Introduction to CSPA Modules

Romain Tailhurst (INSEE)
Trygve Falch (SSB)
“TIMELINE-OVERVIEW”

Explore

Design

Build

Assemble/Deploy

Configure

Use

فرح

[Placeholder: SSB VTL Example]

[Placeholder: ONS Collection Example - hosting in cloud env]

[Placeholder: possible SCB demo of process designer]

(Production)

[Based on Developed and Deployed Services - Configure for CPI and GDP]

DEMO

DEMO

DEMO

DEMO
CSPA challenges today

- Re-usability of services is difficult in its current state
- Different NSI’s have different investments in different technology platforms
- CSPA lacks good guidance for how services should be implemented and made shareable.
Friction points

- “My data layer is Postgre not MySQL”
- “I’m using an event broker, i can’t use your REST endpoint”
- “I don’t have an ESB!”

- Many other friction points related to information architecture, etc.

⇒ CSPA Modules to the rescue!
CSPA Modules

- A CSPA Module is a small piece of functionality that communicates with the core business-logic of the service, and with the outside world.
- Modules are optional for a CSPA service, but it will make sharing easier if you decide to use it.
- A module will be the technical “glue” for binding the CSPA Service to a local NSI’s environment while still retaining the core business logic of the Service.
Different layers that affect sharing

- Packaging and distribution
- CSPA Service w/modules
- Core Logic

Module
Shareability scale

Not modular

“"I get an immediate benefit from this service even if I should adapt my process to the communication module”

Fully modular

“I can change the data adapter module of this service by developing my own version but I still get the core logic”
REST Validation service example

INSEE

Validation logic (Java)

Rest API module

PostgreSQL JDBC Data adaptor Module
Event driven Validation service example

- Event handler module
- DataStore API
- Data adaptor Module

Validation logic (Java)
HOW TO MAKE MODULES HAPPEN?

Software engineering principles + OPEN-SOURCE Collaboration