Deutsche Telekom AG
TNAP Portal
Portal Technology + Demo
Georg Schweflinghaus, Andreas Geißler
AGENDA

01 Intro
Why we do what we do?
Who we are?

02 Portal Development
How do we do it?

03 Demo
Where are we right now?

04 Issues
How to cope?

05 Outlook
What is planned next?
The conditions for our work are:

- Deutsche Telekom (DT) is trailing ORAN
- The first real world use case for the ONAP platform is managing the Open Radio Access Network (O-RAN).
- ONAP already features various web applications.

In this context the goal is to:

- fill gaps in the User Experience and functionality provided by standard ONAP components
- provide a twist towards radio engineering so that DT radio staff recognize their RAN domain, and we satisfy their requirements
- still remain generic to cater for new use cases while adapting the experience to the RAN domain
DT Portal team is one scrum team inside the TNAP project.

As of today:
- developers
- test automation engineers

Slovakia: Kosice
Germany: Münster, Bonn
Portal Focus:

- Role Based Access
- UI Application launch (ONAP + TNAP)
- Service Instantiation (VID replacement)
- Service Inventory (AII topology view)
- Service Monitoring (DCAE+)
- Business Mgmt UI (planned) (Customer, LoB, Project...)
- Service Infra UI (planned) (ESR UI replacement)
- Use case specific UIs
Portal Development - Context

Portal is:

- Accessible
- Secure
- Manage ORAN as first use case
- Leveraging ORAN promise to onboard several vendors
Layered architecture

Frontend
✓ Developed using Angular
✓ Provides Single Page Application
✓ Communicates with Backend via REST requests

Backend
✓ Developed using Java 11 + Spring Boot
✓ Microservice based
✓ Reduces complexity of southbound APIs for Frontend

Southbound
✓ ONAP services: SDC, A&AI, ...
✓ Infrastructure services: API Gateway, Authentication, Maps
02 Portal Development – Detailed View

Signle Page Application

Portal Frontend

[https]

Portal Backend

[Java11, Spring Boot]

Portal Preferences

[Java11, Spring Boot]

Portal Service

[Java11, Spring Boot]

NGINX

Reverse Proxy

KEYCLOAK

IAM System

Map Tile server
e.g. Open Streetmap

fetch instance data

fetch [data]

instantiate models

fetch [data]

fetch [data]

fetch [data]

fetch alarm data

fetch [data]

fetch [data]

fetch data

[REST]

fetch data

[REST]

fetch data

[REST]

provide config data

[REST]

provide app data

[REST]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]

fetch [data]
02 Portal Development - How we work!

Mocking APIs with WireMock - [http://wiremock.org/](http://wiremock.org/)

Generate clients, servers, and documentation from OpenAPI documents - [https://openapi-generator.tech](https://openapi-generator.tech)

Serenity [https://serenity-bdd.info/what-is-serenity/](https://serenity-bdd.info/what-is-serenity/)
End to End testing
Is achieved with a combination of serenity and Allure
03 Demo – The Real Thing

Functions shown in the demo

01 App Starter
02 Model Deployment
03 Instance View
04 Topology View
05 Alarm View
04 Issues

Size of ONAP deployment

- Parallel development with several teams requires multiple setups e.g. for independent automatic tests
- ONAP has extensive resource requirements and running several clusters is thereby expensive

Incomplete API descriptions

- APIs provided via „Open API File“ are partially not described, described wrong or even not existing
  - As one example, for SO we had to write/reverse engineer our own file to get instantiation working
  - Every cloud has a silver lining, the approach „Use the source Luke!“ was working for us.
04 Issues

Teach ONAP to NW operation or customize ONAP to mimic domain of NW operation

- Development teams face the challenge of mastering: SW Domain + ONAP Domain + RAN Domain
- Users are experts in their domain but usually not in the ONAP domain
05 Outlook – What is planned next?

“Just saying, just some keywords”

Specific for the portal:

- Day 2 Network Management e.g.:
  - Radio Site/Cell handling
  - RAN monitoring
  - RAN trouble shooting
- Use domain APIs that are built on top of NetConf
- Treeview for NW discovery
- Business context management (Customer, Project, LineOfBusiness,...)
- Service Infrastructure Management (Cloud region, ESR GUI replacement)

General:

- Midterm prepare for contribution of portal to ONAP
- ORAN a first domain handled as plug-in/module