



# OpenCAPIF

*by ETSI*

## Tutorial OpenCAPIF Hackathon Demo

Jorge Moratinos, Stavros Charismiadis and Pelayo  
Torres

2/2/2026

# Agenda

---

- ① Introduction
- ① Hands On:
  - ② The big picture
  - ② Deploy Local API Service
  - ② Creation of ProviderGW
  - ② Creation of Invoker with SDK
- ① Other templates

# Introduction

# Introduction

---



At this Hands-on we will see how to use OpenAPIF from the perspective of Invoker and Provider Entities, or in other words "What we need to develop to use OpenAPIF".

For that we will explain in next section some templates and examples that you can try today towards OpenAPIF public instance at ETSI premises.

# Introduction

---

## Laptop Readiness: What you'll need:

- ① Laptop with internet access
- ① Docker client installed
- ① Python 3.12
- ① Git client
- ① Development environment (visual studio code as example)
- ① Postman

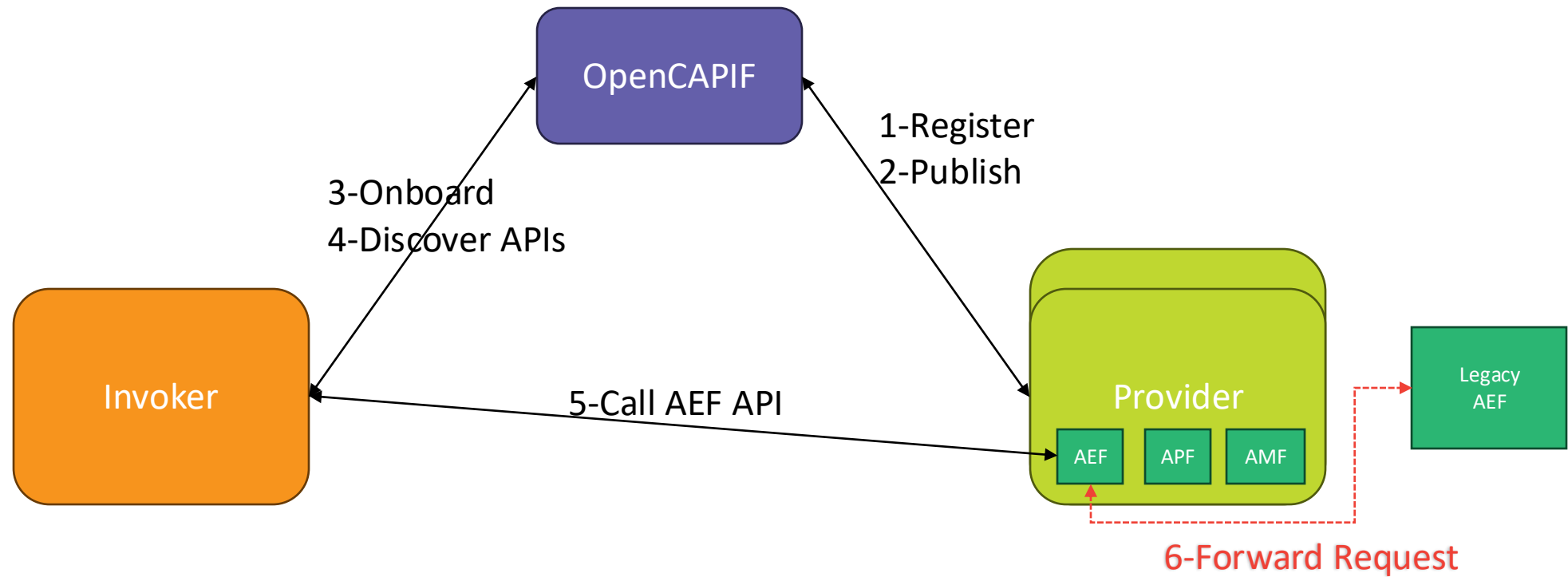
## For Windows Users:

- ① WSL (Windows Subsystem For Linux)

# EU-Funded Research & Innovation Projects



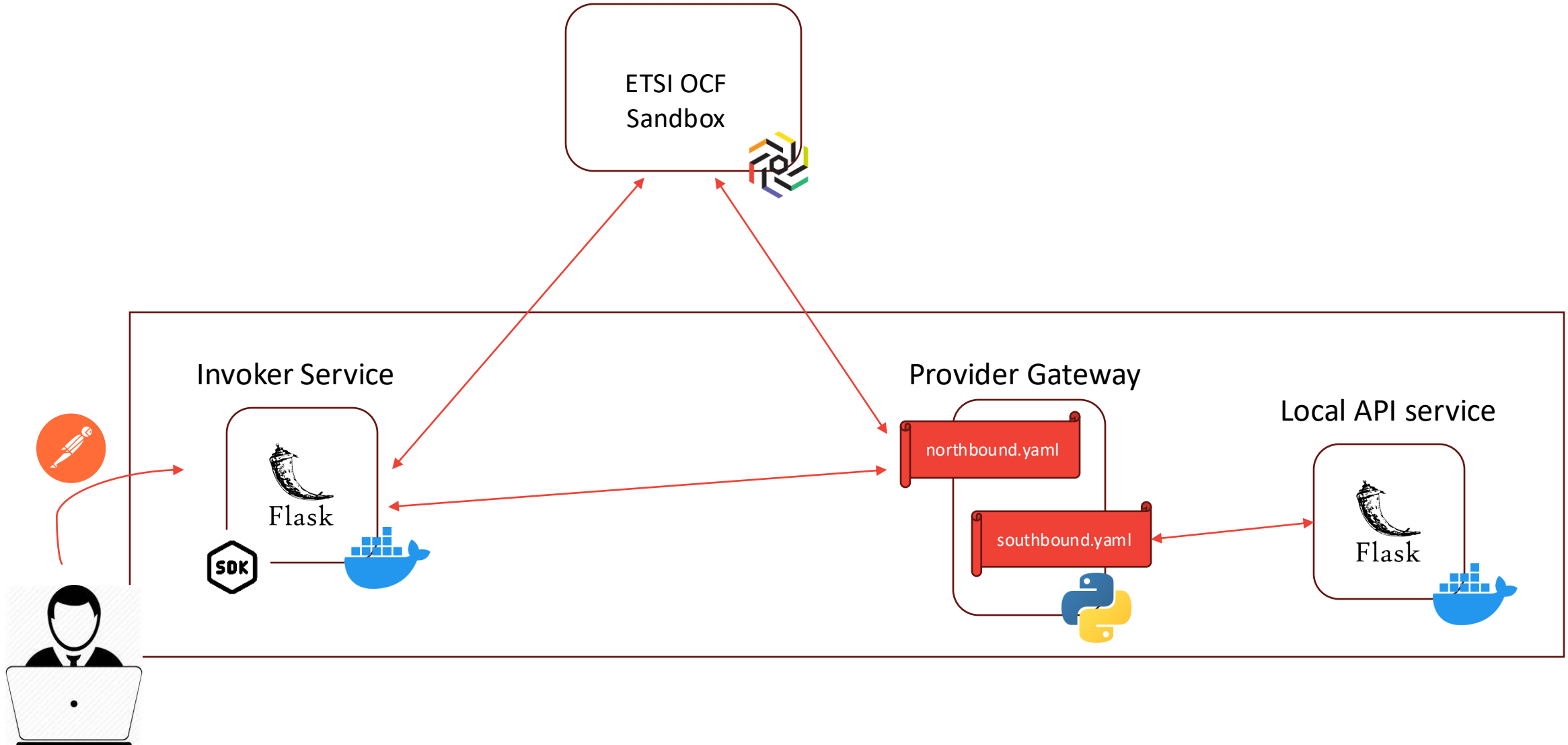
# Entities related



# Hands-On

# The big picture

<https://register-opencapif.etsi.org>  
<https://opencapif.etsi.org>

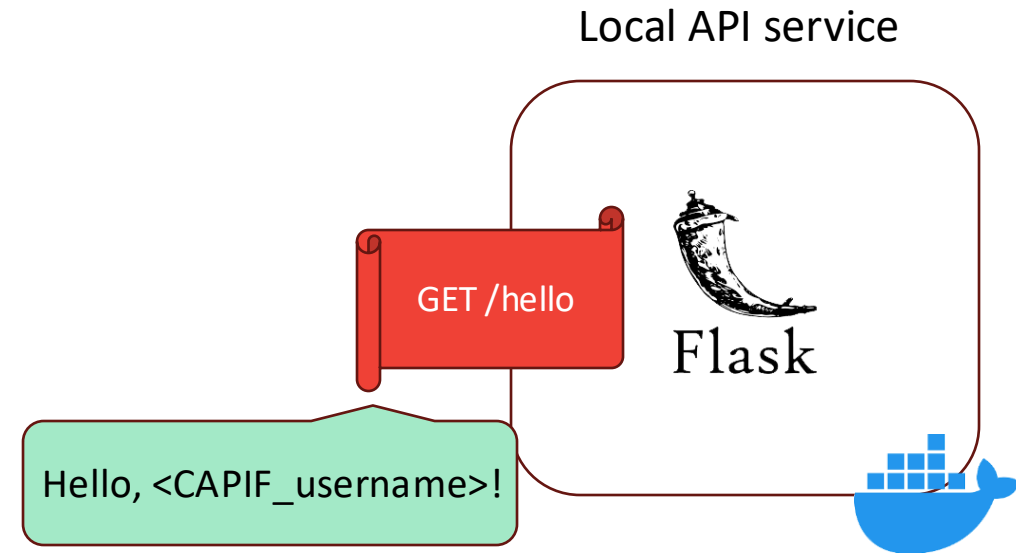


# Deploy Local API Service

# Deploy Local API Service

*What is it?*

- *Dockerized Flask server application*
- *Exposes a simple GET API request*



# Deploy Local API Service

Considering you already have Docker installed ... 😊

- Download the code from [here](#)

```
git clone https://labs.etsi.org/rep/ocf/tools/dummy-aef.git
```

- To deploy the component ...

```
cd dummy-aef  
chmod +x start_dummy_api.sh  
./start_dummy_api.sh <Username provided by organizers>
```

*Additional bash scripts*

```
./logs_dummy_api.sh <Username provided by organizers>  
./stops_dummy_api.sh <Username provided by organizers>
```

- Edit north/southbound yaml files

# Creation of ProviderGW

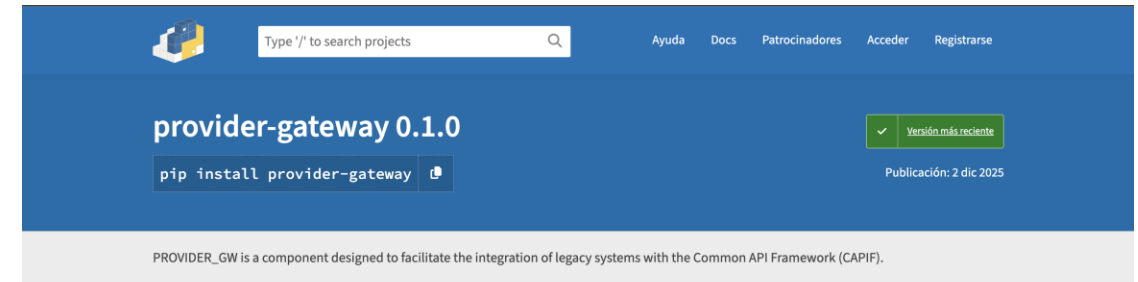
# What is the Provider GW?

---

- PROVIDER\_GW is a gateway component designed to bridge any API with OpenAPIF
- It simplifies the integration process, enabling legacy systems to interact seamlessly with OpenAPIF and adopt modern APIs without extensive modifications.
- The component leverages the [Openapif\\_sdk](#) to manage the interactions between systems and OpenAPIF effectively.
- It also uses a simple configuration with only two yaml files where the legacy system and the connection to APIF are specified.
- Provider\_GW supports exposed APIs with Token-based security, Basic Authentication, and no authentication method.

# How does it work?

- You can install it directly with pip, as it is published on PyPi as an official package.
- **PROVIDER\_GW** acts as an intermediary, translating and exposing APIs from in a format compatible with OpenCAPIF.
- It manages communication between the **Southbound interface** (Exposed API) and the **Northbound interface** (OpenCAPIF).



**Navegación**

- Descripción de proyecto
- Histórico de versiones
- Archivos de descarga

**Descripción de proyecto**

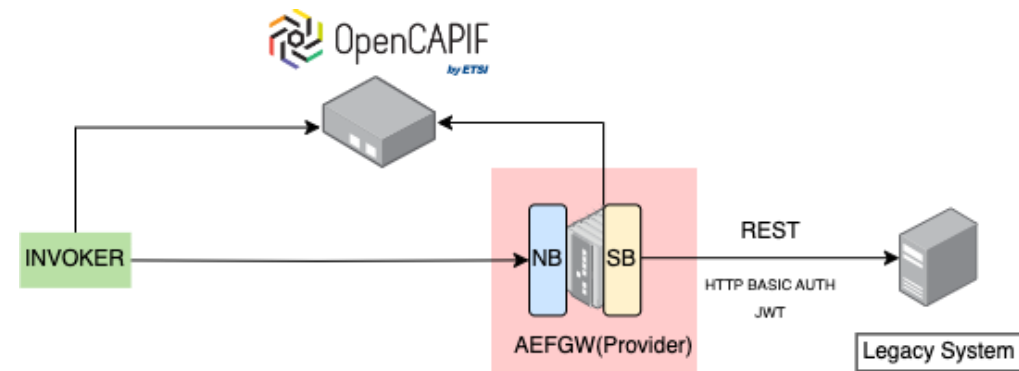
**PROVIDER\_GW**

pypi v0.1 python v3.12+

PROVIDER\_GW is a gateway component designed to bridge legacy systems with the **Common API Framework (CAPIF)**. It simplifies the integration process, enabling legacy systems to interact seamlessly with CAPIF and adopt modern APIs without extensive modifications.

The component leverages the [Opencaapif sdk](#) to manage the interactions between systems and CAPIF effectively.

**Verified details**   
These details have been verified by PyPI



# Deploy Provider\_GW

---

- Required Python version greater than 3.12

- Download the code from [here](#)

```
git clone https://labs.etsi.org/rep/ocf/tools/dummy-aef.git
```

- Install Provider-gateway with PyPi

```
pip install provider-gateway
```

- Edit north/southbound yaml files

- Use the commands to control the gw provider

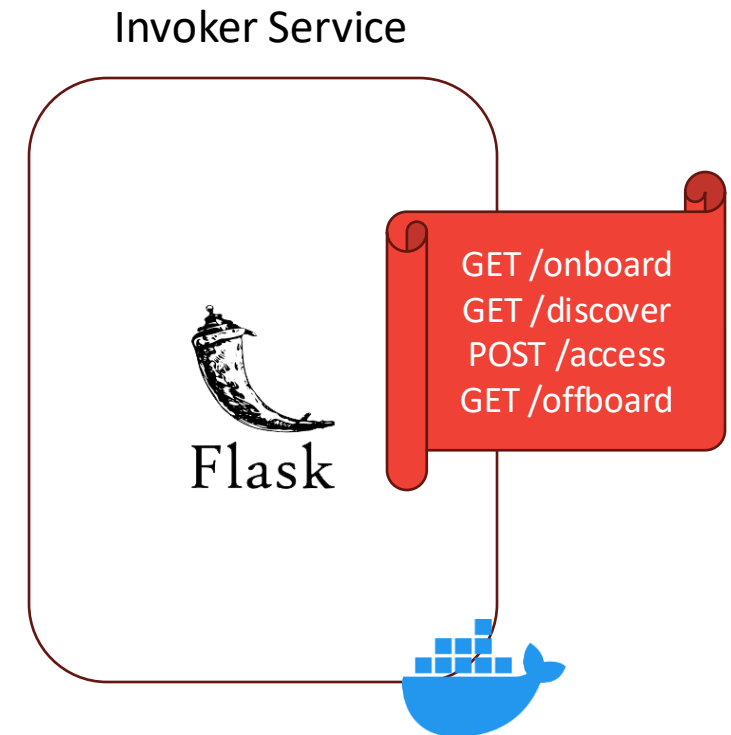
- *provider\_gw start* -> Start the provider\_gw from scratch
- *provider\_gw run* -> Start the provider once you have published the API.
- *provider\_gw remove* -> Delete all provider info in OpenCAPIF

# Creation of Invoker with SDK

# Deploy Invoker Service

## What is it?

- *Dockerized Flask server application*
- *Not fully functional ... yet ... This is our task 😊*
- *Exposes a set of APIs to trigger invoker functionalities*
  - *GET /onboard => Onboard Invoker*
  - *GET /discover => Discover API service (published from Provider GW)*
  - *POST /access => Get authorization and call the service API*
  - *GET /offboard => Offboard Invoker*



# Deploy Invoker Service

---

Considering you have already installed ...

- Docker
- Postman
- Download the code from [here](#)

*git clone <https://labs.etsi.org/rep/ocf/example-clients/invoker-python.git>*

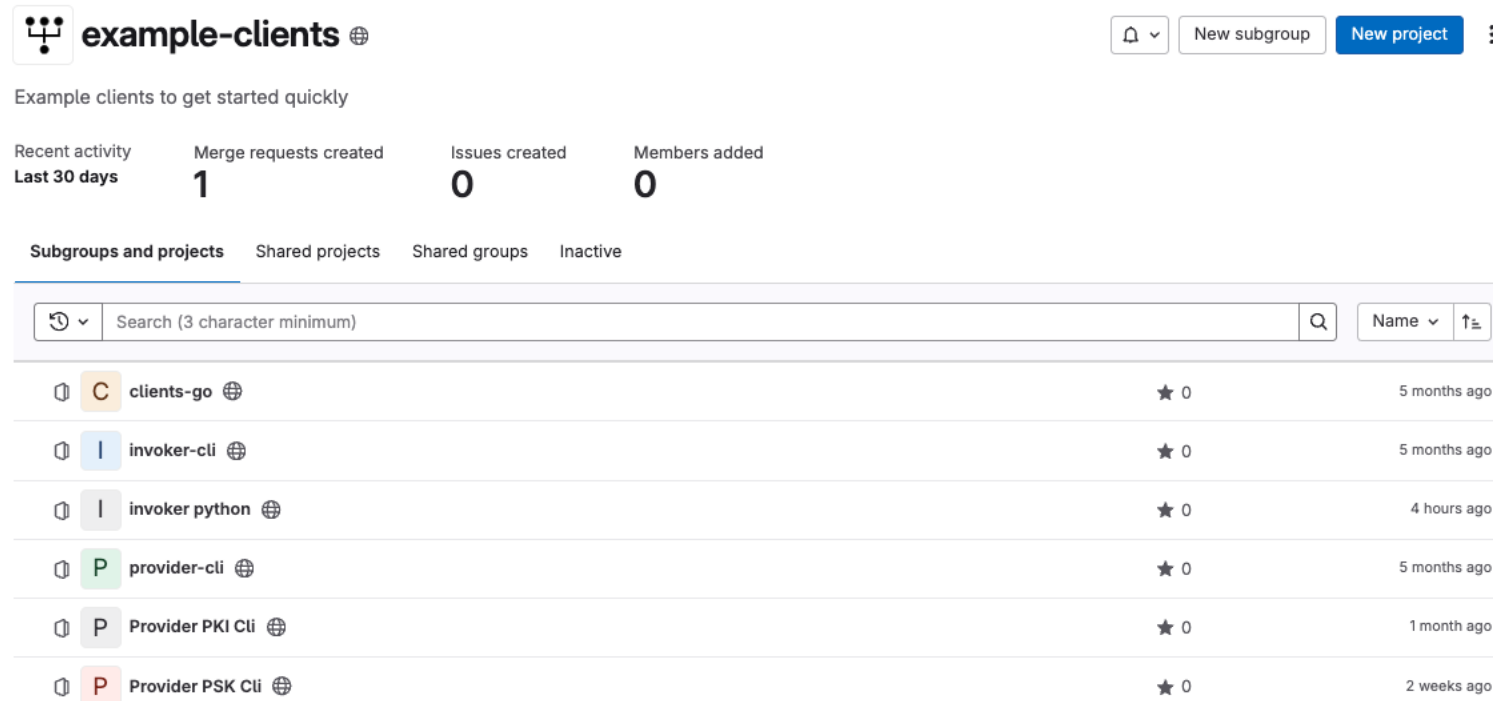
- Follow the instructions on README.md


*Let's go step by step ...*

# Other templates

# Other templates

OCF SDG also provides many others templates to implement your own providers. You check example-clients at gitlab repository <https://labs.etsi.org/rep/ocf/example-clients>





















**example-clients** 

Example clients to get started quickly

Recent activity **Last 30 days** Merge requests created **1** Issues created **0** Members added **0**

**Subgroups and projects** Shared projects Shared groups Inactive

Search (3 character minimum)   Name

  <b>clients-go</b> 	★ 0	5 months ago
  <b>invoker-cli</b> 	★ 0	5 months ago
  <b>invoker python</b> 	★ 0	4 hours ago
  <b>provider-cli</b> 	★ 0	5 months ago
  <b>Provider PKI Cli</b> 	★ 0	1 month ago
  <b>Provider PSK Cli</b> 	★ 0	2 weeks ago

# Other templates

---

Currently Provider GW only support OAUTH, SDG will work to upgrade it in order to support other security methods, but meanwhile if you want to implement your own provider with PKI or PSK security methods you can take a look to next example clients at respository:

- Provider PKI client:
  - <https://labs.etsi.org/rep/ocf/example-clients/provider-pki-cli>
- Provider PSK client
  - <https://labs.etsi.org/rep/ocf/example-clients/provider-psk-cli>

Both examples includes a README.md with all information to try it.

# Other templates

---

Key points of both example clients:

- Include and example of implementation of **AEF\_Security\_API**, defined by **3GPP** as requirement to be compliant with **OpenCAPIF**.
- Implementation by using docker compose and includes many scripts to simplify management.
- You can control clients by APIs and include postman collection.
- Interaction with **OpenCAPIF** based on **OCF SDK**.
- Developed in python with Flask.



OpenCAPIF

*by ETSI*

Thank You!