Software Impacts 9 (2021) 100101

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

PyDentity: A playground for education and experimentation with the Hyperledger verifiable information exchange platform

Will Abramson, Pavlos Papadopoulos *, Nikolaos Pitropakis, William J. Buchanan

Blockpass ID Lab, School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK

ARTICLE INFO

Keywords:
PyDentity
Decentralised identifiers
Verifiable credentials
Hyperledger Aries

ABSTRACT

PyDentity lowers the entry barrier for parties interested in experimenting with the Hyperledger's verifiable information exchange platform. It enables educators, developers and researchers to configure and initialise a set of actors easily as associated Hyperledger Aries agents. This allows them to focus on writing domain-specific business logic within a Jupyter notebook interface for each actor. Through this, actors can be customised to determine their connections with others, the messages they send, and how they respond to messages they received. This simple architecture allows for fast iteration and detailed exploration of the possible interactions and use cases that this platform supports. Additionally, notebooks can easily be made self-documenting and easily replicated by multiple parties. PyDentity has already been used in several research and education projects.

Code metadata

Current code version	v0.3.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-64
Permanent link to Reproducible Capsule	
Legal Code License	Apache-2.0
Code versioning system used	Git
Software code languages, tools, and services used	Python, Jupyter Notebooks, Bash
Compilation requirements, operating environments & dependencies	Linux, MacOS, Docker, docker-compose, Source2Image
If available Link to developer documentation/manual	Playground: https://github.com/wip-abramson/aries-jupyter-playground and https://github.com/OpenMined/PyDentity, Video Demonstration: https://www.youtube.com/watch?v=swiA2op3PiQ, Controller: https://github.com/didx-xyz/aries-cloudcontroller-python
Support email for questions	will.abramson@napier.ac.uk

1. Introduction

PyDentity is an open-source project [1], developed within the OpenMined open-source community in order to create a Self-Sovereign Identity (SSI) framework that utilises attribute-based credentials [2]. This framework combines three open-source Hyperledger projects; Ursa, Indy and Aries. Hyperledger Ursa [3] provides generic trust implementations of common cryptographic protocols, including CL-RSA signatures [4] and BBS+ signatures [5,6], both of which can be used to issue privacy-enhanced cryptographic credentials. The Hyperledger Indy project [7] consists of two repositories: (i) Indy-node containing the code to run a node and contribute to consensus within an Indy-based distributed ledger network, and (ii) Indy-SDK providing an interface layer to interact with Indy ledgers, wallet storage and the relevant cryptographic protocols exposed by Ursa. Finally, Hyperledger Aries [8] defines and manages a set of specifications for the implemented agents to establish secure communication channels with other agents and exchange a series of messages associated with a specified protocol [9]. For example, the issue-credential protocol [10], which defines the set of messages and state transitions required for

* Corresponding author.
E-mail addresses: will.abramson@napier.ac.uk (W. Abramson), pavlos.papadopoulos@napier.ac.uk (P. Papadopoulos), n.pitropakis@napier.ac.uk (N. Pitropakis), b.buchanan@napier.ac.uk (W.J. Buchanan).

https://doi.org/10.1016/j.simpa.2021.100101
Received 7 June 2021; Received in revised form 21 June 2021; Accepted 25 June 2021

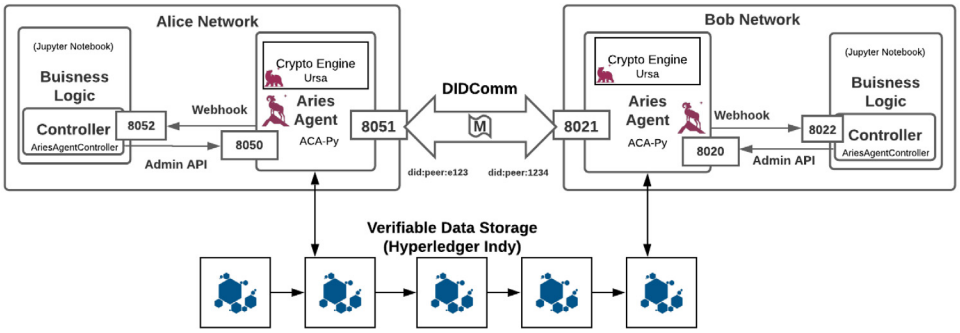

Fig. 1. Aries Jupyter Playground Overview.

two agents in the respective roles of issuer and holder to successfully engage in this protocol such that on completion, the holder has a signed credential from the issuer that they will be able to verifiably present in future interactions with other agents. Open-source code implementing these agents is being produced in a number of different programming languages, including Python, Go, .Net and JavaScript.

2. Description

Our software aims to help researchers and developers experiment and quickly deploy these technologies in multiple use cases. Specifically, within an academic setting, the aim is to simplify the process by which researchers can spin up a set of agents and develop business logic around how these agents interact and exchange verifiable information. Allowing researchers to design and validate specific use cases involving a set of actors, roles, purposes, and information exchanges within an interactive environment.

The outcome was the production of the Aries-Jupyter-Playground [11], which has been extracted into a standalone GitHub repository that anyone can clone and customise accordingly. The playground uses Docker Compose [12] to manage an environment containing the relevant Docker services required for each actor within the playground. These include:

- An Aries-Cloudagent-Python (ACA-Py) instance — This uses a Docker image published and maintained by the open-source ACA-Py project [13]. The agent instance is configurable through an environment file, including defining the Indy network they interact with. An Indy network can be run locally, or a test network such as the Sovrin StagingNet can be used.
- A PostgreSQL database [14] — This stores and persists the state of the agent across multiple instantiations of the environment. Specifically, any cryptographic keys and credential objects they have received.
- A Jupyter Notebook server [15] — This is where custom business logic can be written to model an actor controlling their agent to engage in protocols with other agents within the environment. The ACA-Py agent exposes a Swagger interface [16] and posts events (such as when it receives a message) to a definable endpoint. A custom "pip installable" Python library, the Aries Cloud Controller [17], has been developed to simplify this process within the notebooks. Additionally, each notebook's Docker service has the same recipes folder mounted as a volume providing templates for common interactions, further reducing the time it takes to get a working demo.
- An Ngrok server [18] (optional) — Ngrok can be used to tunnel the HTTP port that the ACA-Py instance exposes to receive messages from other agents to the public internet. This is useful if one wishes to interact with agents not running locally on their computer, for example, mobile agents downloadable from app stores.

The resulting architecture is illustrated in Fig. 1, with two example actors Bob and Alice, as modelled in the default Aries-Jupyter-Playground. The key innovation here is the ability for any set of actors within a system to easily be modelled utilising this environment in a way that minimises the prerequisite knowledge of the different components of the Hyperledger platform and the way they integrate together, which can be complex. Researchers can focus on writing domain-specific business logic within a familiar Jupyter notebook interface. This allows them to explore research related to this technology quickly. The most notable questions include how SSI might be applied to this domain, who are the key actors and what information exchanges need to occur. Although, this playground could also support more technical experiments evaluating the underlying Hyperledger platform. Furthermore, any experiments can easily be made self-documenting using Markdown cells to be straightforward for other researchers to replicate, challenge, or extend.

3. Impacts and future work

This software environment was originally developed to support research into the feasibility of using the mentioned Hyperledger platforms to facilitate the secure communication of privacy-preserving machine learning messages between authenticated actors under a healthcare scenario [19,20]. It has since been applied within the OpenMined community to create education material around decentralised identity as a tool for structured transparency [21]. Additionally, it is currently being used to design a credential ecosystem for the Scottish Healthcare system after engagement with healthcare professionals [22]. Future goals include exploring the possibility for closer integration between the Hyperledger stack, PyVertical [23] and PySyft [24].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank all the contributors and the OpenMined open-source community that participated in this effort. The research leading to these results has been partially funded by the European Union's Horizon 2020 research and innovation programme, through funding of the GLASS project (under grant agreement No 959879).

References

[1] OpenMined, PyDentity, 2021, Available online at https://github.com/OpenMined/PyDentity, Last accessed 01 June 2021.

[2] J. Camenisch, S. Krenn, A. Lehmann, G.L. Mikkelsen, G. Neven, M.Ø. Pedersen, Formal treatment of privacy-enhancing credential systems, in: International Conference on Selected Areas in Cryptography, Springer, 2015, pp. 3–24.

[3] Hyperledger, Hyperledger Ursa, 2021, Available online at https://www.hyperledger.org/use/ursa, Last accessed 01 June 2021.
[4] J. Camenisch, A. Lysyanskaya, A signature scheme with efficient protocols, in: International Conference on Security in Communication Networks, Springer, 2002, pp. 268–289.
[5] M.H. Au, W. Susilo, Y. Mu, Constant-size dynamic k-TAA, in: International Conference on Security and Cryptography for Networks, Springer, 2006, pp. 111–125.
[6] J. Camenisch, M. Drijvers, A. Lehmann, Anonymous attestation using the strong diffie hellman assumption revisited, in: International Conference on Trust and Trustworthy Computing, Springer, 2016, pp. 1–20.
[7] Hyperledger, Hyperledger Indy, 2021, Available online at https://www.hyperledger.org/use/hyperledger-indy, Last accessed 01 June 2021.
[8] Hyperledger, Hyperledger Aries, 2021, Available online at https://www.hyperledger.org/projects/aries, Last accessed 01 June 2021.
[9] W. Abramson, N. Hickman, N. Spencer, Evaluating trust assurance in Indy-based identity networks using public ledger data, Front. Blockchain 4 (2021) 18.
[10] N. Khateev, Issue credential protocol 1.0, 2019, Github requests for comments, Available online at https://github.com/hyperledger/aries-rfcs/blob/master/features/0036-issue-credential/README.md, Last accessed 01 June 2021.
[11] W. Abramson, Aries-Jupyter-playground, 2021, Available online at https://github.com/wip-abramson/aries-jupyter-playground, Last accessed 01 June 2021.
[12] R. Smith, Docker Orchestration, Packt Publishing Ltd, 2017.
[13] Hyperledger, Hyperledger Aries cloud agent - Python, 2019, Available online at https://github.com/hyperledger/aries-cloudagent-python, Last accessed 01 June 2021.
[14] B. Momjian, PostgreSQL: Introduction and Concepts, vol. 192, Addison-Wesley, New York, 2001.
[15] T. Kluyver, B. Ragan-Kelley, F. Pérez, B.E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J.B. Hamrick, J. Grout, S. Corlay, et al., Jupyter Notebooks-a Publishing Format for Reproducible Computational Workflows, vol. 2016, 2016.
[16] V. Surwase, REST API modeling languages-a developer's perspective, Int. J. Sci. Technol. Eng. 2 (10) (2016) 634–637.
[17] DIDx, Aries cloud controller - Python, 2019, Available online at https://pypi.org/project/aries-cloudcontroller/, Last accessed 01 June 2021.
[18] Ngrok, Ngrok service, 2021, Available online at https://ngrok.com/, Last accessed 01 June 2021.
[19] W. Abramson, A.J. Hall, P. Papadopoulos, N. Pitropakis, W.J. Buchanan, A distributed trust framework for privacy-preserving machine learning, in: International Conference on Trust and Privacy in Digital Business, Springer, 2020, pp. 205–220.
[20] P. Papadopoulos, W. Abramson, A.J. Hall, N. Pitropakis, W.J. Buchanan, Privacy and trust redefined in federated machine learning, Mach. Learn. Knowl. Extr. 3 (2) (2021) 333–356.
[21] A. Trask, E. Bluemke, B. Garfinkel, C.G. Cuervas-Mons, A. Dafoe, Beyond privacy trade-offs with structured transparency, 2020, arXiv preprint arXiv:2012.08347.
[22] W. Abramson, N.E. van Deursen, W.J. Buchanan, Trust-by-Design: Evaluating issues and perceptions within clinical passporting, Blockchain Healthc. Today (2020).
[23] D. Romanini, A.J. Hall, P. Papadopoulos, T. Titcombe, A. Ismail, T. Cebere, R. Sandmann, R. Roehm, M.A. Hoeh, PyVertical: A vertical federated learning framework for multi-headed SplitNN, 2021, arXiv preprint arXiv:2104.00489.
[24] A.J. Hall, M. Jay, T. Cebere, B. Cebere, K.L. van der Veen, G. Muraru, T. Xu, P. Cason, W. Abramson, A. Benaissa, et al., Syft 0.5: A platform for universally deployable structured transparency, 2021, arXiv preprint arXiv:2104.12385.