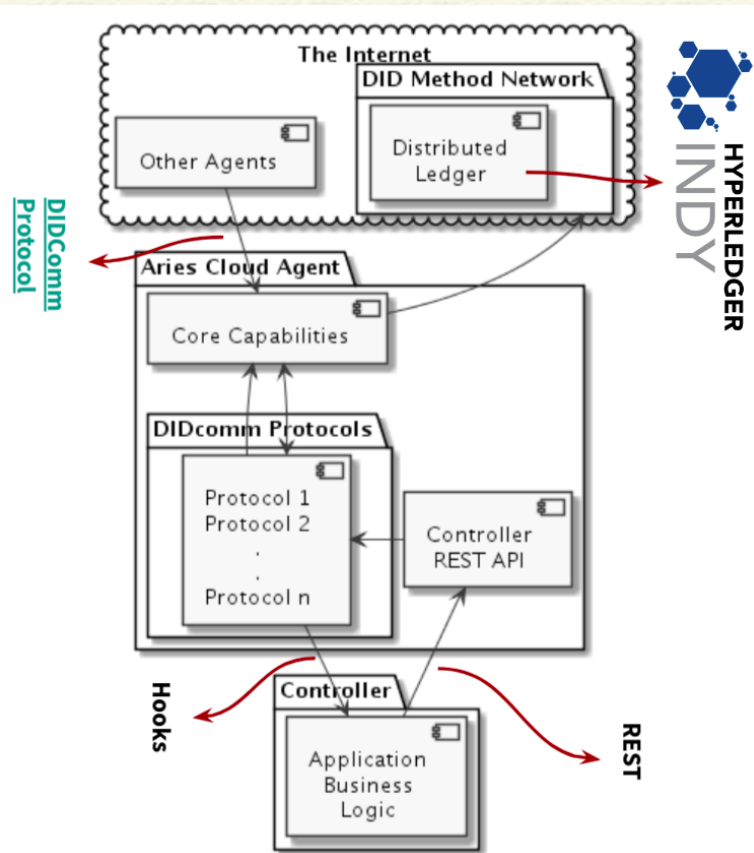
Response events are fed back to the controller as **webhooks**.



# Aries Architecture

## The agent -

- 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



# Installation

- Install libindy and indy-cli
  - `apt-key adv --keyserver keyserver.ubuntu.com --recv-keys CE7709D068DB5E88`
  - `apt-add-repository "deb https://repo.sovrin.org/sdk/deb bionic master" -y`
  - `apt-get update`
  - `apt-get install -y libindy indy-cli`
- Install python3-indy
  - `python3 -m pip install python3-indy`
- Install aries-cloudagent
  - `python3 -m pip install aries-cloudagent`





# Installation

If you installed the PyPi package, the executable `aca-py` should be available on your `PATH`.

Use the following commands to check the version, list the available modes of operation, and see all of the command line parameters:

- `aca-py --version`
- `aca-py --help`
- `aca-py provision --help`
- `aca-py start --help`

```
~$ aca-py
usage: aca-py [-h] [-v] {provision,start} ...

positional arguments:
  {provision,start}
    provision          Provision an agent
    start              Start a new agent process

optional arguments:
  -h, --help          show this help message and exit
  -v, --version        print application version and exit
~$ aca-py --version
0.5.1
```





# Starting Aries Agent

## Provisioning a Wallet

It is possible to provision an Indy wallet before running an agent:

```
aca-py provision --wallet-type indy --seed $SEED
```

**Use the SEED to configure existing nodes in the indy pool, e.g. Stewards.**



# Starting Aries Agent

When starting an agent instance, at least one ***inbound*** and one ***outbound transport*** MUST be specified.

For example:

```
aca-py start --inbound-transport http 0.0.0.0 8000 --outbound-transport http
```

## More Parameters:

- **Transports:** inbound, outbound, endpoint
- **Logging/debugging settings**
- **Label:** self-attested agent name
- **Wallet** implementation and related info
- For controller: **Admin API** configuration
  - URL
  - Security selection
- Protocol **automation flags**
- **Ledger parameters** (e.g. genesis URL, etc.)
- Add timing information to messaging
- Optional protocols to load
- From controller: **Event webhook URL**




# Using Aries with Indy

```
aca-py start --inbound-transport http 0.0.0.0 8000 \  
--outbound-transport http \  
--admin-insecure-mode \  
--admin 0.0.0.0 8001 \  
--seed 00000000000000000000000000Steward1 \  
--replace-public-did \  
--wallet-type indy \  
--genesis-file  
/home/bishakh/Documents/TA/Blockchain/indytutorial/pool1.txn \  
  
--wallet-name agent1 \  
--wallet-key agent1 \  
--log-level debug \  
--log-file /tmp/arieslog.txt  
--admin-insecure-mode \  
--admin 0.0.0.0 8001 \
```

→ **For experimentation and  
debugging only**



# Aries Admin OpenAPI Interface

 Swagger  
Powered by SMARTBEAR

/api/docs/swagger.json

Explore

## Aries Cloud Agent v0.5.1

</api/docs/swagger.json>

### server

GET

/plugins

Fetch the list of loaded plugins

GET

/status

Fetch the server status

POST

/status/reset

Reset statistics

GET

/features

Query supported features

### basicmessage

POST

/connections/{id}/send-message

Send a basic message to a connection

### issue-credential

GET

/issue-credential/mime-types/{credential\_id}

Get attribute MIME types from wallet

GET

/issue-credential/records

Fetch all credential exchange records

GET

/issue-credential/records/{cred\_ex\_id}

Fetch a single credential exchange record

POST

/issue-credential/send

Send holder a credential, automating entire flow





# Conclusion

- Aries
- Communication between participants
- Digital Credentials



*Thank  
you*

