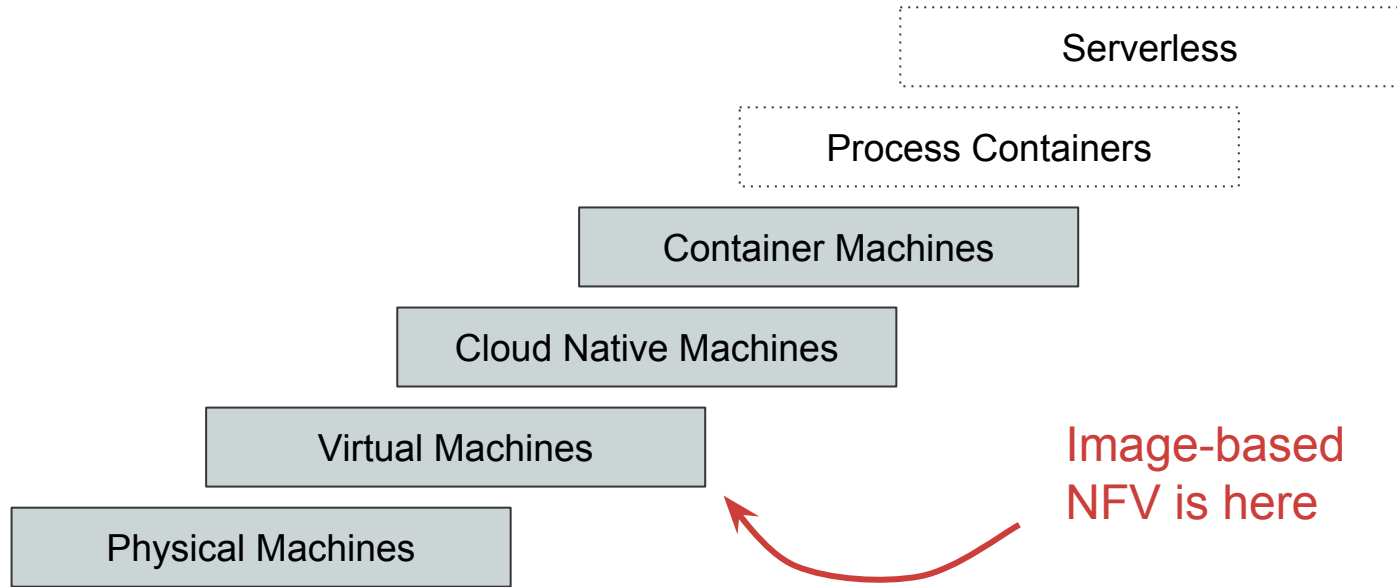# Cloud-Native VNF Modelling

Juju Charms as a Generic VNFM in OSM

In the rest of the world...

Serverless

Process Containers

Container Machines

Cloud Native Machines

Virtual Machines

Physical Machines

Image-based
NFV is here

Appliances → Images → Cloud-Native

Open Source
MANO

This stuff is still crazy even on AWS

Serverless

Process Containers

Container Machines

Cloud Native Machines

Virtual Machines

Physical Machines

Juju gets you to here

Bridging the old and the new

"Stand up a service"

NSO - RIFT.io

"Configure service"

VNF Image

"Stand up an image"

RO - OpenMANO

VIM

Bridging the old and the new

Open Source MANO

Old Appliance

VNF Charm

VNF Image

VCA - Juju

"Stand up a service"

NSO - RIFT.io

"Configure service"

"Stand up an image"
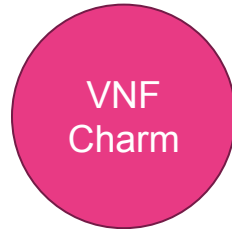
Instantiate Ops Code

RO - OpenMANO

VIM

Charms are shared operations code
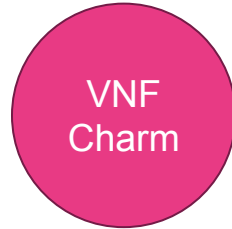
VNF
Charm

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

Shared operations code
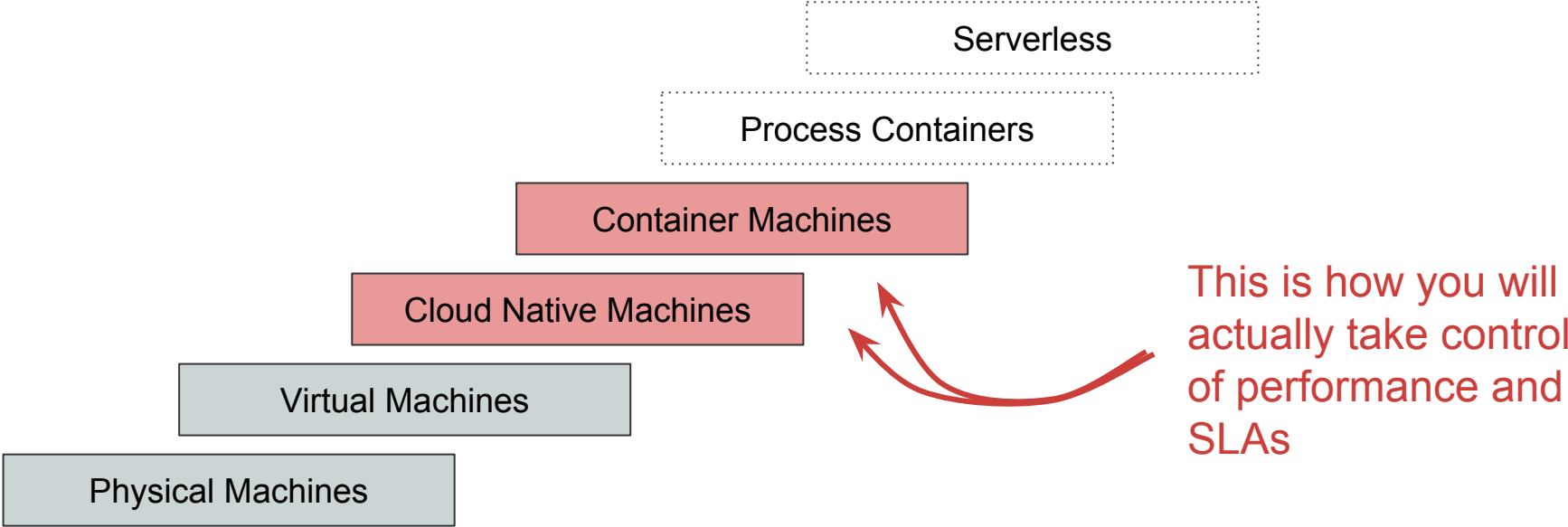reduces the time to onboard
and to train engineers
on a new VNF

Charms are cloud-native model-driven ops

VNF
Charm

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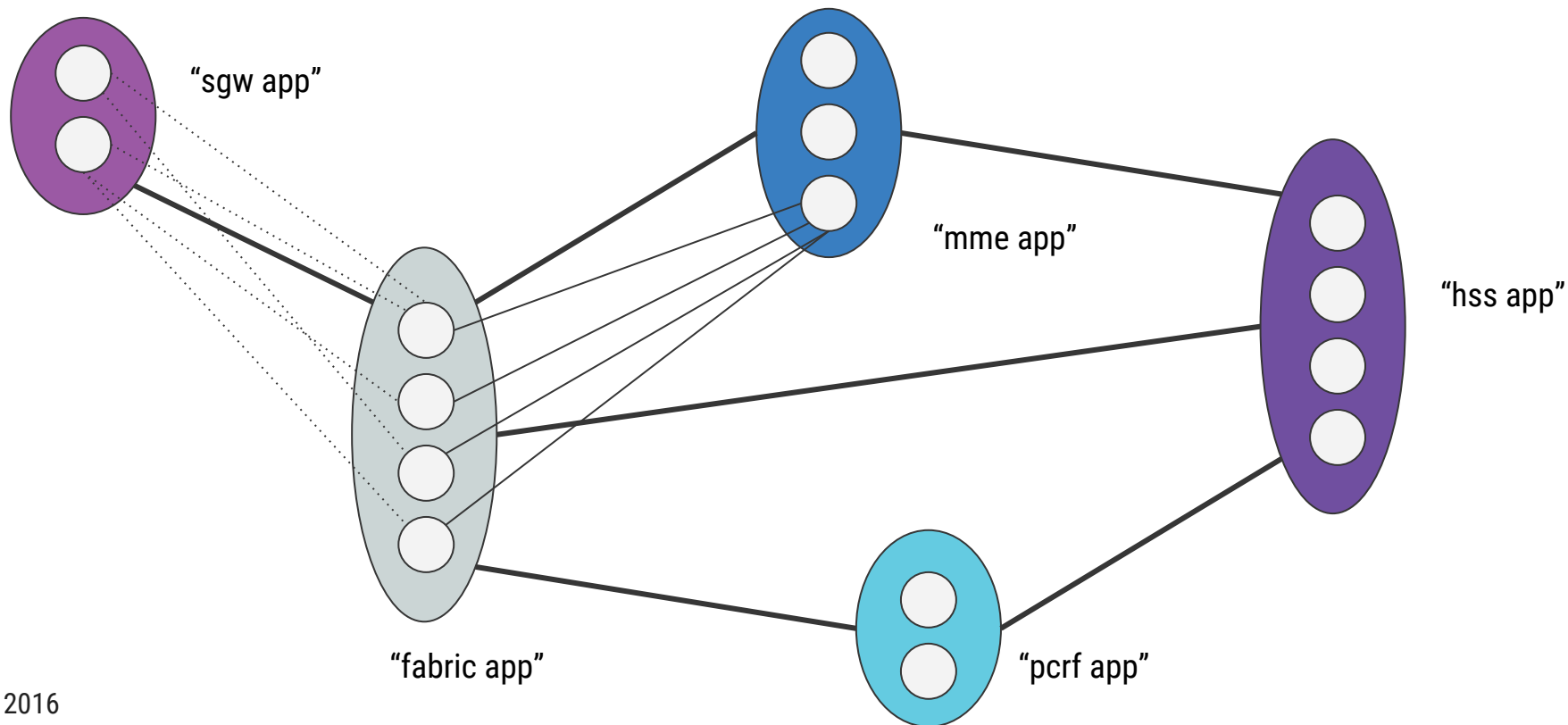- Integration
- Operations - backup, workflows, monitoring

New generation of VNFs are
scale-out and cloud-native

Open Source
MANO

Model-driven operations

Serverless

Process Containers

Container Machines

Cloud Native Machines

Virtual Machines

Physical Machines

This is how you will actually take control of performance and SLAs

What's in a model?

"sgw app"

"mme app"

"hss app"

"fabric app"

"pcrf app"
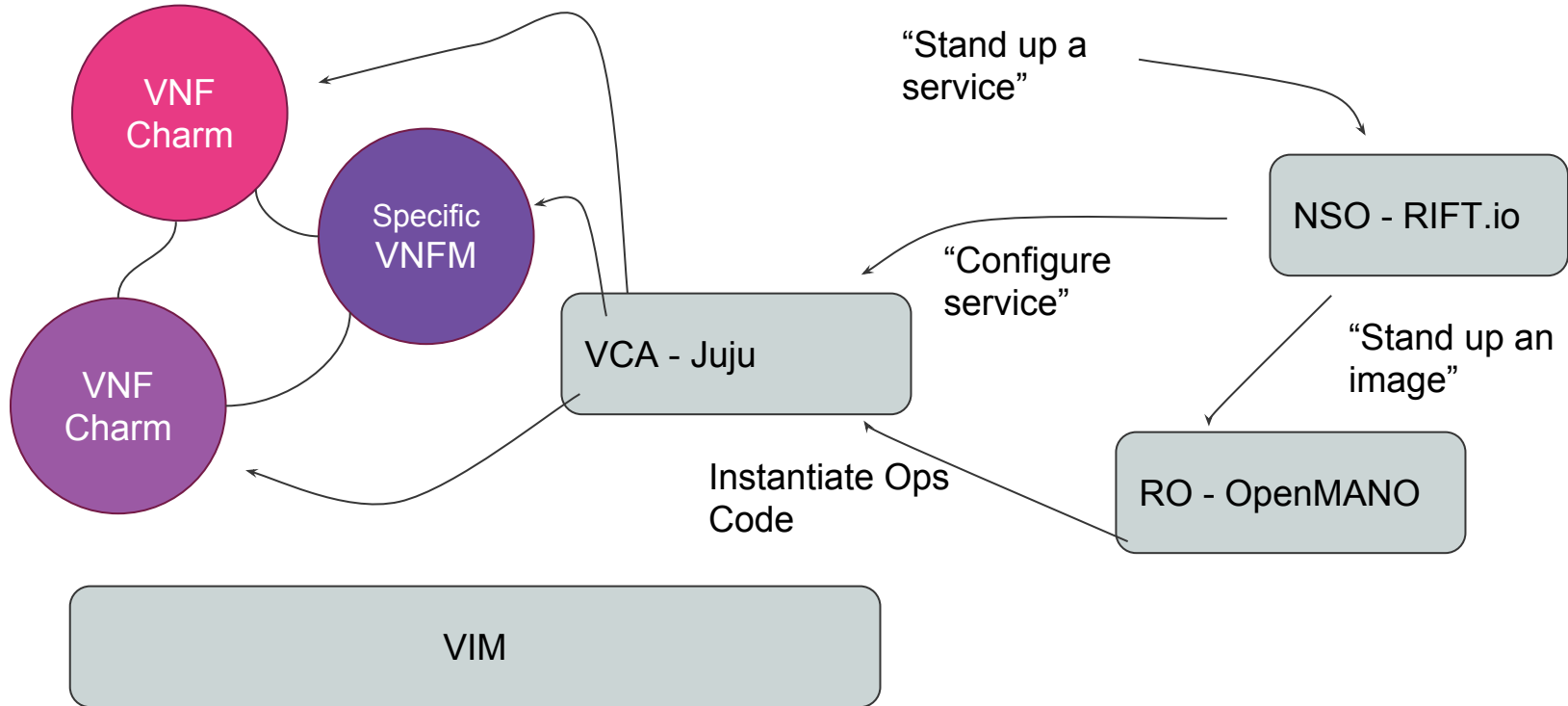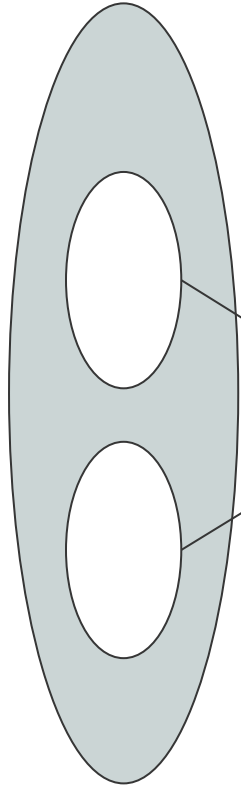
Looking forward

Applications? Units?

"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

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

● Lifecycle - install, upgrade

● Scale - horizontal elasticity

● Integration

● Configuration

Charm Structure

metadata.yaml

./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.
  tags: ['4G-5G', 'EPC', 'OpenAirInterface']
  provides:
    spgw:
      interface: S11
  requires:
    hss:
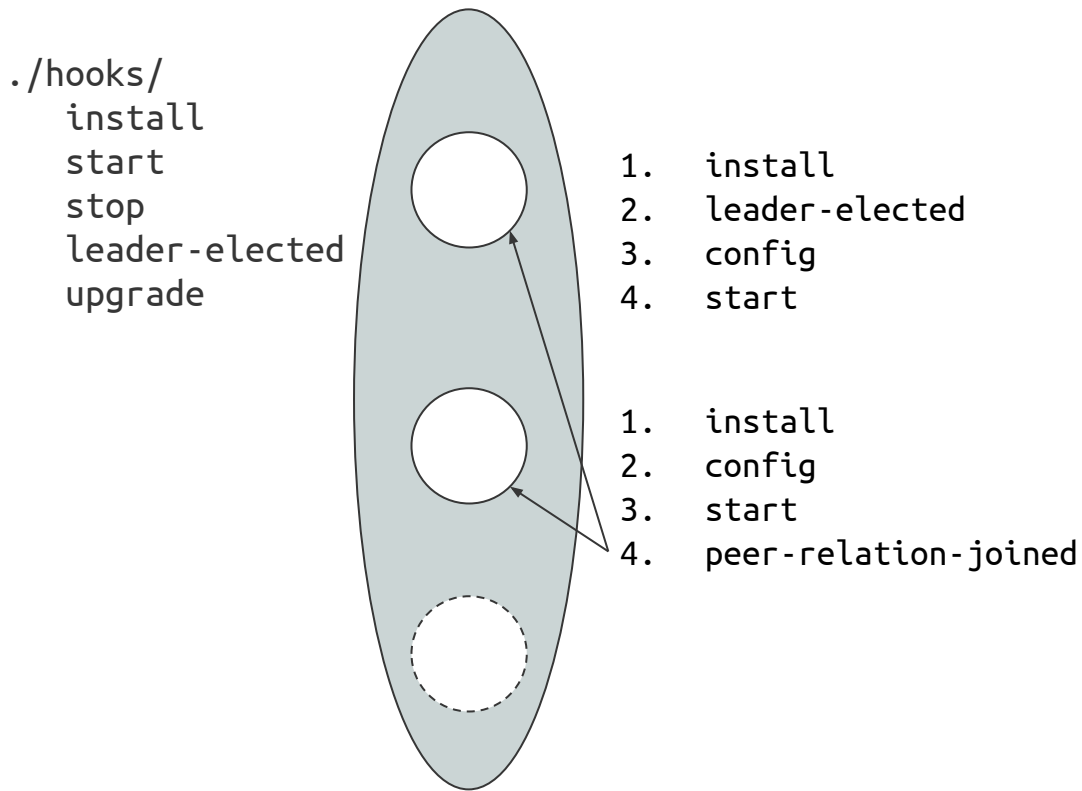   interface: S6a-hss

"consumes S11"

B
mme

"S11 interface"

spgw
A

"provides S11"
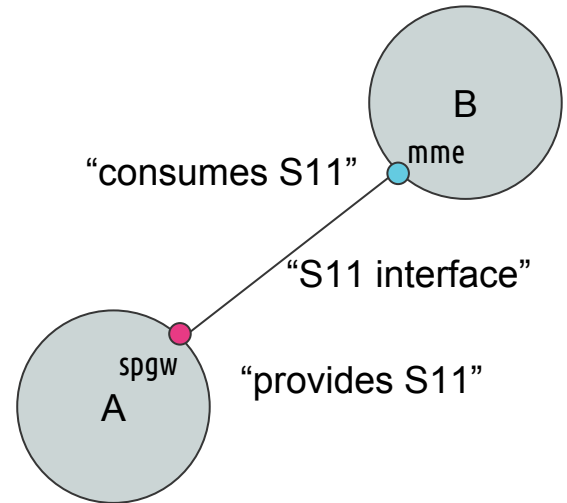
Charm Structure

Lifecycle Hooks

Install

Scale

Update

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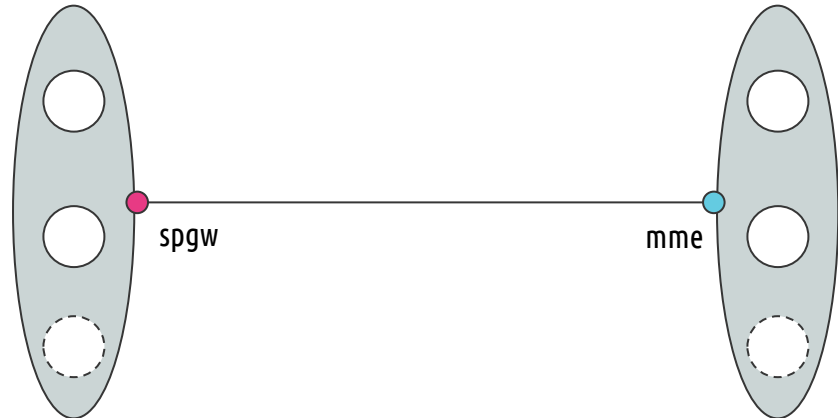
```
./hooks/
    X-relation-joined
    X-relation-changed
    X-relation-broken
    X-relation-departed

    peer-relation-*
```


spgw

mme

Charm Structure

Integration Hooks

Relations and Interfaces

Peers

s11-relation-joined

spgw

mme

Charm Structure

Integration Hooks

Relations and Interfaces

Peers

S11-relation-changed hook

config hook

spgw

mme

Apply new config
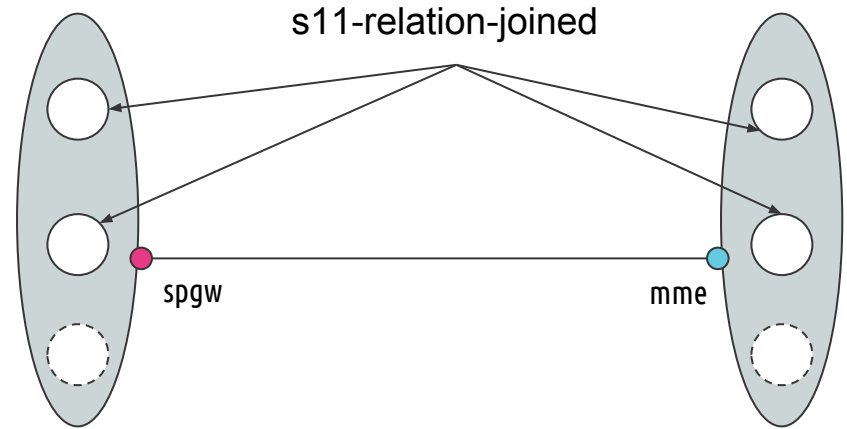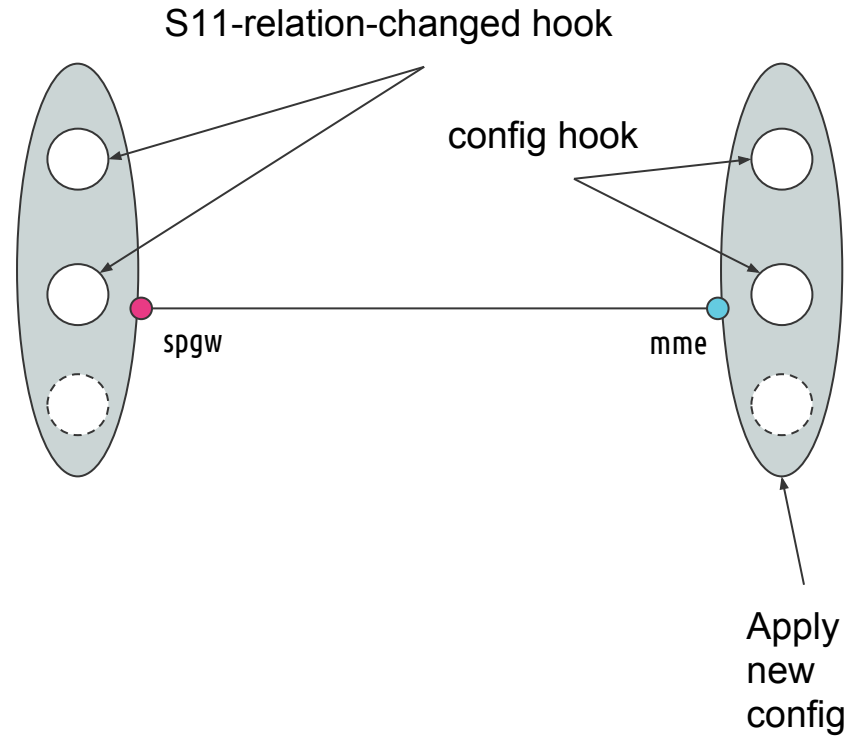
Charm Structure

Operations Hooks

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

spgw