Cloud-Native VNF Modelling

Juju Charms as a Generic VNFM in OSM
In the rest of the world...

- Physical Machines
- Virtual Machines
- Cloud Native Machines
- Container Machines
- Process Containers
- Serverless

Image-based
NFV is here

© ETSI 2016
This stuff is still crazy even on AWS

Juju gets you to here

Physical Machines
Virtual Machines
Cloud Native Machines
Container Machines
Process Containers
Serverless
Bridging the old and the new

NSO - RIFT.io

RO - OpenMANO

“Stand up a service”

“Configure service”

“Stand up an image”

VNF Image

VIM
Bridging the old and the new

- Old Appliance
- VNF Image
- VNF Charm
- VCA - Juju
- Instantiate Ops Code
- VIM
- NSO - RIFT.io
- RO - OpenMANO

- “Stand up a service”
- “Configure service”
- “Stand up an image”
Charms are **shared operations code**

- VNF lifecycle - install, upgrade, remove
- Scale - HA, throughput
- Integration
- Operations - backup, workflows, monitoring

Shared operations code **reduces the time** to onboard and to train engineers on a new VNF
Charms are cloud-native model-driven ops

- VNF lifecycle - install, upgrade, remove
- Scale - HA, throughput
- Integration
- Operations - backup, workflows, monitoring

New generation of VNFs are scale-out and cloud-native
Model-driven operations

This is how you will actually take control of performance and SLAs.
What’s in a model?

“sgw app”

“fabric app”

“mme app”

“hss app”

“pcrf app”
Looking forward

VIM

VNF Charm

Specific VNFM

Instantiate Ops Code

"Stand up a service"

"Configure service"

RO - OpenMANO

NSO - RIFT.io

"Stand up an image"

VCA - Juju

"Configure service"

VNF Charm

© ETSI 2016
“Application” is a logical concept that spans across all the machines where that code is running with a common config, as a scale-out deployment.

These two machines are each hosting part of the application. There are two units in this app. Each unit has the same charm installed. The charm encodes all operations for that app.
Charm Structure

- Lifecycle - install, upgrade
- Scale - horizontal elasticity
- Integration
- Configuration

./metadata.yaml
actions.yaml
config.yaml
README.md
copyright
actions/
hooks/
name: oai-spgw
summary: OpenAirInterface EPC
maintainers:
  - Navid Nikaein <navid.nikaein@eurecom.fr>
  - Andrea Bordone Molini <bordone@eurecom.fr>
description: 4G-5G OpenAirInterface EPC on any cloud infrastructure.
tag: ['4G-5G', 'EPC', 'OpenAirInterface']
provides:
  spgw:
    interface: S11
requires:
  hss:
    interface: S6a-hss

metadata.yaml

A
  "provides S11"
  spgw

B
  "consumes S11"
  mme

S11 interface
Lifecycle Hooks

Install
Scale
Update

Charm Structure

./hooks/
install
start
stop
leader-elected
upgrade

1. install
2. leader-elected
3. config
4. start

1. install
2. config
3. start
4. peer-relation-joined
Charm Structure

Integration Hooks

Relations and Interfaces
Peers

./hooks/
  X-relation-joined
  X-relation-changed
  X-relation-broken
  X-relation-departed

peer-relation-*
Charm Structure

Integration Hooks

Relations and Interfaces
Peers
Charm Structure

Integration Hooks

Relations and Interfaces
Peers
Charm Structure

Operations Hooks

backup-database
  description: Backup MySQL DB
  params:
    db-name:
      type: string
      default:
        description: DB to be backed up
  filename:
    required: [db-name]

./actions/
  backup-database
  vacuum-database

./metadata.yaml
actions.yaml
config.yaml
README.md
copyright
actions/
hooks/