Workshop Workfloworchestrator

Peter Boers

5th of June, 2023

SSID: MakAlbania-MeetingRoom

WIFI Password: Tirana2023

5 June, 2023

Agenda

- 9:00 – 9:30 Introduction
- 9:30 – 10:00 Bootstrapping of the development environment
- 10:00 – 10:30 Details of the Orchestrator core models
- 10:30 – 11:00 Coffee 
- 11:00 – 12:30 Development of your first Orchestrator workflow
- 12:30 – 14:00 Lunch (Rogner Hotel) 
- 14:00 – 15:30 Integration of OSS and BSS to your workflow
- 15:30 – 16:00 Coffee 
- 16:00 – 17:30 Tailoring the Orchestrator to your needs (Discussion)

SURF

Who do we have in the room?

- To get to know each other please do the following:
- State your name
- Affiliation
- What you hope to achieve during this workshop.
- What knowledge and/or experience you bring with you from your home institution.

What is the goal of this workshop?

- The goal of the workshop is to have a reasonable understanding of the following concepts of the workfloworchestrator software:
- Service modelling: Subscriptions, Products and Domain models
- The workflow engine and the anatomy of a workflow and it’s steps
- Have an idea of how to integrate an OSS/BSS software and how you can define sources of truth
- Learn from other NREN’s and discuss Automation and Orchestration use cases
- Create an understanding of how the workfloworchestrator ecosystem of software could help achieve an NREN’s orchestration goals.

AUTOMATION

The automatic operation or control of equipment, a process, or a system. This often encompasses a linear process.

ORCHESTRATION

The execution of (multiple) automations to achieve the desired state of a process or system.

Why automation & orchestration…

Orchestration by example….

- What does the orchestrator solve for SURF?
- Reliable data administration
- Fast data administration
- Accurate data administration
- Reliable service provisioning
- Fast service provisioning
- Accurate service provisioning
- Single pane of glass on all services provisioned on our (inter)national network.

Orchestration in the context of the workfloworchestrator

- It executes arbitrary python functions on objects and stores the result of each function in the database. These functions are called steps.
- A collection steps that follow on each other are called workflows
- Workflows can be run to execute arbitrary tasks, but
- … are usually run on products and/or subscriptions.
- Workflows; create, modify, terminate and validate subscriptions and automate lifecycle tasks
- This creates an audit-trail for each subscription so you can see all actions that have been executed on each subscription
- The workflow engine orchestrates, automations.

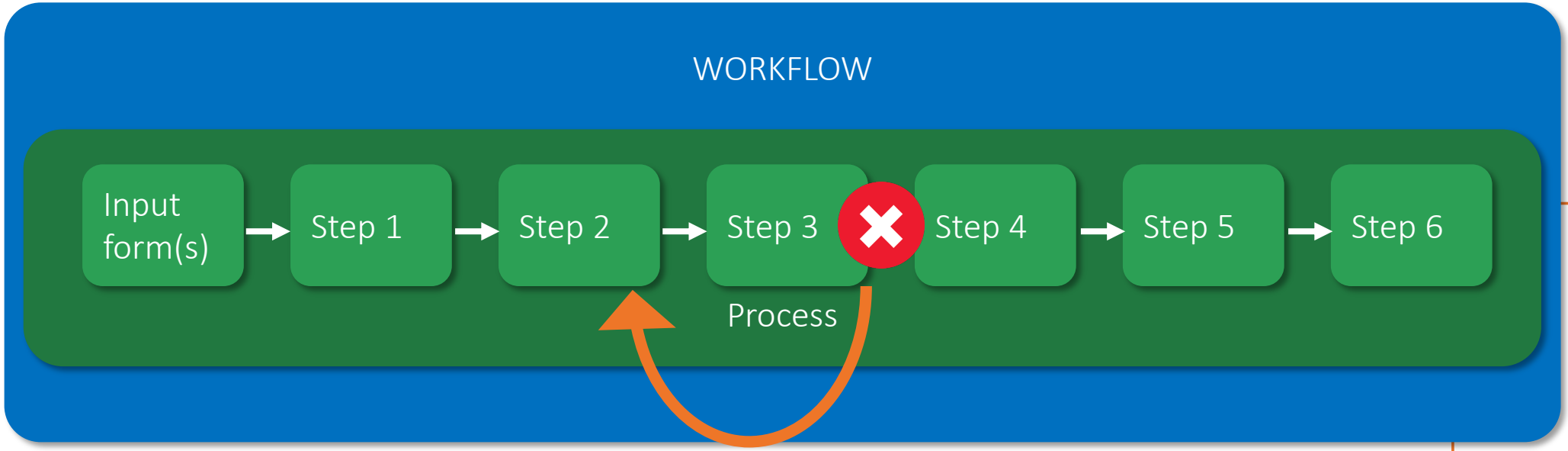
SURF

Orchestration concepts

- Orchestration is executed on higher order abstractions
- The goal is to not only achieve valid network configuration, but to make it possible to define relations between all the data that is needed to provision an arbitrary service
- Inventory
- Customer data
- IP address manangement
- Ticketing
- Planning
- Modelling of abstractions tries create logical relationships between resources that are necessary to provision a service (of any type). The orchestration then makes it possible to define the state of each resource during the lifecycle of a subscription.

SURF

Workflow Engine



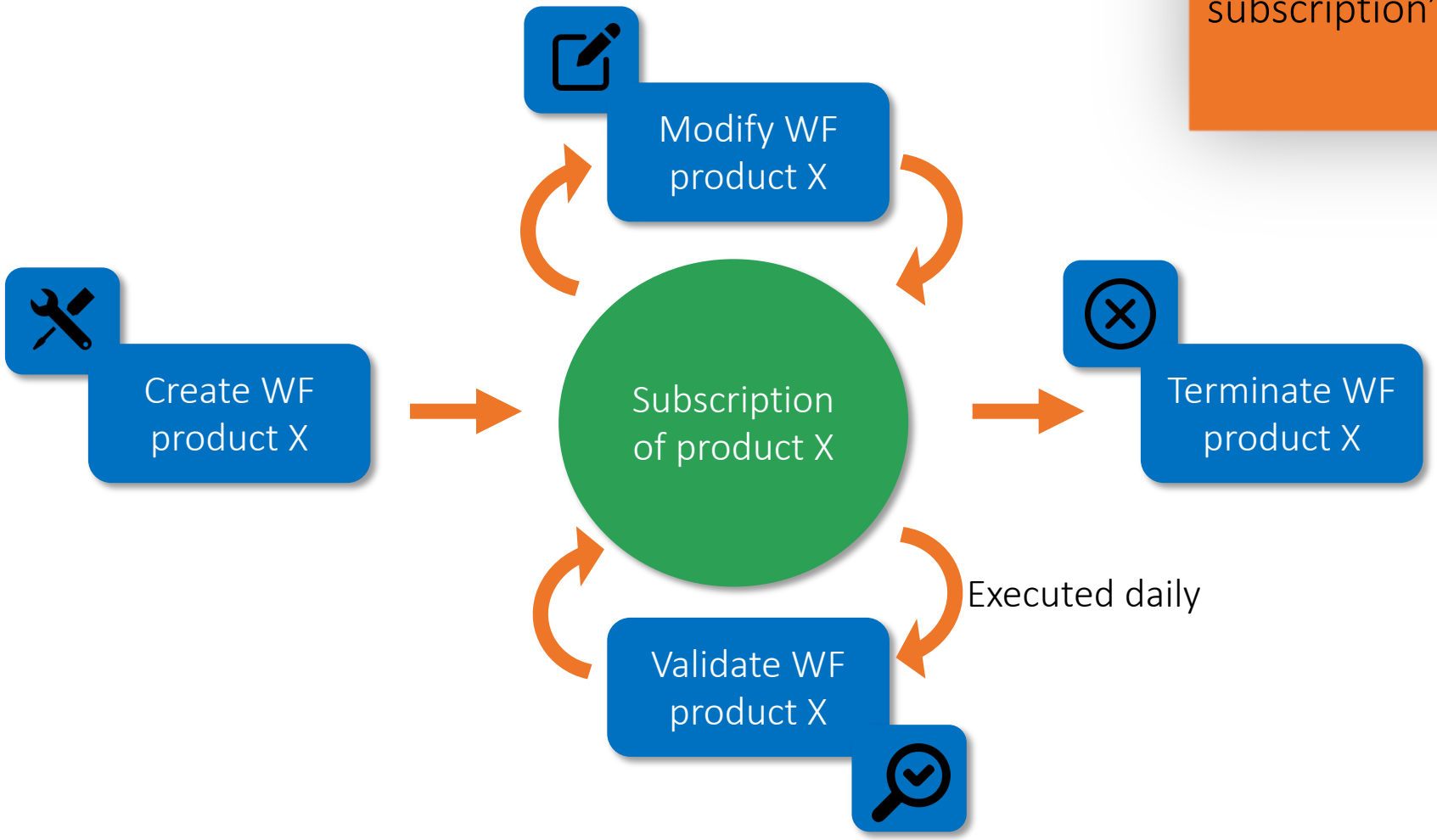
- Each Step writes the state to the database and is used as input for the next step
  - Each (atomic) Step can be retried, making the workflow robust

Workflow Code

```
236 @create_workflow("Create SURFnet8 L2VPN", initial_input_form=initial_input_form_generator)
237 def create_sn8_l2vpn() -> StepList:
238     return (
239         begin
240         >> construct_l2vpn_model
241         >> store_process_subscription(Target.CREATE)
242         >> create_ims_circuit
243         >> create_nso_service_model
244         >> re_deploy_nso
245         >> take_ims_circuit_in_service(is_redundant=False)
246         >> send_confirmation_email()
247     )
248
```

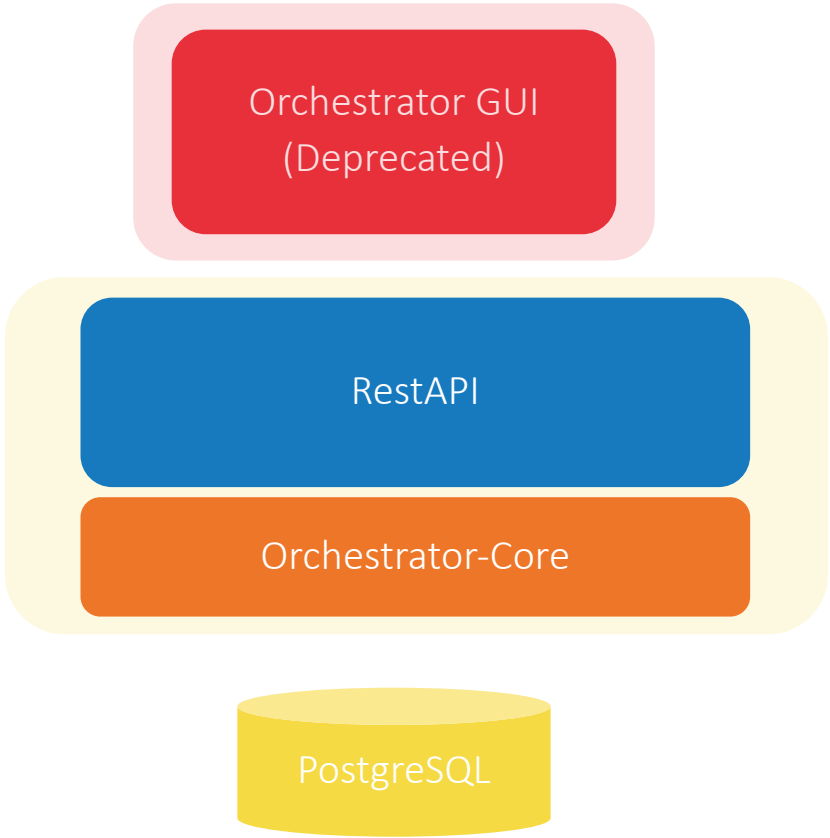
Lifecycle of a service

"a service is an instance of a product, and called subscription"



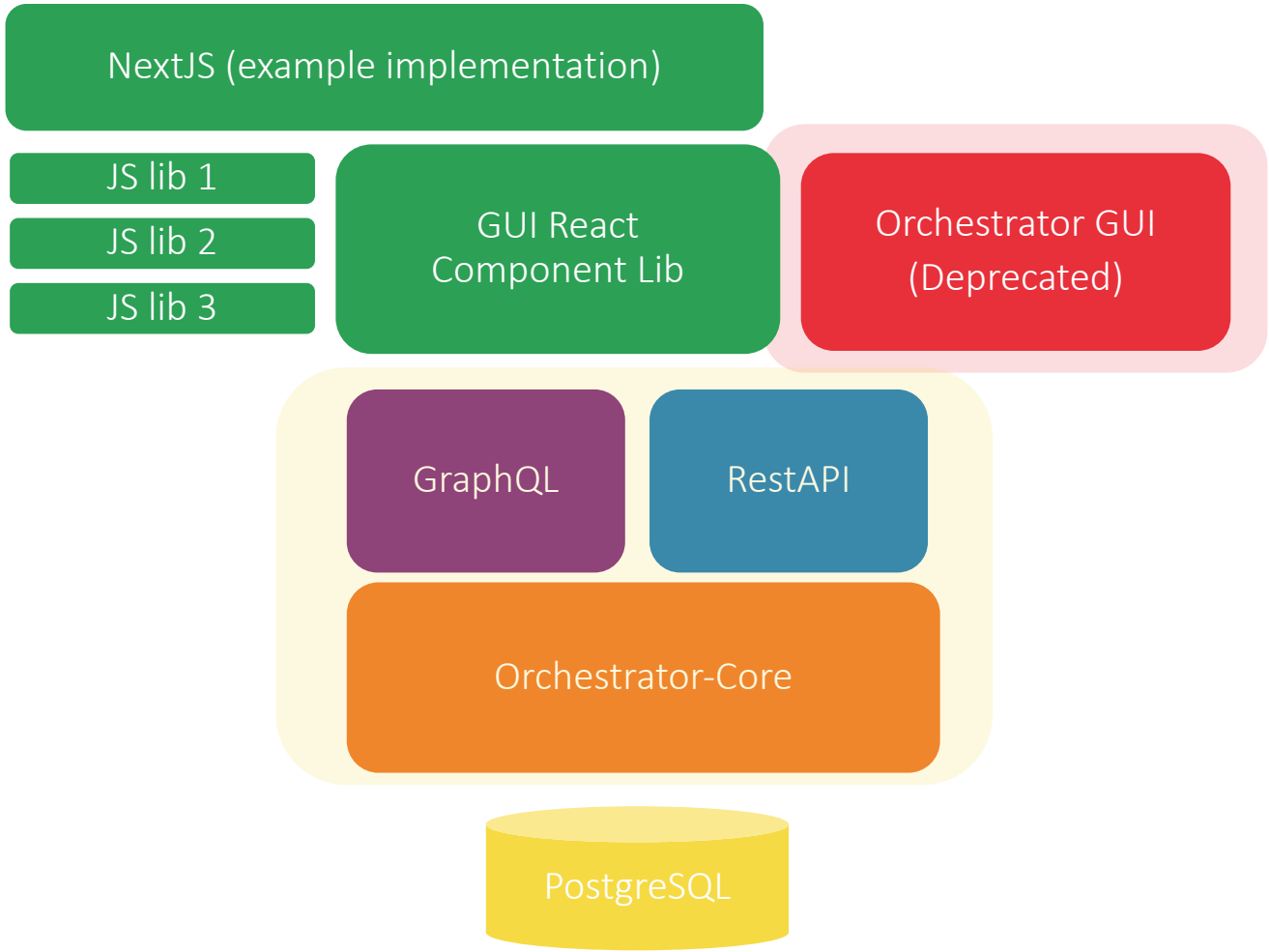
The orchestrator application architecture (Workshop)

- Python API based on FastAPI and Pydantic
- Rest
- PostgreSQL database
- React application
- EUI components
- Uniforms



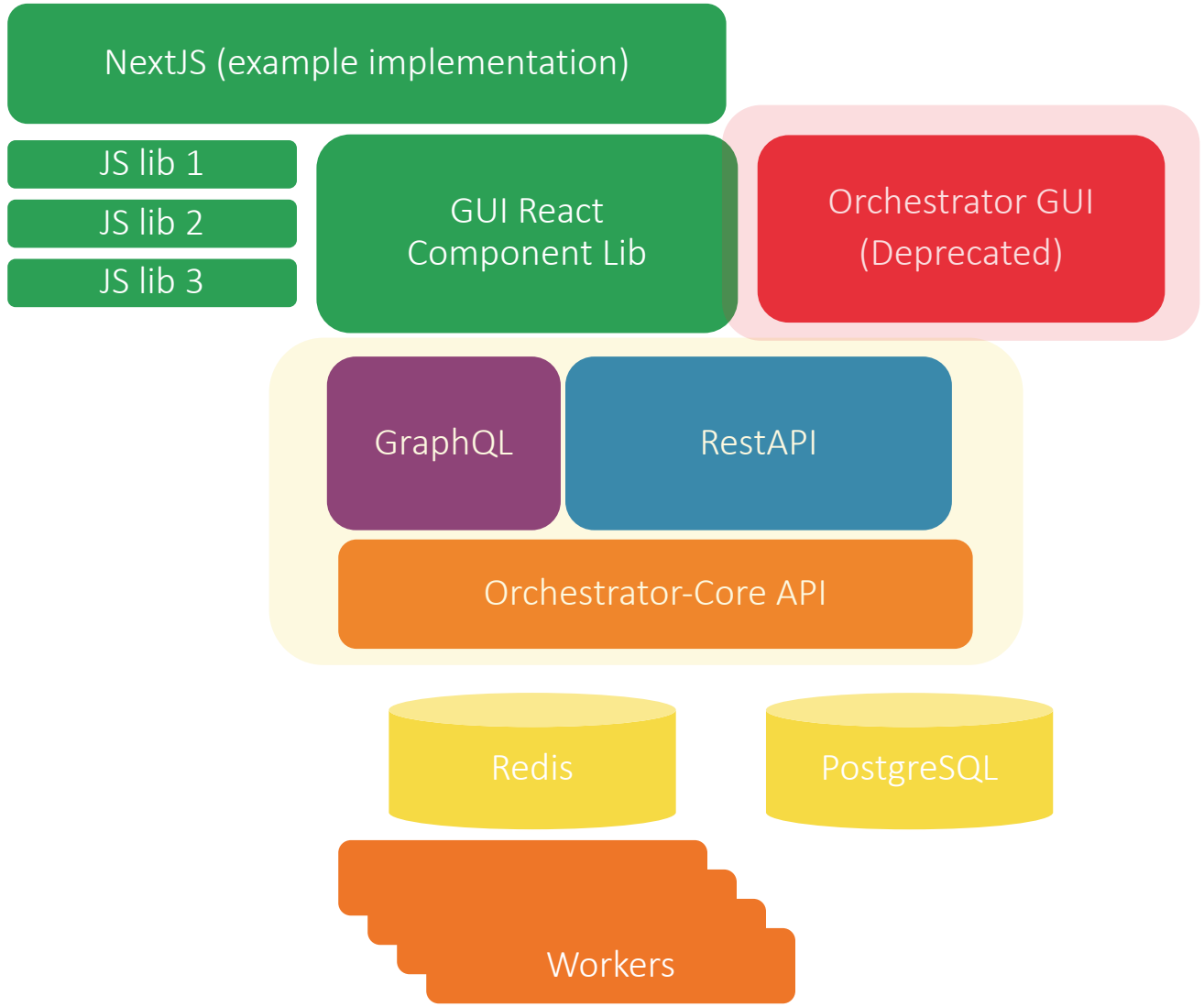
The orchestrator application architecture (basic)

- Python API based on FastAPI and Pydantic
- Rest
- GraphQL
- PostgreSQL database
- React application
- EUI components
- NextJS
- Uniforms



The orchestrator application architecture (at scale)

- Python API based on FastAPI and Pydantic
- Rest
- GraphQL
- **Celery**
- PostgreSQL database
- **Redis**
- React application
- EUI components
- NextJS
- Uniforms



https://edu.nl/uteaq