

TIIME2026

Session title	IdentityPython (https://idpy.org/)
Scribe(s)	Laura Paglione; Ivan Kanakarakis

Matthew Economou demonstrated a mocked up test SAML flow in python. He used this to create a test platform to test SAML interoperability : <https://federationcheck.incommon.org/>

<https://github.com/xenophonf/mock-saml-flow>

What would be useful for those using Identity Python tools?

[SaToSa](#) is a helpful tool, however...

- “As a proxy I always go to SaToSa”, though, as a library, they are too complicated to configure for a prototype use case
- Perhaps we should not spend any effort, time, money on SAML... OIDC / Federation tools could be a better strategy – but folks are still using SAML – to what degree to we continue to support them – Maintenance only or continue to add new functionality?

What’s the difference between the different efforts?

- Identity Python as an organization is relatively new - pySaml2 is the initial library that came to be a thing 16yrs ago–
- SaToSa - in practice, this effort is using things that flask is doing; goal to make it easy to just “plug in” this work
- Web – flask <https://flask-saml2.readthedocs.io/en/latest/>
- django Saml - <https://github.com/grafana/django-saml2-auth>

Challenges for Identity Python

- Documentation and links to other efforts are difficult to find for Identity Python tools
- Additional plugins for other frameworks would be helpful
- There was a sunit project that Roland worked on with OpenIDFed. Perhaps we bring this into Identity Python. – there has been some work, but it’s not in the main branch at the moment.
 - The library would need to be stable at some point – eventually available with other tools
 - Needs new front end and support
 - Does the hype match the need?

- *Given the limited resources that this community has, we should be selective about what we provide and fill a gap that won't be done by someone else*

Where should we focus efforts?

- *Fill documentation gap: Better / more documentation*
- *Address the following use cases within SaToSa – differing opinions:*
 - *Enable proof of concept uses – libraries that are suited for straight-forward use cases with a high degree of ease of use. (plug in / abstraction layer) – “Python is my go-to language for proof of concept” (satosa is good a good defaults)*
 - *Improving SaToSa - Flexibility for middle layer / proxy layer – SaToSa is the main thing being used*
 - *Testing*
 - *Additional flexibility*
 - *Create a pre-packaged solution for specific use cases? (EntreID) – or maybe create this in documentation?*
- *PySAML2 – make it easier to integrate with other frameworks (though not as well used as SaToSa)*
- *Re-establish the developer calls – increase engagement and feedback models*
 - *Will there be sufficient interest in participating?*
 - *What is the ideal cadence?*
 - *Increased promotion – who should be in this conversation?*
 - *CERN (Hannah Short)*
 - *SURF*
 - *CILogon*
 - *InAcademia (GÉANT)*
 - *Core AAI Platform (GÉANT)*
 - *AARC Compendium – software stacks to implement proxies – should SaToSa be included?*
 - *EntreID Tenants << available as a service... EduID as a proxy << SUNET*
 - *DAASI*
 - *Some commercial companies*
- ***PyFF** – Still maintained by SUNET (one guy is still working on this) – **may be a tool uniquely provided by IdentityPython***

You can subscribe to the mailing list here:

<https://lists.sunet.se/postorius/lists/idpy-discuss.lists.sunet.se/>

PyInfra for infrastructure vs Ansible

- *PyInfra does the job; one participant mentioned not choosing PyInfra, but it was purely because of non-technical reasons*

Federation Adapter (EntreID doesn't support multilateral federation – it needs an adapter)