



Rebuilding the Tower of Babel

A Choreography for the Management of Complex Solutions
Adrián Gallego Sánchez & Bernard Tsai - Deutsche Telekom



Breaking the tower: How did we get into this mess?

What we hoped for:



The promise of a brighter future via:

“Softwarization & Disaggregation of Network Functions”

Leading to:

- Flexible adaption to customer needs,
- more vendors,
- faster innovation,
- optimized resource utilization,
- energy-efficient components

What we experienced:



The harsh reality:

“Operators need to become Integrators as well”

And we are not good at that since:

- Integration is extremely hard,
- handling the dynamics of SW requires new operational procedures,
- resource/energy/cost consumption increases.

CSPs are in this domain (currently) too slow and inflexible, too ineffective and expensive and still lack stability.

What went wrong:



It's not a technology problem, but a:

“Lack of Common Understanding how to work together effectively”

Regarding how to:

- Plan network solutions,
- build network solutions,
- operate network solutions and,
- Configure then in a sustainable approach

Many proposals have been made to address these issues which do not converge, and legacy considerations still cloud our vision

CSPs, infrastructure and network function providers as well as potential integrators require a commonly accepted simplified understanding of how to work together



PLAN

Imagine we
had...



“Component Manifests”

which describe in a machine-readable format how to handle a component (container image + operator) and allows architects to design integrated network solutions containing

- the **schema for the sustainable configuration** of the component
- the list of **requirements towards other components** along with the schema for the configuration parameters
- the schema for **accessing the service endpoints** provided by the component
- the description of all **exposed service metrics** and their schemas provided by the component including those related to sustainability (sustainability by-design)

per default **provided by the vendors of the components.**



manifest.yaml

Metadata:

gvk: <group/version/kind>

Configuration:

schema: <JSON Schema for Spec section>

Requirements:

- **name:** <dependency name>
- gvk:** <group/version/kind of dependency>
- schema:** <JSON Schema for spec section>
- [...]

Endpoints:

- **name:** <endpoint name>
- protocol:** <protocol name>
- urlschema:** <schema for endpoint url>
- [...]

Metrics:

- **name:** <endpoint name>
- protocol:** <protocol name>
- urlschema:** <schema for metric url>
- [...]

Exemplary

Eco-
score



PLAN

How to specify cloud infrastructure requirements?



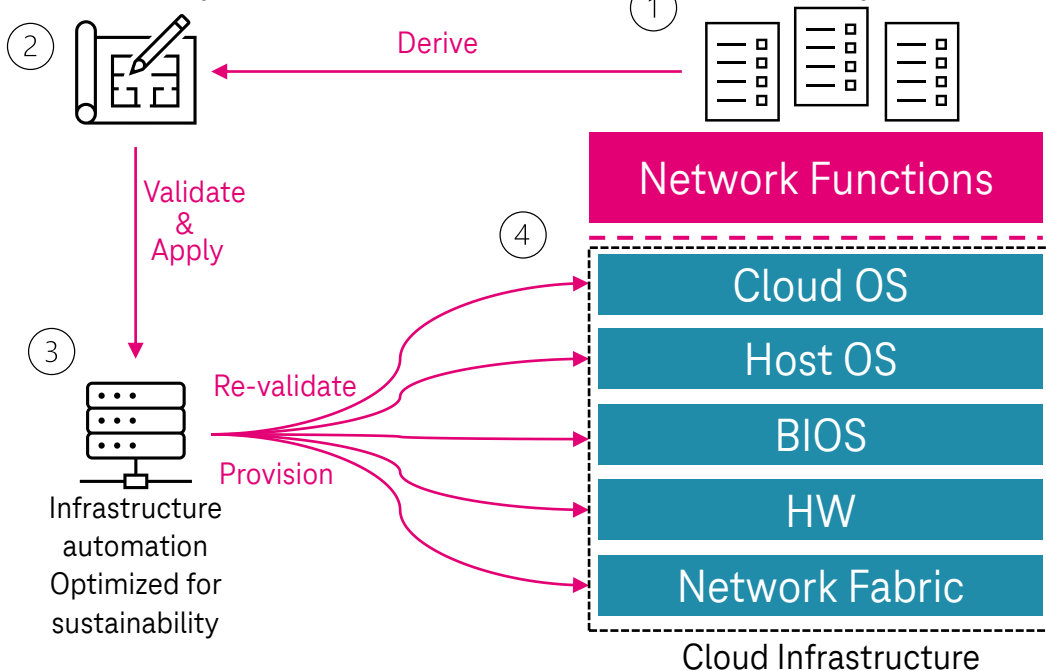
Imagine we
had...

*“Common Infrastructure
Configuration Templates”*

that would allow to capture infrastructure requirements of network functions and automate the provisioning of a suitable cloud infrastructure.

infrastructure.yaml

manifest.yaml



infrastructure.yaml

Exemplary

```
Metadata:
  cluster:<cluster>
  version:<version>

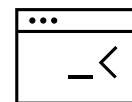
Network:
  [network specifications]

Hardware:
  [hardware specifications]

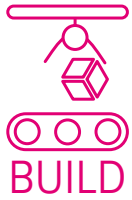
BIOS:
  [BIOS specifications]

HostOS:
  [Host OS specifications]

CloudOS:
  [Cloud OS specifications]
```



& **infrastructure validation functions**



How to continuously integrate & deploy solutions?



Imagine we
had a...

Common Best-Practice CI/CD Process

(the bread & butter tasks handled by an integrator)



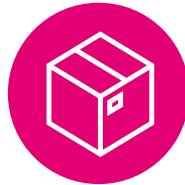
Sourcing

Retrieval of software
from suppliers



Configuration

Definition of target
state of a solution



Integration

Construction of
release bundles



Distribution

Distribution of
release bundles



Deployment

Rollout of releases
into runtime env.



Validation

Functional and non-
functional tests

where we would need to give clear guidance and request support from our vendors

- | | | | | | |
|-----------------------|-----------------------|--------------------|--------------------|-------------------|-------------------|
| ▪ simple replication, | ▪ config. schemas, | ▪ simple to-use | ▪ support of | ▪ non-proprietary | ▪ pluggable |
| ▪ signed releases, | ▪ if really needed, | packaging format, | distributed | and easy to debug | validation tools, |
| ▪ sem. versioning, | pluggable config | ▪ optimized CI | secrets | deployment | ▪ support for |
| ▪ regular updates of: | tools, | pipelines, | management, | mechanisms, | debugging, |
| ▪ manifest files | ▪ dry-run capability, | ▪ usage of energy- | ▪ delta updates, | ▪ simplified & | tooling for |
| ▪ images | ▪ defaulting to | aware testing | ▪ minimizing | harmonized | sustainability |
| ▪ operators | energy-efficient | environments | artifact | lifecycle | metrics, |
| ▪ config tools | settings | | duplication across | management with | ▪ power usage |
| ▪ validation tools | | | stakeholders | sustainable ops. | checks |



How can we continuously change without disruption?

Imagine we had...

“Harmonized Lifecycle Management”

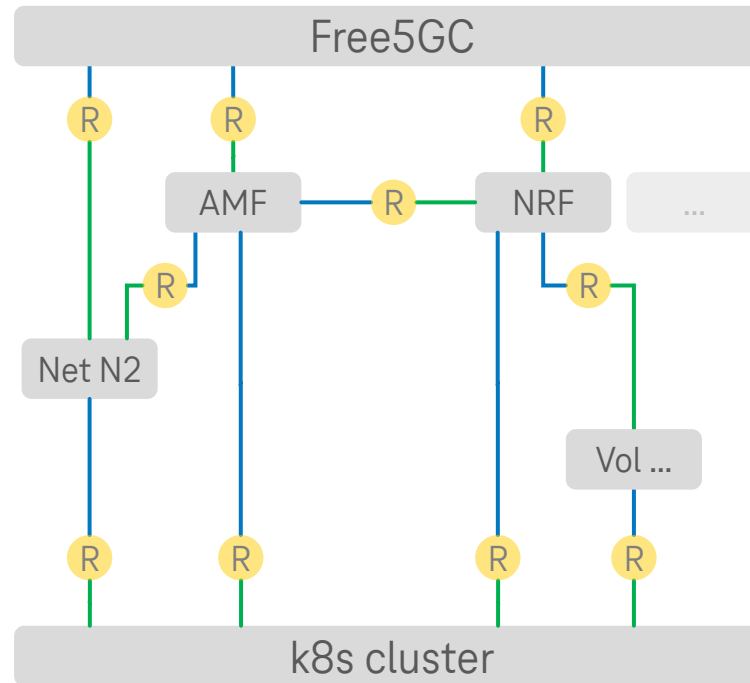
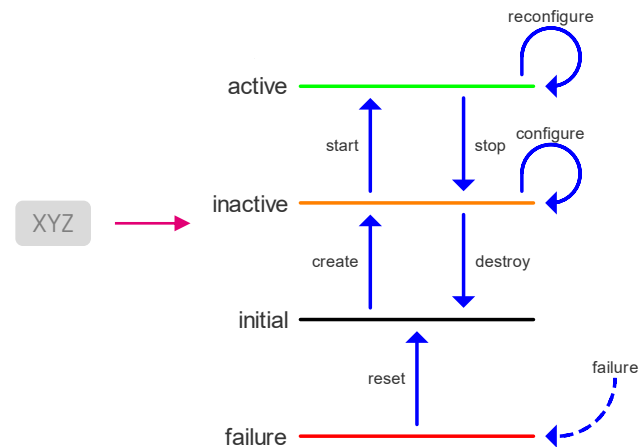
of

Network Solutions

(as example a subset of Free 5GC)

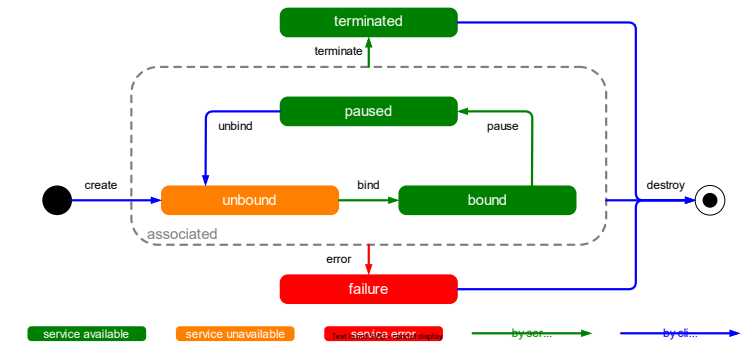
Components XYZ

CSPs make use of a generic high-level lifecycle state model to manage the components of network solutions



Relationships R

CSPs use in the same manner a generic high-level lifecycle state model to signal the status of the relationships between components



XYZ component R relationship dependency service

Converging towards a common understanding

The Community

We propose to establish a **community of communication service providers**.

 Their mission is to jointly develop and maintain an evolving **canonical catalogue of recommended best practices** for the management of disaggregated network functions.

The Proposal
 Consolidate,
 Simplify & Refine!

The Methodology

1. Workstreams on priority topics, e.g.
 - component manifests
 - infrastructure templates
 - validation functions
 - CI/CD process
 - harmonized lifecycle management
 - ...
2. Evaluation of the outcome of existing community initiatives & SDOs and selection of recommended best-practices
3. Refinement of the best-practices into results documents/procedures used in real-life implementations

The Results

Each workstream defines the expected results to be used in the actual operational context of the CSPs, e.g.

- Requirements for RFQs
- Guidelines for Configuration
- Guidelines for sustainable NFs design
- Guidelines for CI/CD Procedures
- Validation Procedures
- Measurable Sustainability KPIs

And provides feedback towards vendors, community initiatives and SDOs as well as Q&A support.



Q&A



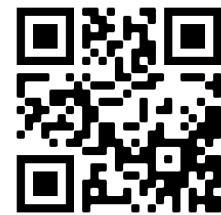
Contacts



Deutsche Telekom
5G and Cloud Architect
Adrian Gallego Sanchez



Deutsche Telekom
Lead Architect
Bernard Tsai



Disclaimer



Co-funded by
the European Union

6G SNS

This work is Co-funded by the European Union under Grant Agreement 101191936. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of all SUSTAIN-6G consortium parties nor those of the European Union or the SNS JU (granting authority). Neither the European Union nor the granting authority can be held responsible for them.

