Automation of SP deployment

Petrunin Aleksandr

GEANT T&I Incubator

Public Sprint Demo

GÉANT
Networks · Services · People

Abstract

We want to solve challenges in

❑ Manual processes in deployment

❑ Configuring SPs within emerging federations

→ Simplify the initial setup through automation

Current frustrations

of emerging and low-resource federations

Infrastructure setup

Resource constraints

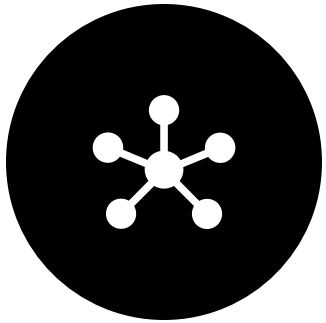Diverse technologies

Trust establishment
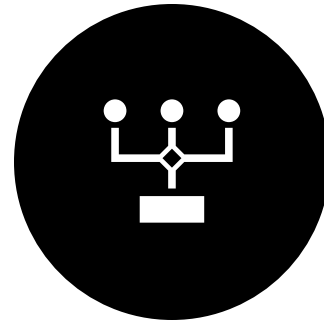
Manual processes

Problem statement

WHAT: Challenges in setting up and maintaining the federation's initial set of SPs

WHY: Manual processes, diverse technologies, and a steep learning curve for federation ops
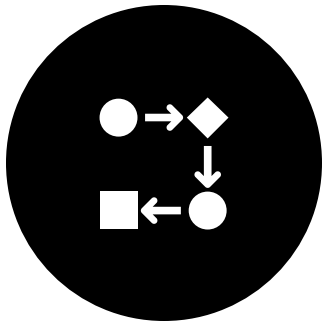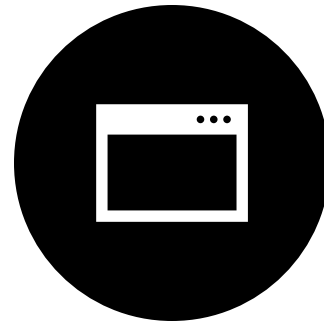
Proposed solution

Deployment of a proxy in front of SPs

Deployment of ready-built applications

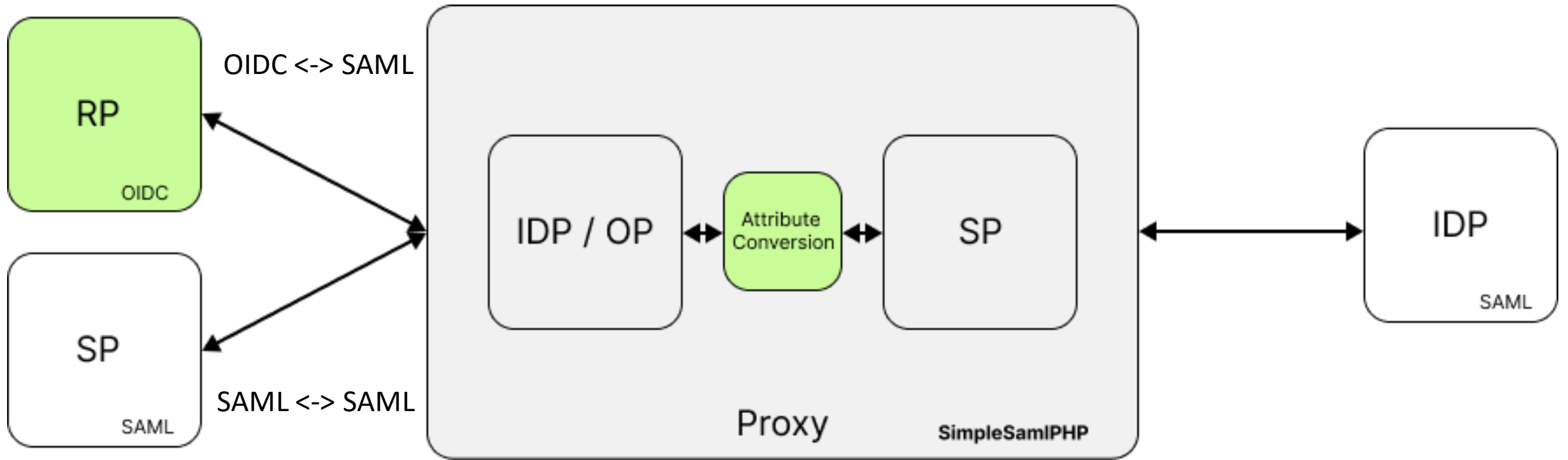Proxy handles business logic, reducing the burden on federation Ops
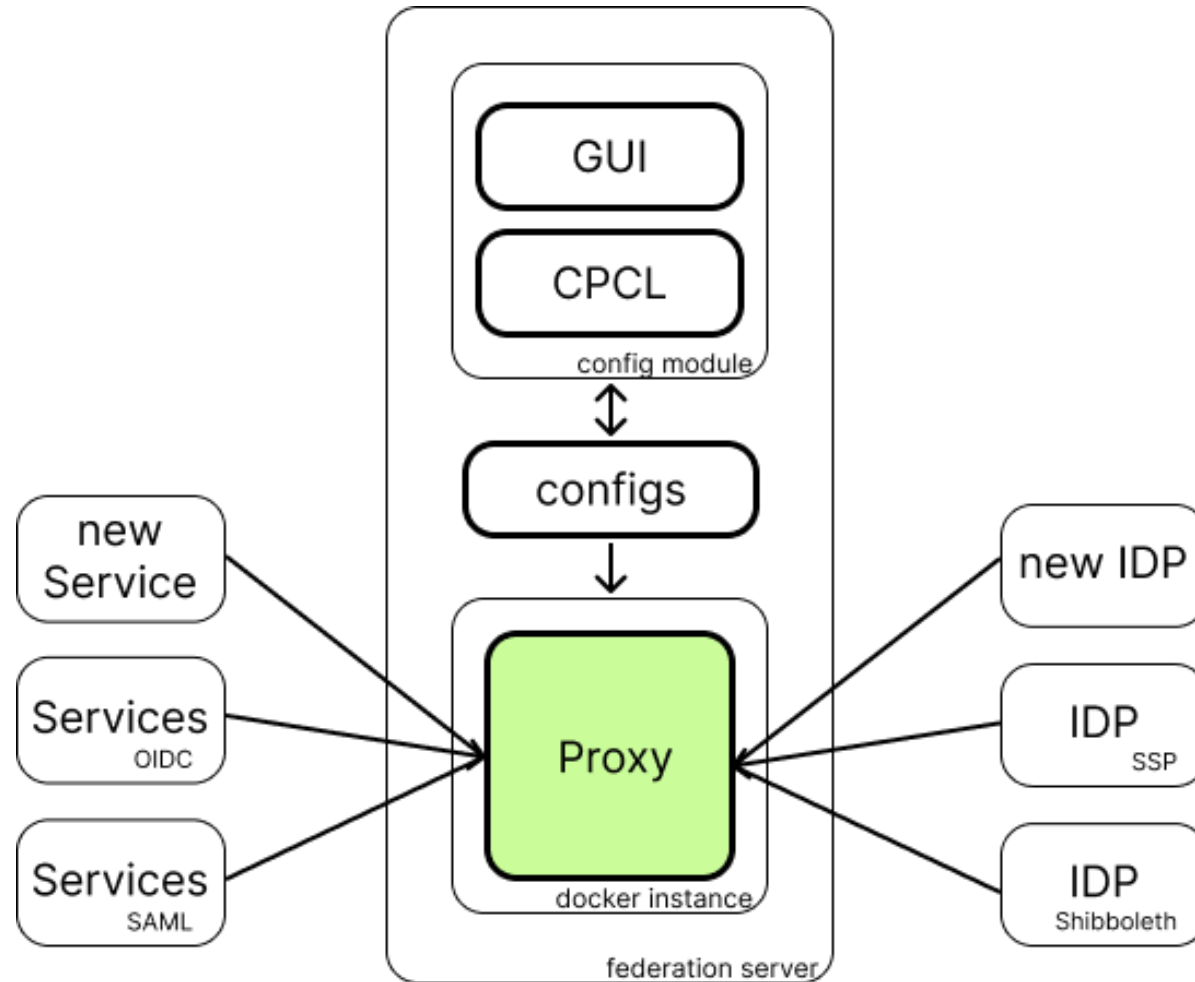
GUI for easy selection and deployment

Solution advantages

❑Reduced complexity of dealing with configuration permutations

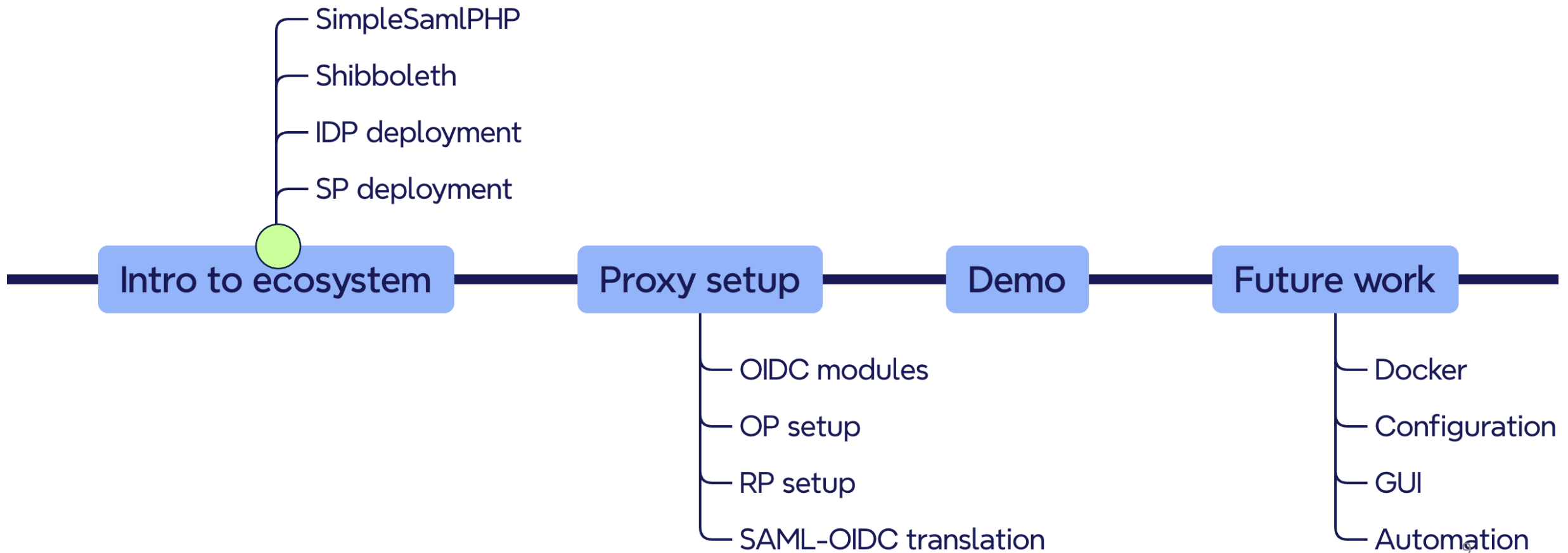❑Streamlined deployment of SPs

Current setup architecture

High-level architecture

Milestones & contributions

❏ Delving into T&I core concepts

❏ Progress on initial proxy development

SimpleSamlPHP

Shibboleth

IDP deployment

SP deployment

Intro to ecosystem

Proxy setup

Demo

Future work
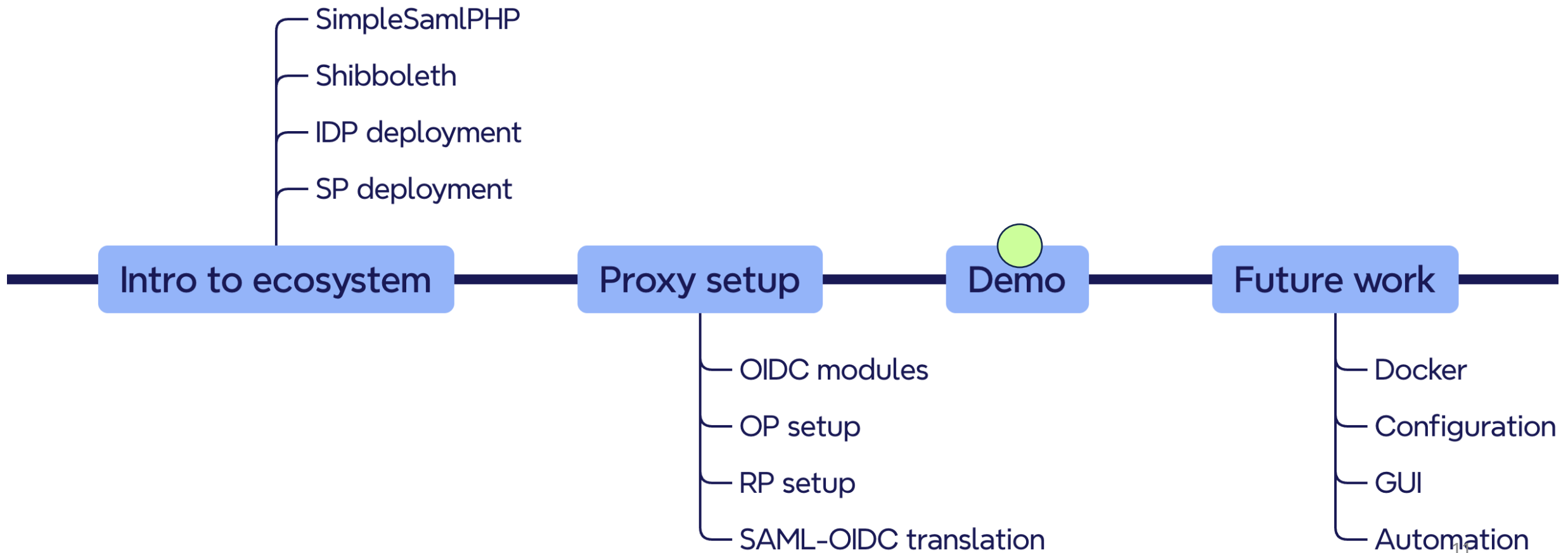
OIDC modules

OP setup

RP setup

SAML-OIDC translation

Docker

Configuration

GUI

Automation

Milestones & contributions

❑ Delving into T&I core concepts

❑ Progress on initial proxy development

SimpleSamlPHP

Shibboleth

IDP deployment

SP deployment

Intro to ecosystem

Proxy setup

Demo

Future work

OIDC modules

OP setup

RP setup

SAML-OIDC translation

Docker

Configuration

GUI

Automation

Milestones & contributions

❑Delving into T&I core concepts

❑Progress on initial proxy development

SimpleSamlPHP

Shibboleth

IDP deployment

SP deployment

Intro to ecosystem

Proxy setup

Demo

Future work

OIDC modules

OP setup

RP setup

SAML-OIDC translation

Docker

Configuration

GUI

Automation
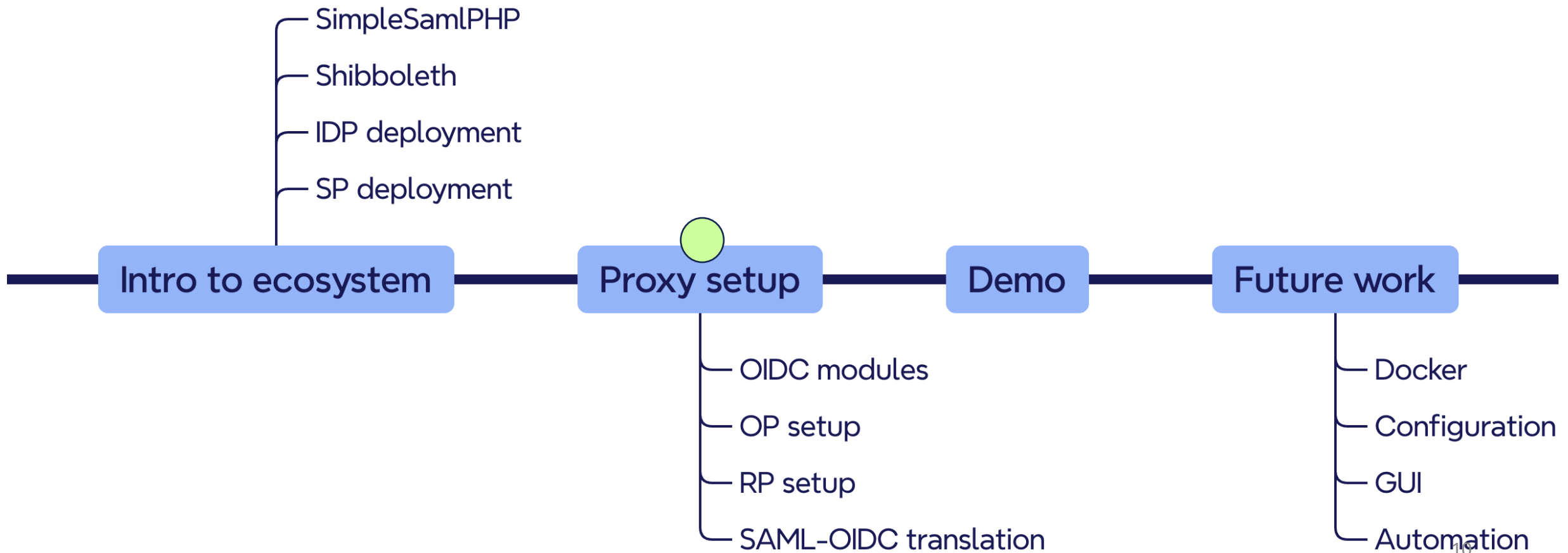
SimpleSAMLphp: Proxy

Configuration   Test   Federation   Log out

Hosted entities

default-sp
EntityID: `https://alpe3.incubator.geant.org/saml-sp`
Type: **SAML 2.0 SP metadata**

⌄

https://alpe3.incubator.geant.org/saml-idp
EntityID: `https://alpe3.incubator.geant.org/saml-idp` (hostname: default)
Type: **SAML 2.0 IdP metadata**

⌄

Trusted entities

SAML 2.0 IdP metadata

https://alpe1.incubator.geant.org/saml-idp

SAML 2.0 SP metadata

SimpleSAMLphp

OpenID Connect Client Registry

Search   Reset

Client list   **+   Add client**

● **Test RP**   👁 ✎ 🗑

‹   1   ›

Demo: OIDC <-> SAML

SimpleSAMLphp: RP

Configuration   Test   Federation   Log out

You are running an outdated version of SimpleSAMLphp. Please update to the latest version as soon as possible.

SimpleSAMLphp is installed in: `/var/simplesamlphp/`
You are running version `2.0.6`.

Modules

You have the following modules installed (🚫 means the module is not enabled):

🚫 SAML 2.0 IdP	✔ core	✔ mymodule
🚫 adfs	🚫 cron	✔ saml
✔ admin	✔ exampleauth	🚫 themes
✔ authoauth2	🚫 multiauth	

Details

Diagnostics on hostname, port and protocol
Information on your PHP installation

Your PHP installation

Demo: SAML <-> SAML

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.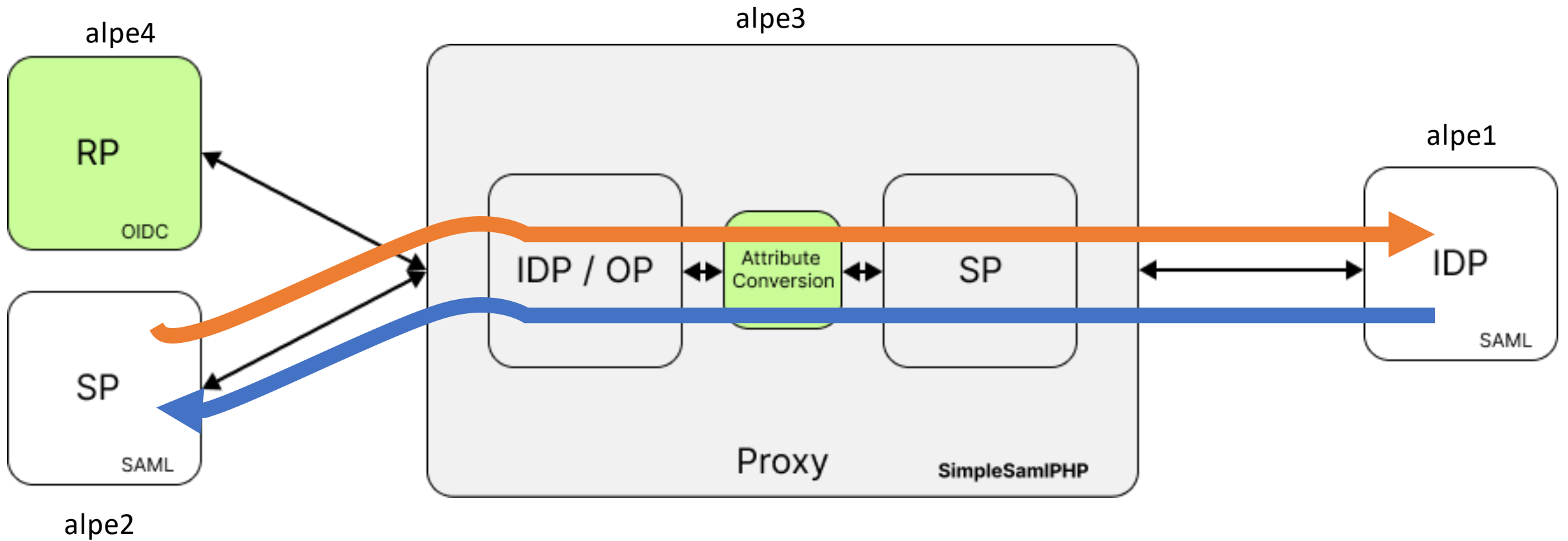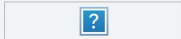 You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

Milestones & contributions

❑ Delving into T&I core concepts

❑ Progress on initial proxy development

```
                  ┌── SimpleSamlPHP
                  ├── Shibboleth
                  ├── IDP deployment
                  └── SP deployment

[Intro to ecosystem]────[Proxy setup]────[Demo]────[Future work]

                              ┌── OIDC modules              ┌── Docker
                              ├── OP setup                  ├── Configuration
                              ├── RP setup                  ├── GUI
                              └── SAML-OIDC translation     └── Automation
```

Milestones & contributions

❑Delving into T&I core concepts

❑Progress on initial proxy development