5G INDUCE

Open cooperative 5G experimentation platforms for the industrial sector NetApps

www.5G-induce.eu

Infocom World Conference & Exhibition 2022
Research-oriented Workshop:
"Research Projects for creating the Future and Innovative Telecoms Market"

November 29th, 2022

5G-INDUCE Platform Design for Industrial Sector Network Applications

Chiara Lombardo
CNIT

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No. 952672

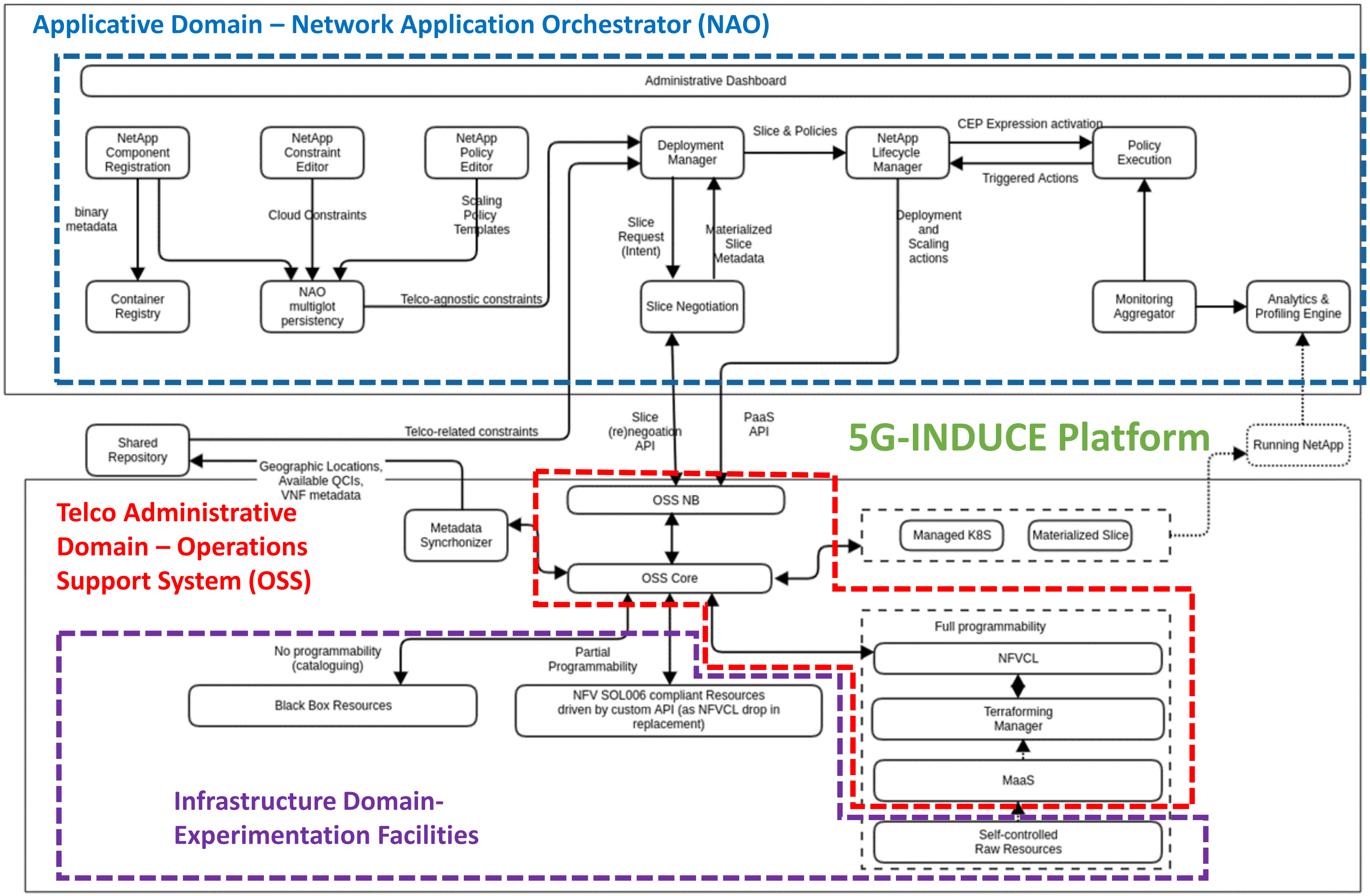
The 5G-INDUCE Project

- The 5G-INDUCE project aims to build open and cooperative 5G network platforms for the showcasing and evaluation of advanced network applications supporting innovative services related to the Industry 4.0 context.
- **Goal:** provide realistic experimentation facilities for the seamless deployment of network functions, forming the building blocks of market oriented industrial applications, while providing an attractive platform for service providers.

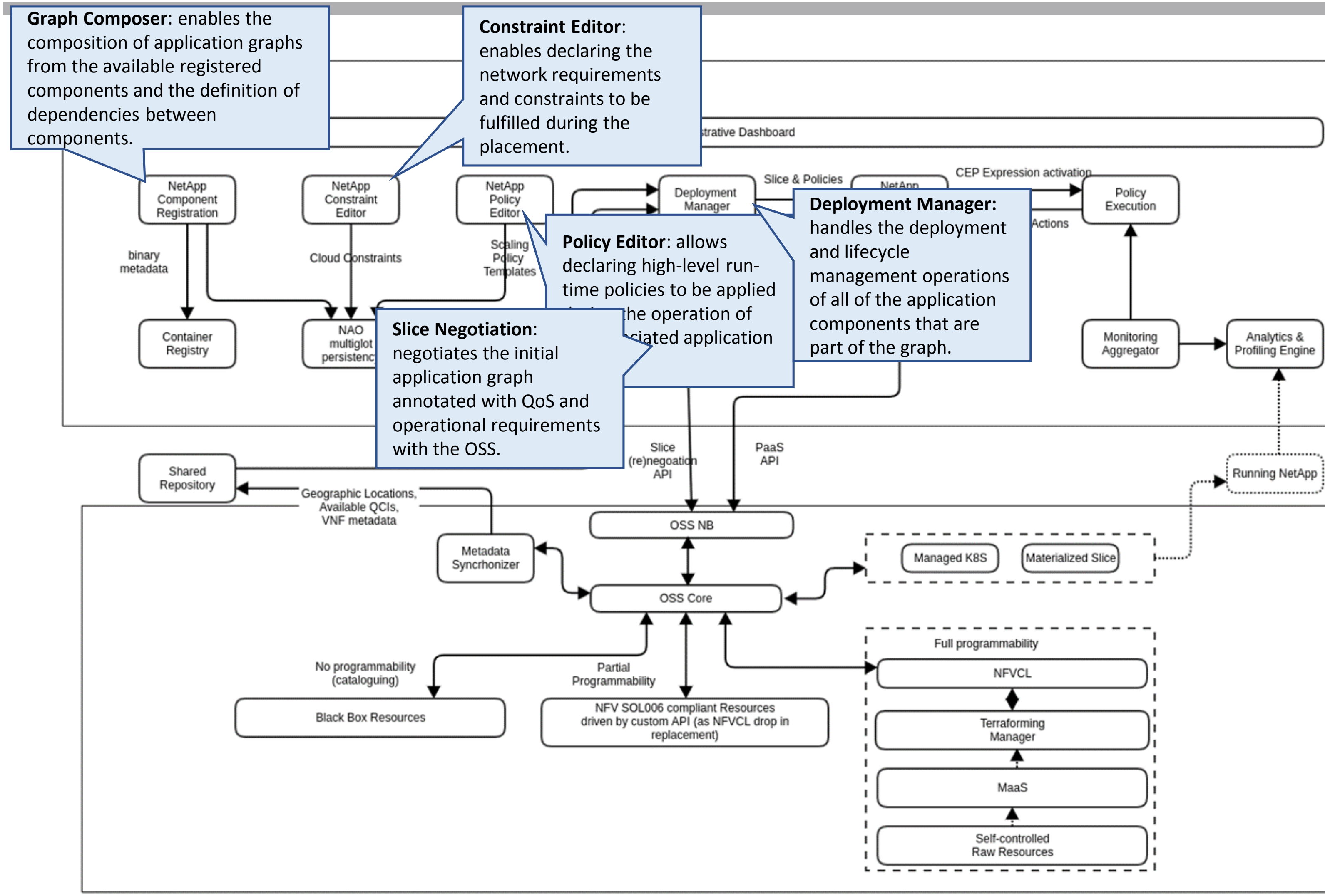
The 5G-INDUCE Platform

- To this end, 5G-INDUCE aims to deliver a holistic platform to manage applications on top of programmable 5G slices, and to bridge the gap between the application and the 5G network domains, by
  - Enabling vertical application developers and service providers to design and deploy 5G-ready applications, by integrating common microservices design patterns with requirements that drive the following deployment over the 5G infrastructure
  - Supporting the smooth deployment of the 5G-ready application over the 5G infrastructure, including the automatic provision of the supporting 5G network slice.
- The proposed platform aims to hide the complexity of the 5G environment to application developers and providers and make the development, deployment and operation of applications similar to the well-known processes in cloud computing environments.

5G-INDUCE Platform Architecture

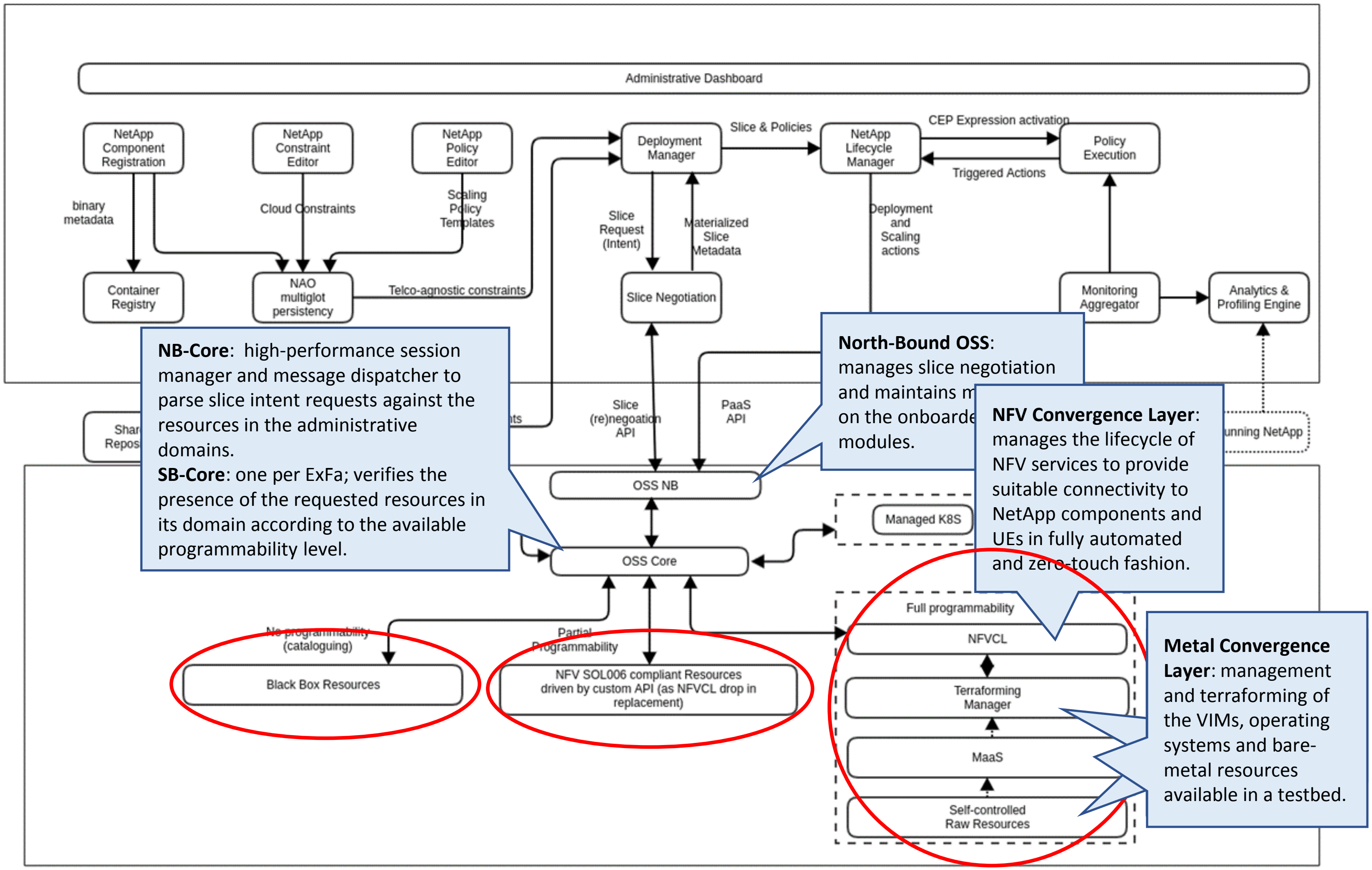


The NAO



Graph Composer: enables the composition of application graphs from the available registered components and the definition of dependencies between components.
Constraint Editor: enables declaring the network requirements and constraints to be fulfilled during the placement.
Policy Editor: allows declaring high-level run-time policies to be applied the operation of ciated application
Deployment Manager: handles the deployment and lifecycle management operations of all of the application components that are part of the graph.
Slice Negotiation: negotiates the initial application graph annotated with QoS and operational requirements with the OSS.
strative Dashboard
NetApp Component Registration
NetApp Constraint Editor
NetApp Policy Editor
Deployment Manager
Slice & Policies
NetApp
CEP Expression activation
Policy Execution
Actions
binary metadata
Cloud Constraints
Scaling Policy Templates
Container Registry
NAO multiglot persistenc
Monitoring Aggregator
Analytics & Profiling Engine
Shared Repository
Geographic Locations, Available QCIs, VNF metadata
Slice (re)negoatiation API
PaaS API
Running NetApp
OSS NB
Metadata Syncrhonizer
Managed K8S
Materialized Slice
OSS Core
Full programmability
NFVCL
No programmability (cataloguing)
Partial Programmability
Black Box Resources
NFV SOL006 compliant Resources driven by custom API (as NFVCL drop in replacement)
Terraforming Manager
MaaS
Self-controlled Raw Resources

The OSS



NB-Core: high-performance session manager and message dispatcher to parse slice intent requests against the resources in the administrative domains.
SB-Core: one per ExFa; verifies the presence of the requested resources in its domain according to the available programmability level.

North-Bound OSS: manages slice negotiation and maintains m… on the onboarde… modules.

NFV Convergence Layer: manages the lifecycle of NFV services to provide suitable connectivity to NetApp components and UEs in fully automated and zero-touch fashion.

Metal Convergence Layer: management and terraforming of the VIMs, operating systems and bare-metal resources available in a testbed.

Slice Intent Request and Materialization (1)

1. The NAO sends the NB-OSS a request that includes all the computing resources, networking services and constraints that are necessary to run application components (Slice Intent).
2. Upon reception of the slice intent, the NB-Core parses it against the resources in the administrative domains made available by registered SB-Cores; the first SB-OSS confirming the feasibility of the slice is chosen.
3. The NB-OSS returns to the NAO the candidate slice materialization.
4. Upon positive confirmation from the NAO, the NB-Core asks the selected SB-Core to materialize the slice and returns back a notification when the environment is ready.

Slice Intent Request and Materialization (2)

5. Depending on the programmability level available in the infrastructure, there are three options:
   a) non-programmable domains: the SB-Core maintains a catalogue of the pre-allocated slices and performs an admission control for slice intent requests. If the slice intent requirements fit, it will assign and reserve the resources.
   b) configurable domains: programmability is handled using ETSI NFV-SOL 006 as reference API to represent the NSDs, the onboarding of NSDs being the positive feedback requested to guarantee the feasibility of the slice-intent.
   c) programmable domains: from bare-metal to IaaS and PaaS-level programmability, it entails the presence of the NFVCL and the MetalCL at varying degrees.
6. For c), upon request from the OSS-SB Core, the NFVCL selects the most suitable blueprint (e.g., a generalized structure realizing complete network environments) and provides feedbacks to the SB-Core when all the VNFs and PNFs have been successfully configured.
7. When the required slice has been successfully created, a new slice ID is returned to the NAO in the form of a candidate materialized slice.

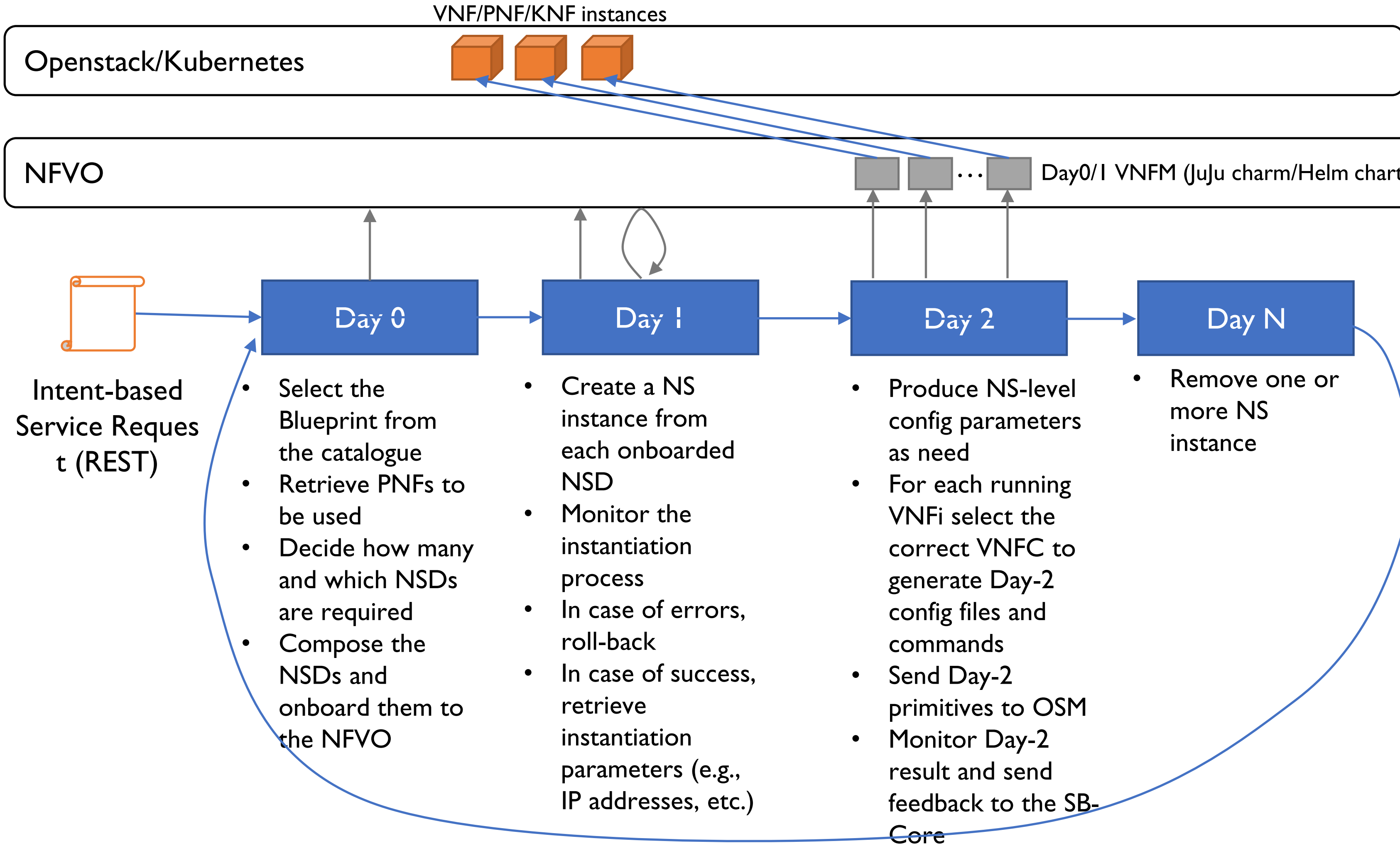
The NS Blueprint

- The NS Descriptor (NSD) specified by ETSI NFV is composed of a pre-determined, unmodifiable number of different VNFs and links.
  - No standard VNF Manager, only a standard “container for VNFM” (i.e., Juju)
- Network service blueprint: a new, generalized structure can be seen as an LCM manager of a coordinated set of NFV NSs to realize a comprehensive network service (e.g., a radio-mobile network, a VoIP system, etc.):
  - Day 0: terraforming VIMs with needed resources, types of PNFs/VNFs/KNFs, their inter-connections, and the virtual networks to be used towards the outside.
  - Day 1/2: run-time information collection (e.g., dynamic IP addresses, KPIs, etc.), configuration files and commands (both as templates filled by run-time data) to run on SW processes inside PNFs/VNFs/KNFs.
  - Day N: cleaning resources and instances (even in a part of NSs within the blueprint).

Lifecycle Management

- Day-2 operations can be triggered by the NAO (e.g., upon changes in the SLA) or by health checks on the resources.
- A monitoring framework triggered by the NFVCL allows retrieving metrics from Kubernetes and OpenStack and providing them to external applications.
- Such applications can use them to perform analytics and request the NFVCL to perform scaling/maintenance operations at runtime.

Lifecycle Management Workflow



THANK YOU!

Chiara Lombardo

CNIT

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No. 952672