Day-0, day-1 and day-2 configuration in OSM

Gerardo García (Telefónica)
Traditional network management

**Day 0**
- PNF installation
- Initial configuration to make PNF reachable (user, pwd, network, etc.)

**Day 1**
- License activation
- Injection of configuration
- Neighbor configuration
- Network configuration

**Day 2**
- Service provisioning
- Business provisioning
- BSS

© ETSI 2018
NFV management

Day 0
- VNF deployment
- NS deployment (complex topology)

Day 1
- OSS
- EMS
- VNF configuration
- License activation
- Neighbor configuration
- Network Service
- MANO

Day 2
- OSS
- EMS
- MANO
- Network Service
- VNF
- Service provisioning
- Business provisioning
- BSS
VNF’s operational procedures are embedded in the VNF Package...

**VNF package**
- VNFD
- VNF artifacts
- Additional metadata?

**Capacity aspects**
- VNF resource orchestration info (EPA resources and internal connectivity)

**Functionality aspects**
- Descriptive information
  - metadata.yaml
  - config.yaml
  - actions.yaml
- Executables
  - Hooks
  - Actions
- Additional info (icon, README)

© ETSI 2018
... ready to be used in complex Network Services, which define their own E2E operational procedures...
VNF operational procedures are encapsulated in charms, which are controlled by VCA.
Juju charms in layman’s terms
What is a charm?

- A charm is a set of actions and hooks
  - Actions are programs
  - Hooks are events/signals
- For commodity and reusability, those actions and hooks are grouped in layers
- A charm will always have one layer:
  - That layer has some actions and hooks
  - In addition, that layer can import other layers
- The resulting charm has all the actions and hooks from all the layers joined together, plus additional default actions and hooks (e.g. ‘config’ action)
What is a charm?
Charm design

layers

hooks

actions

red

blue

green

‘config’ action
What is a charm?
Charm build
From charms to VNF primitives
VNF descriptors map charm actions to VNF primitives, thus providing a full set of enablers for NFV management.

**OSM NB API:**
- Create new NS instance
- Get NS instance record
- Get VNF metrics
- Call NS primitive
- Scale in/out NS instance

**VNFD:**
- Basic configuration (SSH keys, hostname, user-data scripts, etc.)
- Initial config primitive
- Config primitive
- Charms ➔ Actions
- Charms ➔ Metrics

**NSD:**
- Initial config primitive
- Service primitive
- Scaling groups
- Pre and post scaling primitives
- Scripts
Day-0 configuration (at instantiation time)

OSM NB API:
- Create new NS instance

VNFD:
- Basic configuration (SSH keys, hostname, user-data scripts, etc.)

UI → OSS → SO → VCA → RO

Injection of basic configuration
Day-1 configuration (at instantiation time)

OSM NB API:
- Create new NS instance

VNFD:
- Initial-config-primitive
- Charms -> Actions

NSD:
- Initial config primitive
- Scripts
Day-2 configuration

- OSM NB API:
  - Get NS instance record
  - Call NS or VNF primitive

- VNFD:
  - Config primitive
  - Charms -> Actions

- NSD:
  - Service primitive
  - Scripts
Day-2 elasticity (scaling)

OSM NB API:
- Get NS instance record
- Scale in/out NS instance

VNFD:
- Basic configuration (SSH keys, hostname, cloud-init scripts, etc.)
- Charms → Actions

NSD:
- Scaling groups
- Pre and post scaling primitives