OSM#9 Hackfest
Introduction to OSM Primitives
David Garcia, Mark Beierl, Dominik Fleischmann (Canonical)
Reality is messy and mixed
One VNF is many apps and integration
VNFD

“Charms” - operations code
- Lifecycle
- Configuration
- Integration
- Actions / Primitives

Metadata

DECLARATIVE

CODE
“Charms” - operations code
- Lifecycle
- Configuration
- Integration
- Actions / Primitives
OSM “Primitives” are mapped to Charm Actions
Operating “proxy” workloads

‘Operator’ instance

Proxy Charm

OSM

Workload
PNF/VNF/KNF

Where can we run our own operations code for this workload?
Operating “native” workloads

Workloads optimised for OSM have a charm that drives the workload directly.
Integration is first-class in the VNFD

Lines of integration in the VNFD
Charms declare **typed integration points**

“I can use a MySQL database”

“I can use LDAP”

“I can send my logs to a syslog”
Matching integration points can be related

Lines of integration between matching integration points on different charms
Composition gives complex integrations
VNFds can describe complex integrations

Lifecycle scripts
Config scripts
Integration scripts
Action scripts
OSM primitives are Charm Action scripts

- Backup
- Monitor
- Debug
- Add users, policies, rules, etc.
- Manage certificates, keys, etc.
- Rotate logs

Each ‘primitive’ is a charm action script that takes parameters and produces output.
Charms are packages of scripts to drive apps

Lifecycle scripts
- install
- config
- update
- remove
- scale

“Action” scripts are OSM Primitives
- “action: backup”
- “action: restore”
- “action: scan-viruses”
- “action: health-check”
- “action: add-repo”
- “action: ...”
- “action: ...”
- “action: ...”

Integration scripts
- relate-mysql
- relate-ldap
- relate-proxy
- relate-...

These are your operations primitives.
Charm describes Action parameters

Charm metadata describes the action parameters.

Each Action is a script, usually in Python or Bash.

```json
addurl:
  description: "Add squid config"
  params:
    url:
      description: "URL that will be allowed"
      type: string
      default: ""

deleteurl:
  description: "Delete allowed URL squid config"
  params:
    url:
      description: "URL that will stop to be allowed"
      type: string
      default: ""
```
Charm Action script in bash

Actions can be written in bash for very simple cases.

```bash
#!/bin/bash

URL=`action-get url`

if ! grep -Fxq "http_access allow allowedurls" /etc/squid/squid.conf
then
    sed -i '/^# And finally deny all .*/i http_access allow allowedurls\n' /etc/squid/squid.conf
fi

sed -i "/^http_access allow allowedurls.*\)/ etc/squid/squid.conf dstdomain \.$URL" /etc/squid/squid.conf

kill -HUP `cat /var/run/squid.pid`
```
def on_deleteurl_action(self, event):
    """Handle the deleteurl action."""
    url = event.params["url"]

    line_to_delete = "acl allowedurls dstdomain .{}".format(url)
    line_deleted = False

    with open("/etc/squid/squid.conf", "r") as f:
        lines = f.readlines()
    with open("/etc/squid/squid.conf", "w") as f:
        for line in lines:
            if line_to_delete not in line:
                f.write(line)
            else:
                line_deleted = True

    if line_deleted:
        event.set_results({"output": "URL deleted succesfully"})
        subprocess.check_output("kill -HUP `cat /var/run/squid.pid`", shell=True)
    else:
        event.fail("No URL was deleted")
The Juju Controller

Introduction to VCA
Juju drives application operations on machine and Kubernetes substrates.

Install, update, configure, scale, integrate, and actions.
Juju architecture

- Machine
- Charm
- Application model
- Juju Controller
- Juju client
- LCM
- VCA
- Juju agent
- Workload PNF/VNF/KNF

© ETSI 2020
Juju controller manages multiple models
VCA uses multiple models for scenario

Juju client

Juju Controller

Model
K8s cloud

Model
VIM cloud

LCM

VCA

KNF

KNF

Kubernetes

VIM

VNF

VNF

VNF

© ETSI 2020
VCA can be high availability
VCA coordinates all OSM Primitives
Configure

https_proxy: xxx
ca_cert: yyy
...
Actions

Do backup!

LCM → VCA

Charm

Workload
PNF/VNF/KNF

© ETSI 2020
Integration

Integrate!

LCM → VCA

PNF, VNF, KNF

Charm

© ETSI 2020
Reality is messy and mixed

LXD Operators
Proxy Charm
Ops Code

PNF PNF

VNF VNF

OSM

Native Charm
VNF

Native Charm

PNF PNF

VNF VNF

KNF KNF

Racks

VIM

Kubernetes

K8s Operators
Proxy Charm
Ops Code