From Zero to Orchestrated

Peter Boers – SURF and Simone Spinelli - Geant

Agenda

Time	Activity
9:00 – 9:45	Introduction
9:45 – 10:00	History of the Project @SURF and @Geant
10:00 – 10:45	Workshop software overview and environment Setup
10:45 – 11:15	Break
11:14 - 11:45	Automation vs Orchestration
11:45 +/- 12:00	How do workflows work?
12:00 – 13:15	Lunch
13:15 – 14:30	Workshop exercises: Python, Docker & Containerlab
14:30 – 15:00	Break
15:00 – 16:15	Integration of OSS and BSS into your workflow
16:15 – 16:45	Where do you go from here?
16:45 – 17:00	Wrap-up

Who are we?

- Peter Boers: Software Architect, responsible for the Automation & Orchestration stack @SURF and Tech Lead of the workflow orchestrator program
- Simone Spinelli: Software Architect and responsible for the Automation & Orchestration stack at @Geant

Who are we? (2)

- Migiel de Vos: Team lead Network development @SURF, chair of the workflow orchestrator program
- Wouter Huisman: Business Developer and Product Owner of the workflow orchestrator @ SURF
- Arthur Nieuwland: Software Engineer/DevOps engineer @SURF and responsible for SURF network dashboard
- Karel van Klink: Software Engineer @Geant
- Mohammad Torkashvand: Software Engineer @Geant

Let’s get to know eachother (Menti)

https://www.menti.com/
Code: 5840 7513

Goals of this workshop

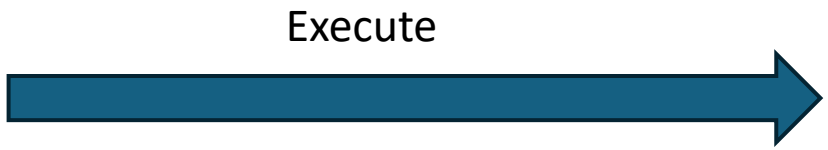
- The goal of this workshop is to have a reasonable understanding of the Workflow Orchestrator software, and how it can help you orchestrate service.
- Service modelling: Lines of configuration versus Services to end users
- A global understanding of the workflow engine
- Have an idea of how to integrate an OSS/BSS software in your business process and how you could define sources of truth
- Learn from other organisations and discuss Automation and Orchestration use cases

What do you get?

- A fully working setup including:
  - Docker
  - Orchestrator & UI
  - Netbox (Source of Truth)
  - LSO (lightweight service orchestrator)
  - Containerlab environment
- A network of like minded individuals
- … hopefully some inspiration on where to start/go on your orchestration journey.
- **Stickers!!!!!**

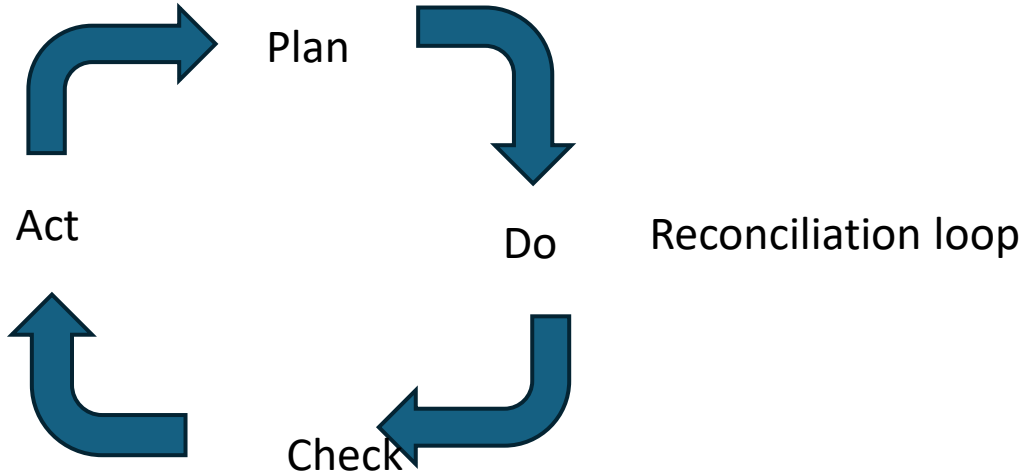
Automation:

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.



Orchestration concepts

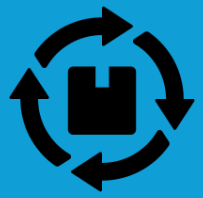
- Orchestration is executed on higher order abstractions
- The goal is to not only achieve valid network configuration, but to make it possible to define relations between all the data that is needed to provision an arbitrary service
  - Inventory
  - Customer data
  - IP address manangement
  - Ticketing
  - 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.

Why Automation & Orchestration?

Eliminate repetitive
& time consuming tasks

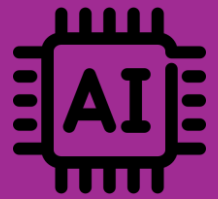
Prevent human
mistakes

Customer dashboard



Up-to-date
service lifecycle

Enable self service

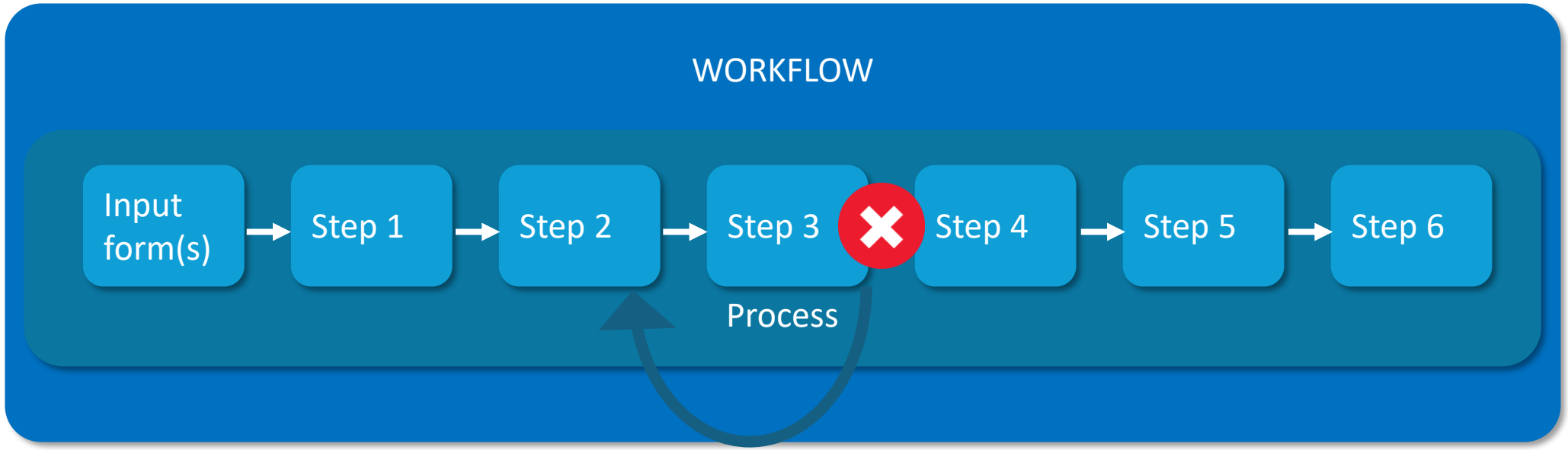


AI

What is the Workflow Orchestrator?

- A framework of tools developed by SURF, ESnet and Geant
- The orchestrator-core and UI provide a means for users to write workflows in Python that manage Services.
- 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. We call these processes.
- **The workflow engine orchestrates, automations.**

The Workflow Engine

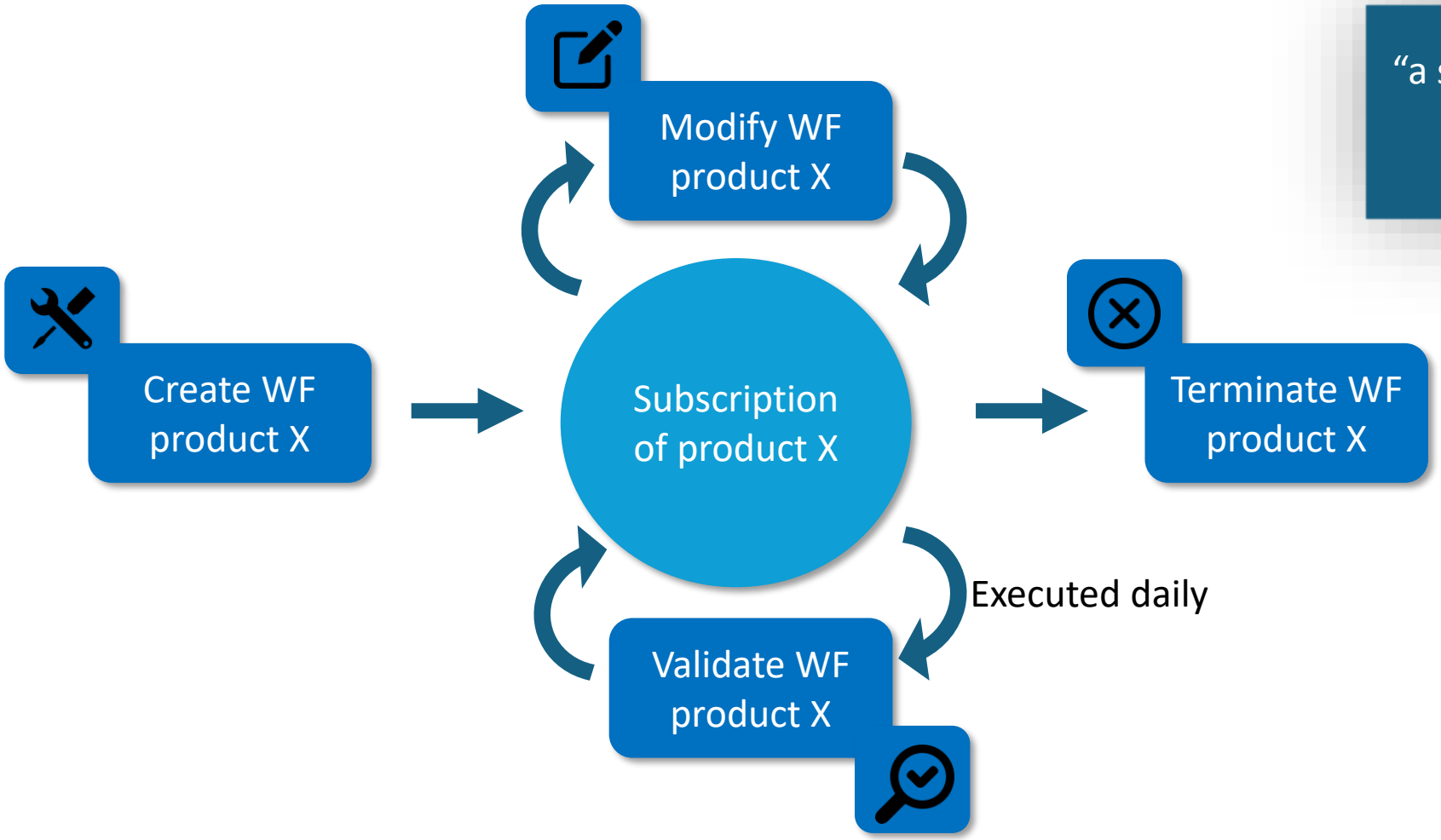


- Each Step writes the state to the database and is used as input for the next step
  - Each (atomic) Step can be retried, making the workflow robust

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
```

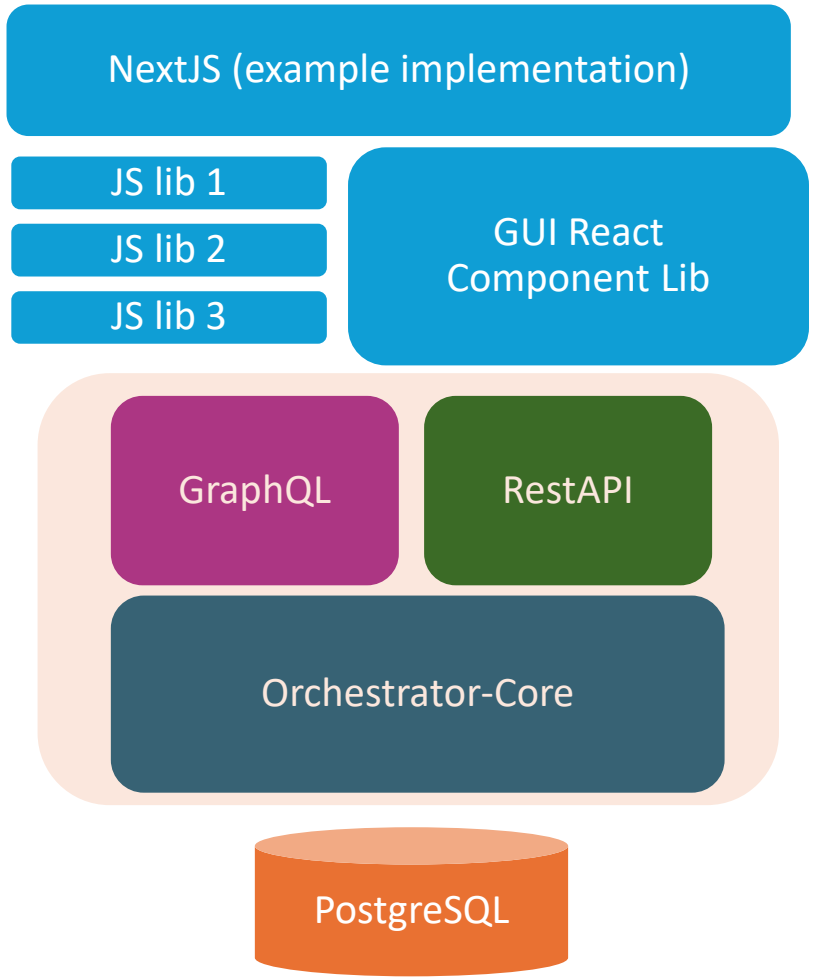
Lifecycle of a service



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

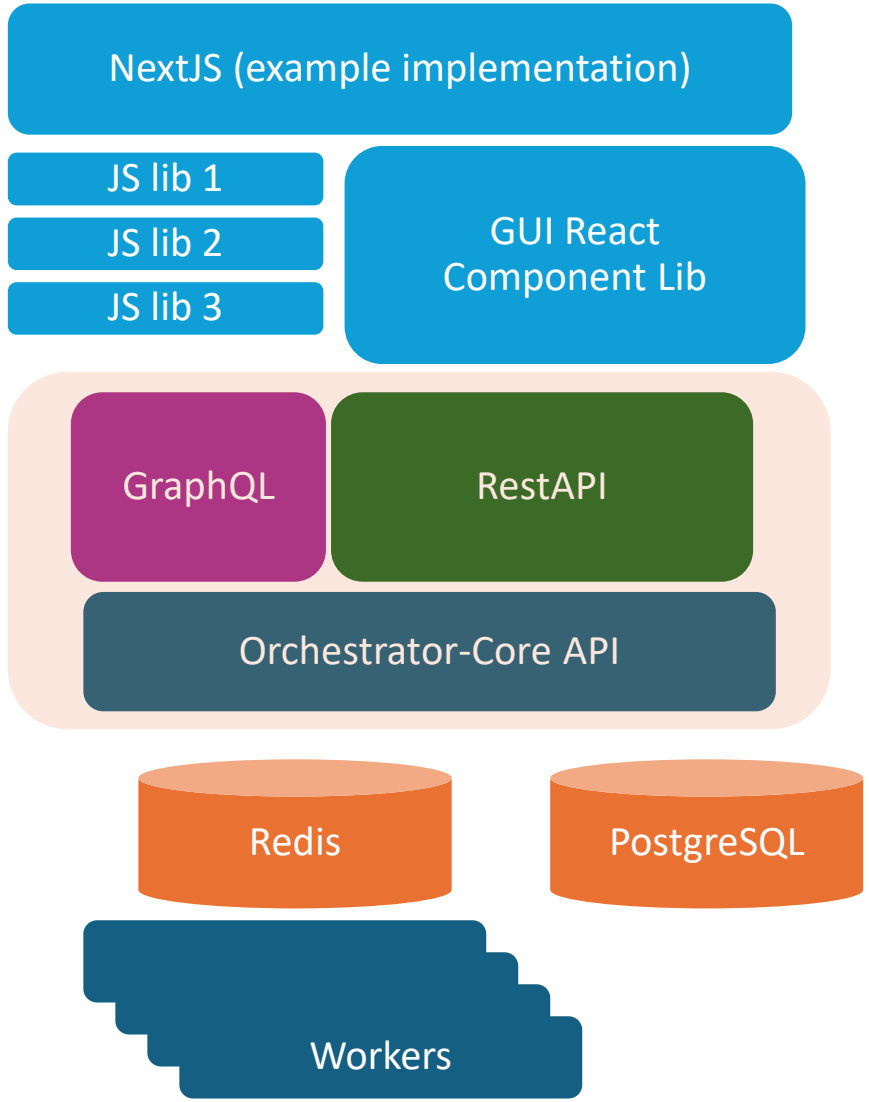
The orchestrator application architecture (basic)

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

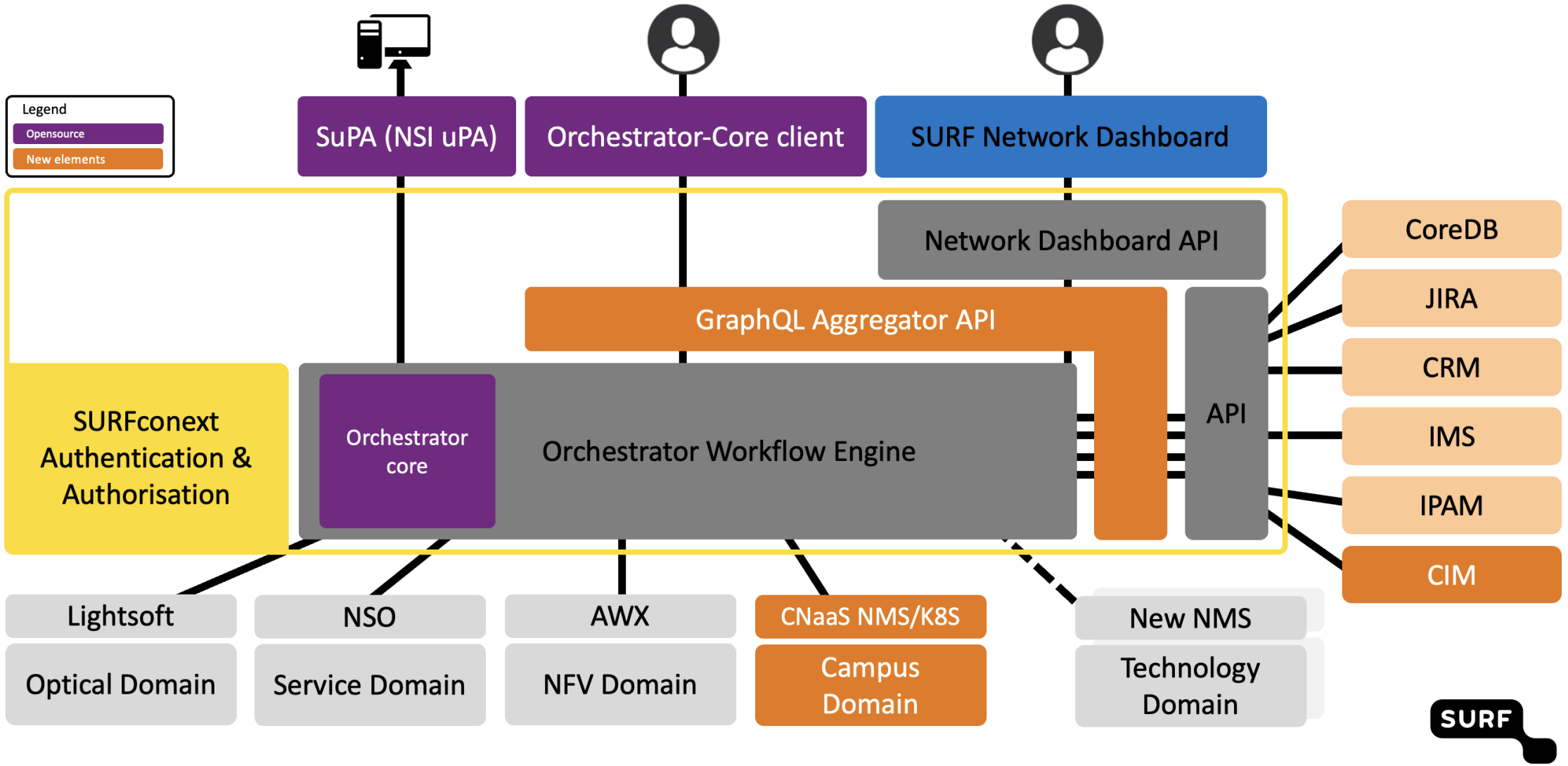


The orchestrator application architecture (at scale)

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



Example Software Architecture @SURF



DEMO

- Orchestrator UI
- Creating a Service
- Network Dashboard integration

Workflow Orchestrator vs Other workflow engines

- Orchestrator-core is a highly opionated framework
  - This enables you to focus on the heavy lifting: writing the integrations (90% of the work)
  - You only need to know Python
- Compared to other workflow engines like Camunda and Airflow it has less features
  - But it is also less complex. No DAG support ☺
- Built around the notion of a product portfolio and service offering, not about a business process
- Forces you to think about how users will interact with the system
- **In our experience, has enough features to get the job done**

History of the Orchestrator @SURF

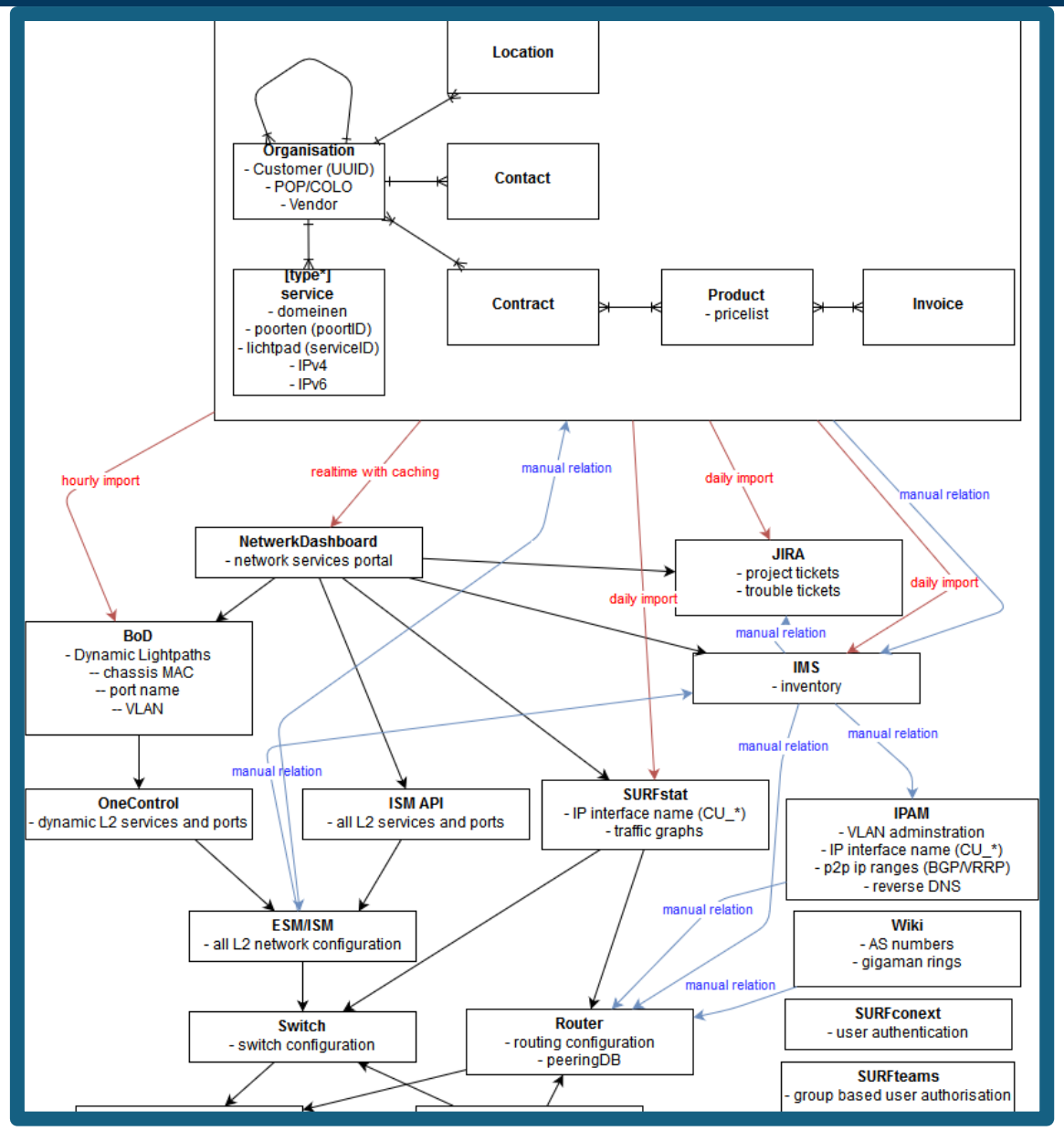
Peter Boers – SURF

What is SURF?

- SURF is the Dutch National Research and Education Network
  - Eduroam
  - Provides internet access to +/- 8% of the Dutch population daily
- Total network throughput, around 1 TB/s
- Connectivity services
- Security Services
- Procurment
- HPC

Humble Beginnings

- SURF's current automation journey started in 2016
  - Ambition was to fully automate the provisioning of SURFnet8
- The project started by evaluating workflow engines
  - Activiti
- Defining sources of Truth
- Reviewing the data architecture and processes
- "Defining our ideal business process"

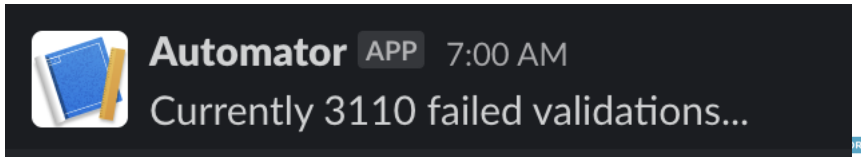

Location
Organisation
- Customer (UUID)
- POP/COLO
- Vendor
Contact
[type*]
service
- domeinen
- poorten (poortID)
- lichtpad (serviceID)
- IPv4
- IPv6
Contract
Product
- pricelist
Invoice
hourly import
realtime with caching
manual relation
daily import
manual relation
NetwerkDashboard
- network services portal
JIRA
- project tickets
- trouble tickets
daily import
daily import
manual relation
BoD
- Dynamic Lightpaths
-- chassis MAC
-- port name
-- VLAN
IMS
- inventory
manual relation
manual relation
manual relation
OneControl
- dynamic L2 services and ports
ISM API
- all L2 services and ports
SURFstat
- IP interface name (CU_*)
- traffic graphs
IPAM
- VLAN adminstration
- IP interface name (CU_*)
- p2p ip ranges (BGP/VRRP)
- reverse DNS
manual relation
ESM/ISM
- all L2 network configuration
Wiki
- AS numbers
- gigaman rings
manual relation
Switch
- switch configuration
Router
- routing configuration
- peeringDB
SURFconext
- user authentication
SURFteams
- group based user authorisation

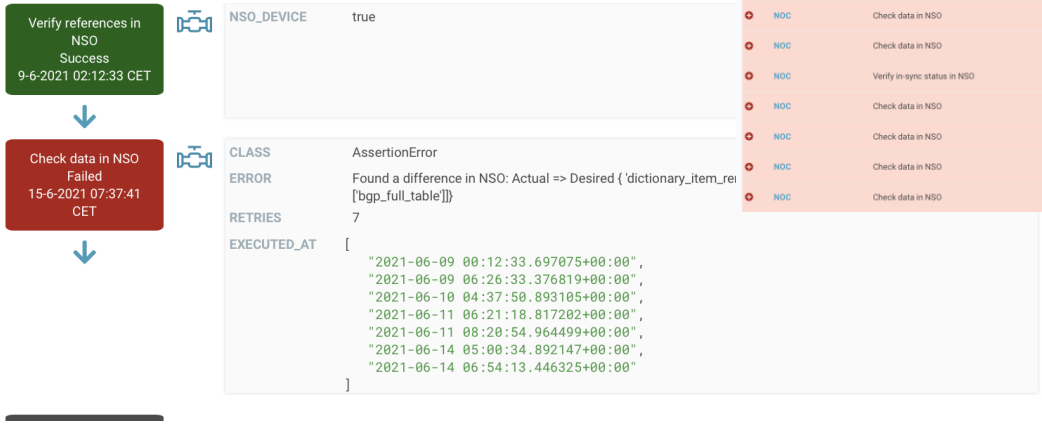

My head is exploding.

Throw everything away and start again

- First version of the Orchestrator was built as a PoC in one month at the end of 2017
  - At it’s core still part of the code-base today
- Many iterations later we went into production early 2018 by importing our previous network
  - Data cleanup and validation
- Re-implementing our business processes and products to better reflect reality
- Build our networkdashboard

Validations…..





Late 2018 - 2020 From legacy to service oriented abstractions

- The development pace was too slow:
  - Implementing a new product needed to come down from **6 months** to 6 weeks.
- Introduction of Domain models.
- From 90 products back to 25
  - Streamlining process and product modelling
- On critical path to out migration to SURFnet8
  - A.k.a critical stakeholder buy in
- Onboarding of ESnet

2020-now Iterating and growing

- Establishment of the Workflow Orchestrator program
- More and more complex products
- Cross-domain orchestration
  - Firewalls
  - Optical services
  - Wireless
- International Collaboration
- Autcon1 !

What did we learn?

- Most difficult part is stakeholder buy in
  - How do you convince people to use your system
- Closing the loop is very difficult
  - Validations are key
- Human error in creating configuration is reduced to 0%
- Reaching 100% coverage is difficult, keeping 100% coverage is easy
- Once everything is Orchestrated, you don’t want to do it any other way – you need to make it an integral part of service delivery, service design and customer experience
- The heavy lifting is done in the integration – 90-95% of the work.
  - It doesn’t matter what orchestrator you use –
    - But the workflow orchestrator is the best ;)