NOKIA

Template system for VNF management

An open and standards compliant solution

- Dániel Fey — Product Manager, CloudBand Application Manager
- Gábor Márton — Chief Architect, CloudBand Application Manager

- 07-06-2017

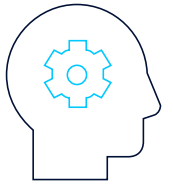
The future is now: telco goes cloud

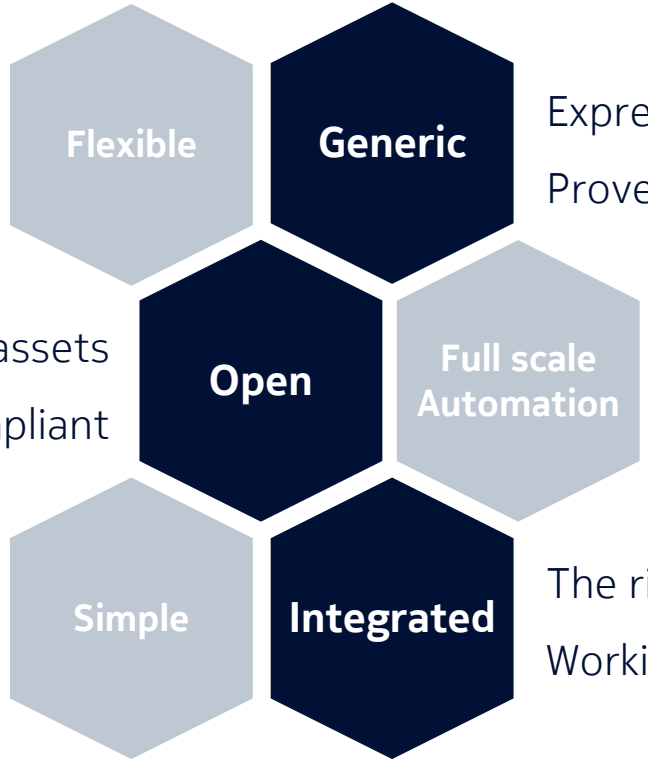
Template System: the key for repeatable automation

Repeatable

Automation

Template System: the cornerstone of cloud adoption





Expressive and capable

Proven in the field

Builds on community assets

Standards compliant

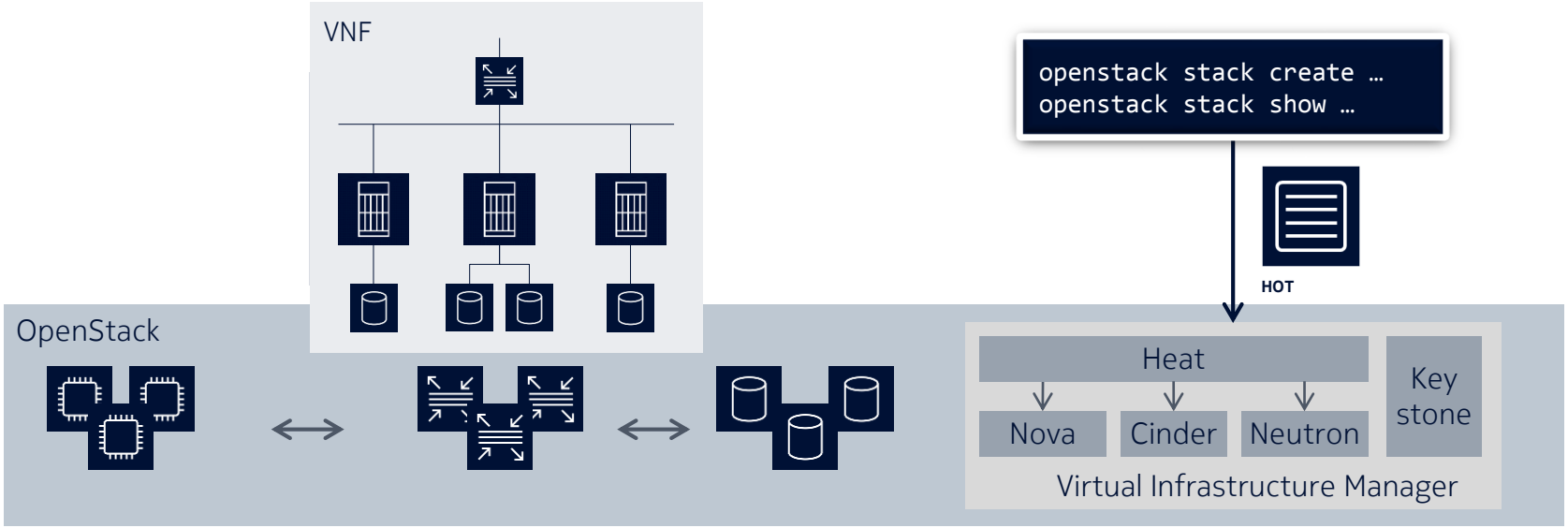
The right tools for the right job

Working all smoothly together

Manage cloud resources

1

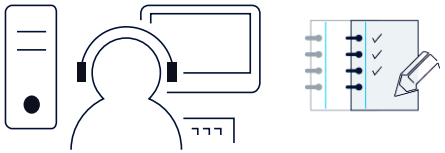
- Joe creates cloud resources
- Joe verifies that everything OK and hands over

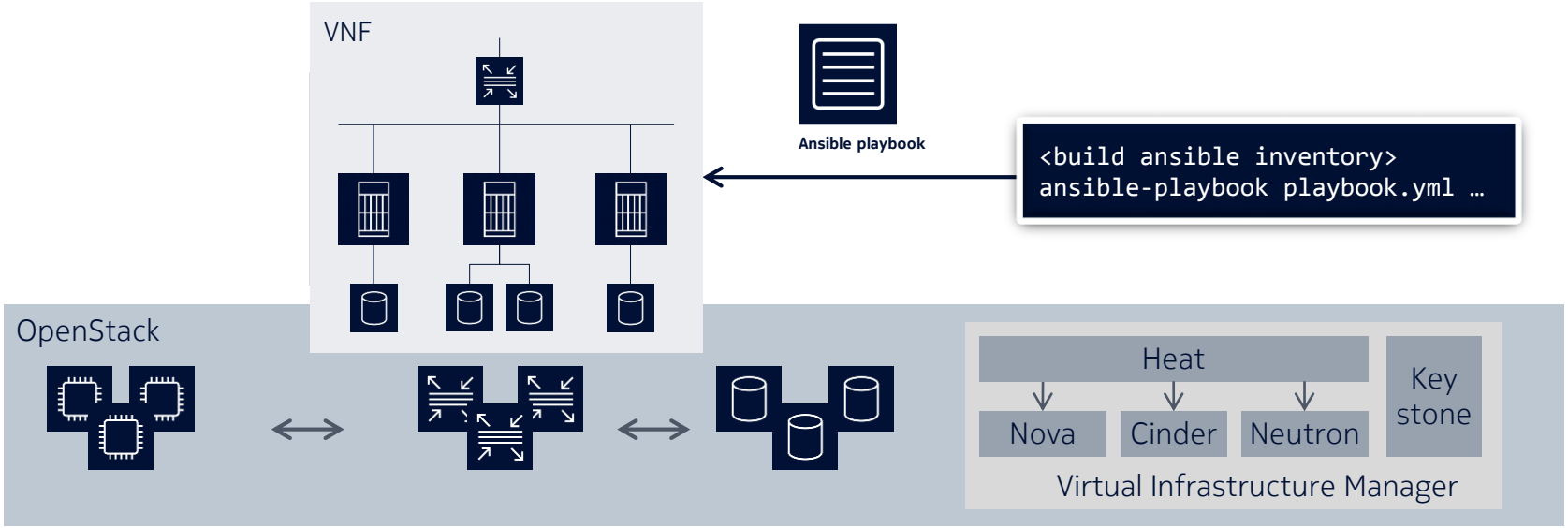


Commission the VNF

2

- Paul does initial configuration of the VNF
- Paul hands over to EMS configuration

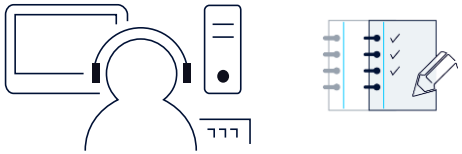


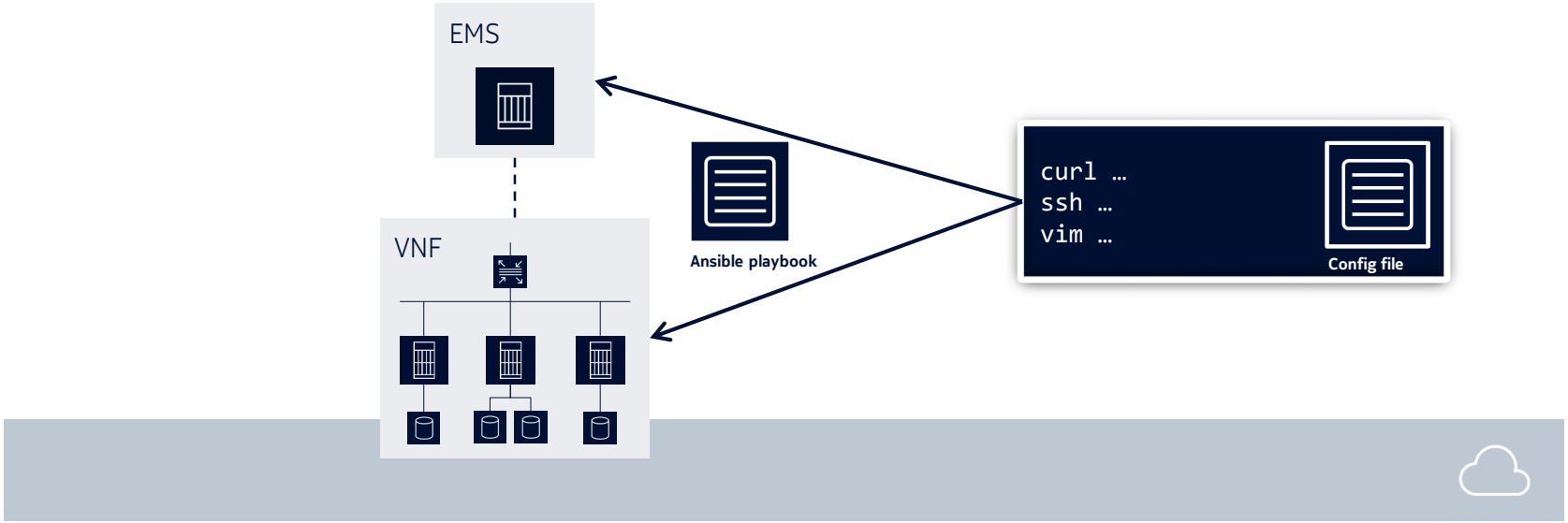


Integrate with the EMS

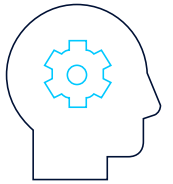
3

- Greg integrates the VNF with the EMS
- EMS starts VNF configuration and monitoring





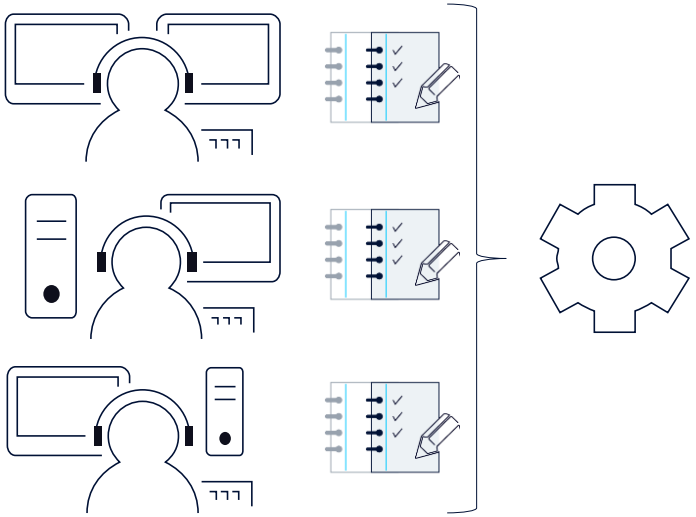
Manually intensive tasks



1
- Joe creates cloud resources
- Joe verifies that everything OK and hands over

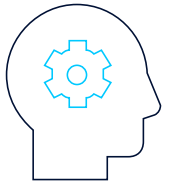
2
- Paul does initial configuration of the VNF
- Paul hands over to EMS configuration

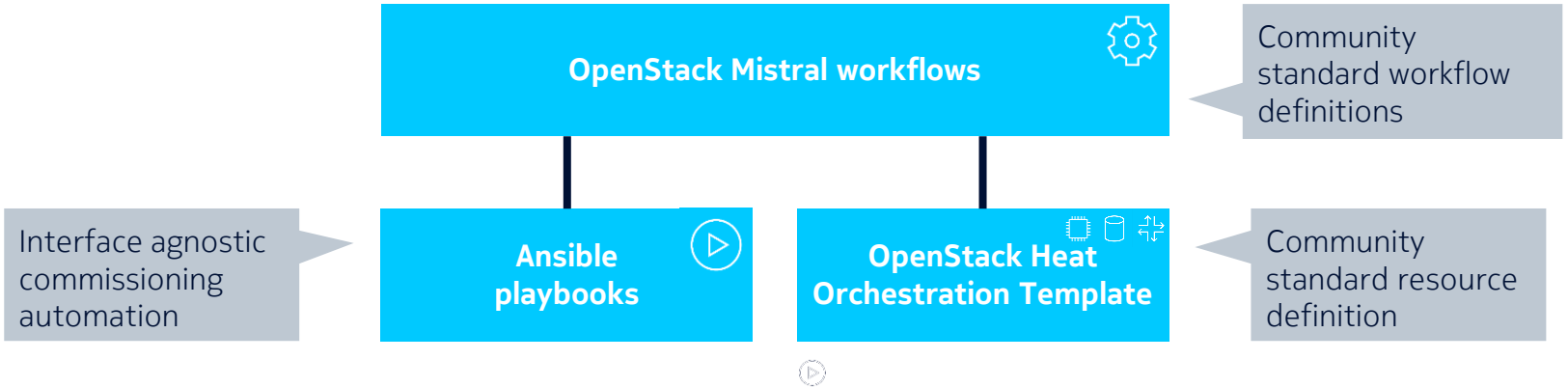
3
- Greg integrates the VNF with the EMS



Higher scale and reduced human errors drive the need for repeatable automation

Composing the VNF management template system – automation

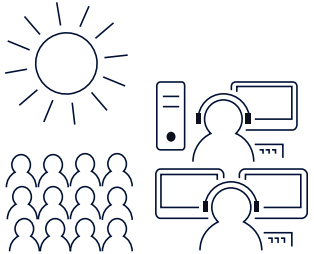


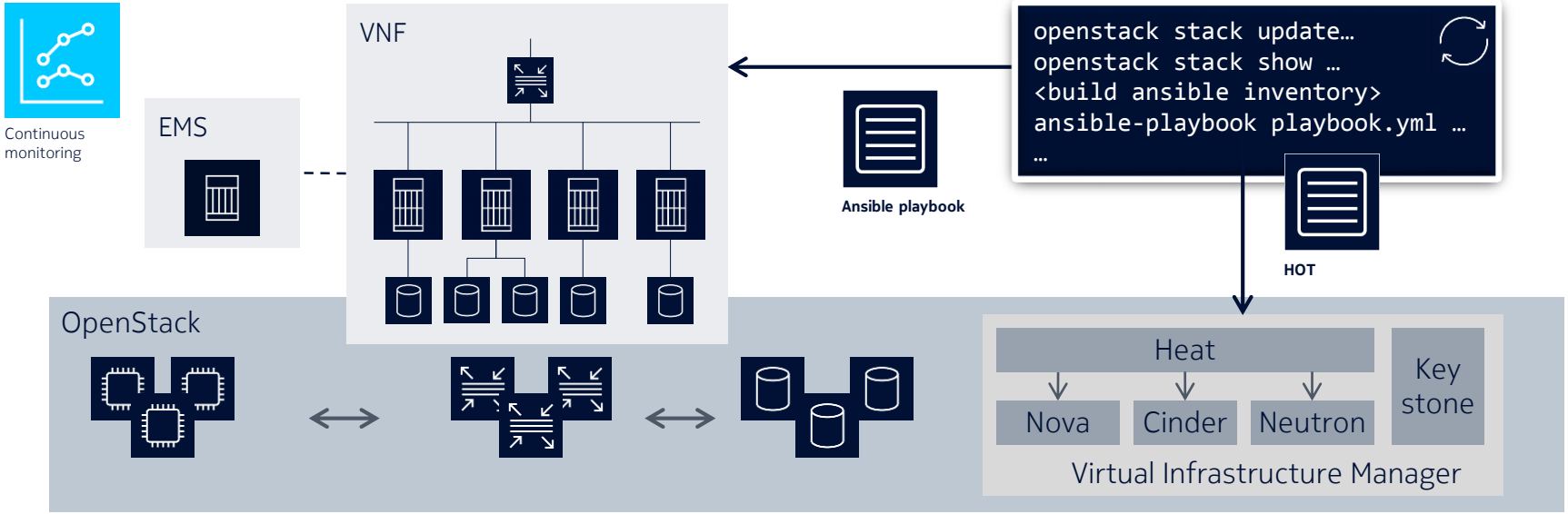


Scaling the VNF

4

- VNF scaling is a natural advantage of cloud
- But it must be automated

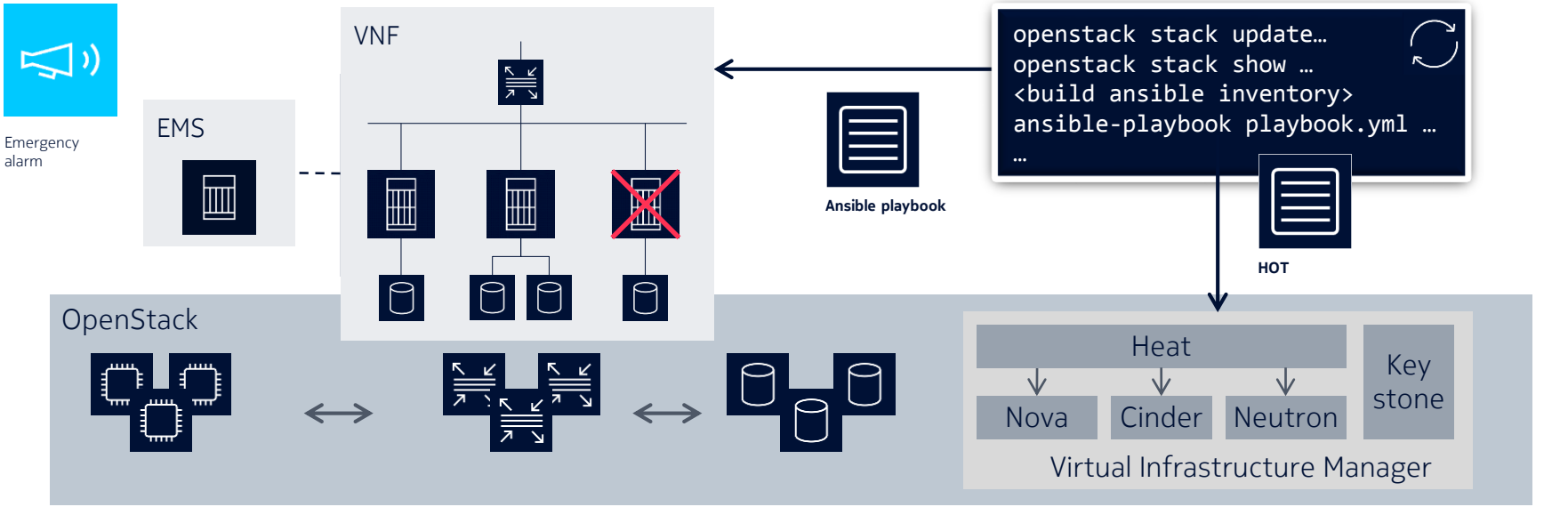




Healing the VNF

5

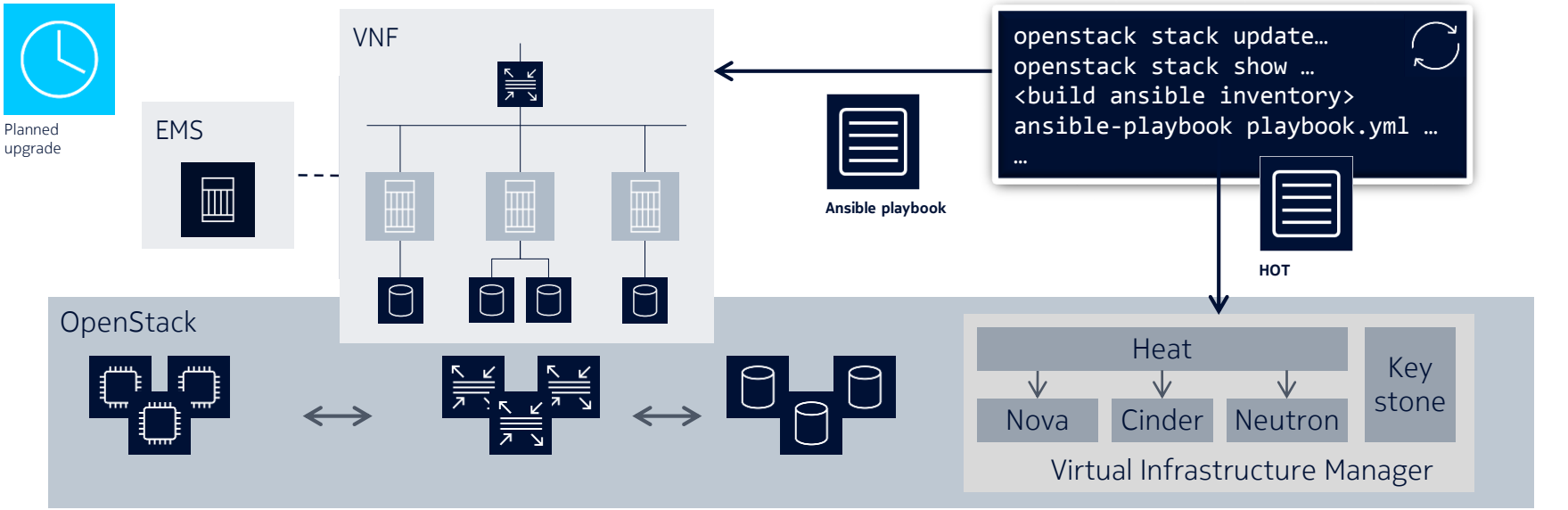
- Healing requires cross-functional efforts
- Whole team joins and performs recovery MOPs



Upgrading the VNFs

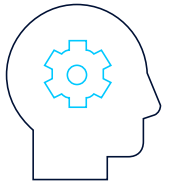
6

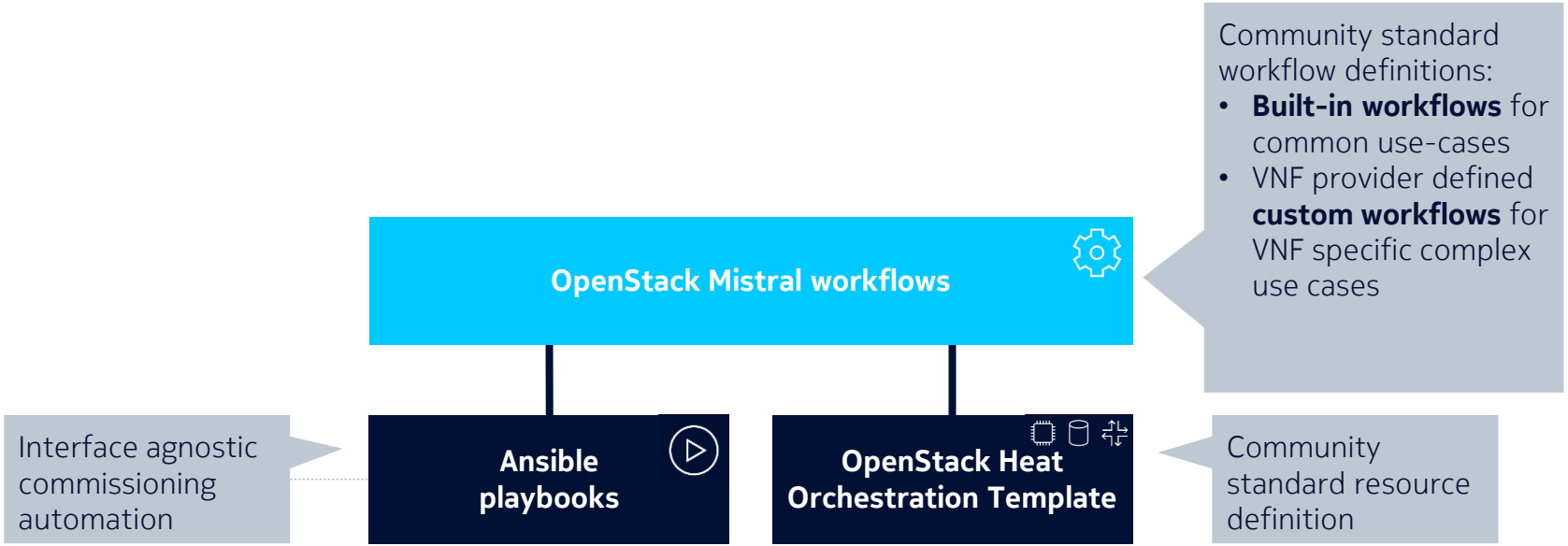
- Simplifying one of the most complex procedures (in-service, online upgrades)



NOKIA

Composing the VNF management template system – Mistral workflows

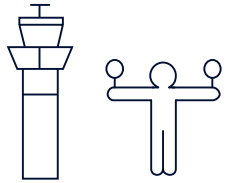


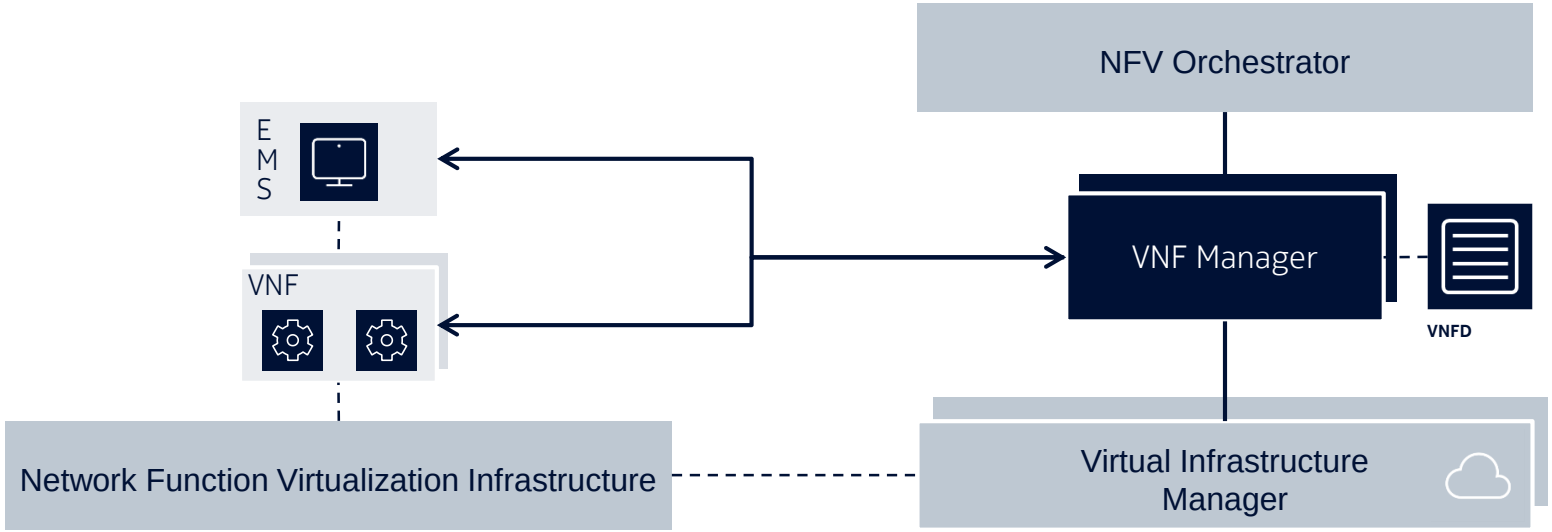


NFVO interoperability

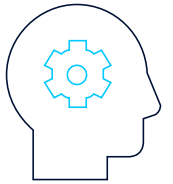
7

- NFVO controls the end-to-end infrastructure
- VNFM – NFVO interoperability extends template system





But which VNFD format to use?



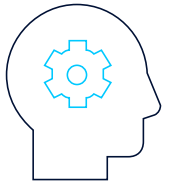
Invent yet another VNFD format?

- Adds noise to complicated market
- No reuse of existing artifacts
- Requires steep learning curve
- Risks vendor lock-in (even with open-source projects)

Go for TOSCA!

- ETSI NFV delivered a normative spec for the VNFD data model: IFA 011
- TOSCA shows deep expertise in cloud orchestration templates
- ETSI NFV selected OASIS TOSCA for standardizing the VNFD template format

Composing the VNF management template system – VNFD



Standards compliant VNF lifecycle capabilities and external exposure:

- **Lifecycle operations** and their parameters
- **VNF level parameters**
- VIM-agnostic **topology definition**
- VIM agnostic **resource consumption**

VNF Descriptor in TOSCA

OpenStack Mistral workflows

Community standard workflow definitions:

- Built-in workflows for common use-cases
- VNF provider defined custom workflows for VNF specific complex use cases

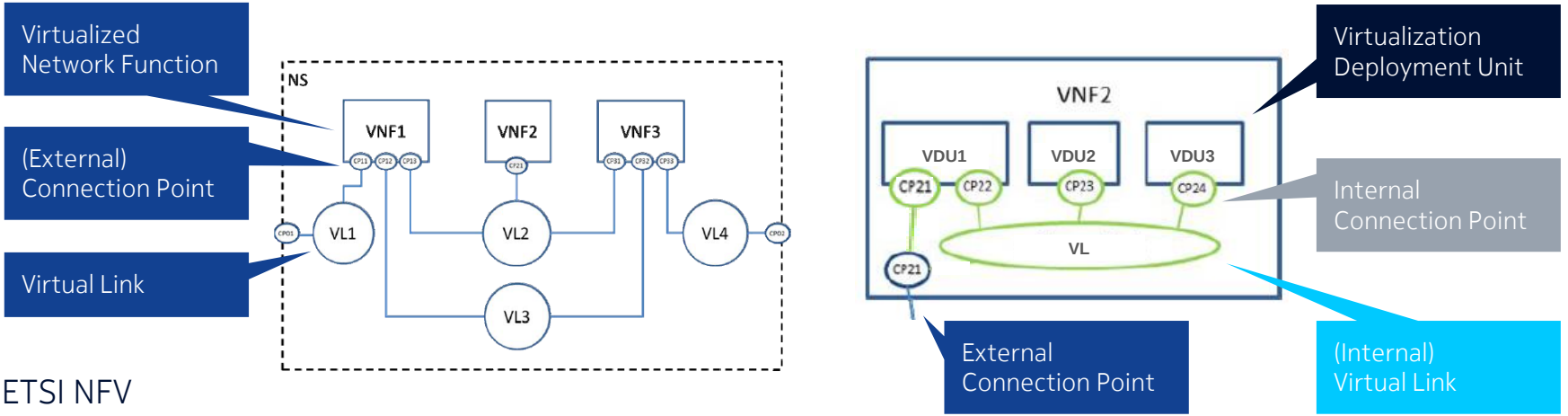
Interface agnostic commissioning automation

Ansible playbooks

OpenStack Heat Orchestration Template

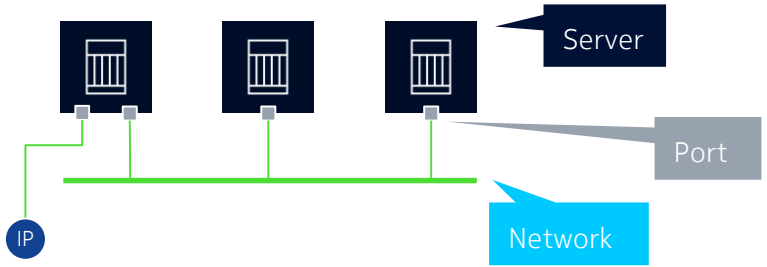
Community standard resource definition

Composing the VNF management template system – VNFD
Key concepts



ETSI NFV

OpenStack



Composing the VNF management template system – VNFD
TOSCA structure overview: overall

VNFD = TOSCA YAML service template

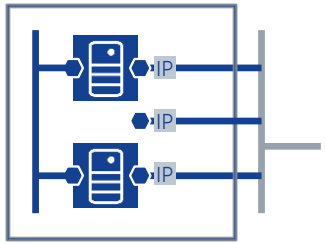
boilerplate

node_templates, policies, …

VDU, VL, ECP, …

Heat mapping, …

Composing the VNF management template system – VNFD
Overall topology and LCM operations




```
topology_template:
  substitution_mappings:
    type: tosca.nodes.nfv.VNF
    properties:
      provider: Nokia
      product_name: vFW
      software_version: 1.0
      ..
    interfaces:
      Basic:
        instantiate:
          inputs:
            additional_parameters:
              param: null
        terminate:
      Scalable:
        scale:
    requirements:
      - virtual_link: [ server1_ecp,
                        external_virtual_link]
    ..
  node_templates:
    server1:
      type: tosca.nodes.nfv.VDU
      ..
```

Enhanced semantics: partially defined node *template* (beyond node *type*)

LCM operations supported

Expose ECP

VDUs

```
    ..
    internal_vl:
      type: tosca.nodes.nfv.VL
    ..
    server1_ecp:
      type: tosca.nodes.nfv.ECP
      ..
    server1_icp:
      type: tosca.nodes.nfv.ICP
      ..

    moving_ip_ecp:
      type: tosca.nodes.nfv.ECP
      properties:
        layer_protocol: ipv4
      requirements:
        - internal_connection_point: moving_ip_icp
    moving_ip_icp:
      type: tosca.nodes.nfv.ICP
      ..
```

Virtual link

External connection point

Internal connection point

Composing the VNF management template system – VNFD
Customizable LCM operations (Nokia-specific extensions)

```
topology_template:
  substitution_mappings:
    node_type: tosca.nodes.nfv.VNF
    ..
    interfaces:
      Basic:
        instantiate:
          inputs:
            extensions:
              pre_actions:
                - javascript: javascript/assign-zones.js
                  output: stack_parameters
              post_actions:
                - ansible: ansible/configure-all.ansible.yaml
        terminate:
          ..
      Healable:
        heal:
          implementation: heal.workbook.mistral.yaml
      Custom:
        do_something:
          implementation: do-something.workbook.mistral.yaml
```

run before stack manipulation

extend/change pre-calculated stack parameters

run playbook on hosts (host groups) in VNF

custom implementation of standard LCM operation

custom operation

Composing the VNF management template system – VNFD
Using NFV model information in HOT: VDU

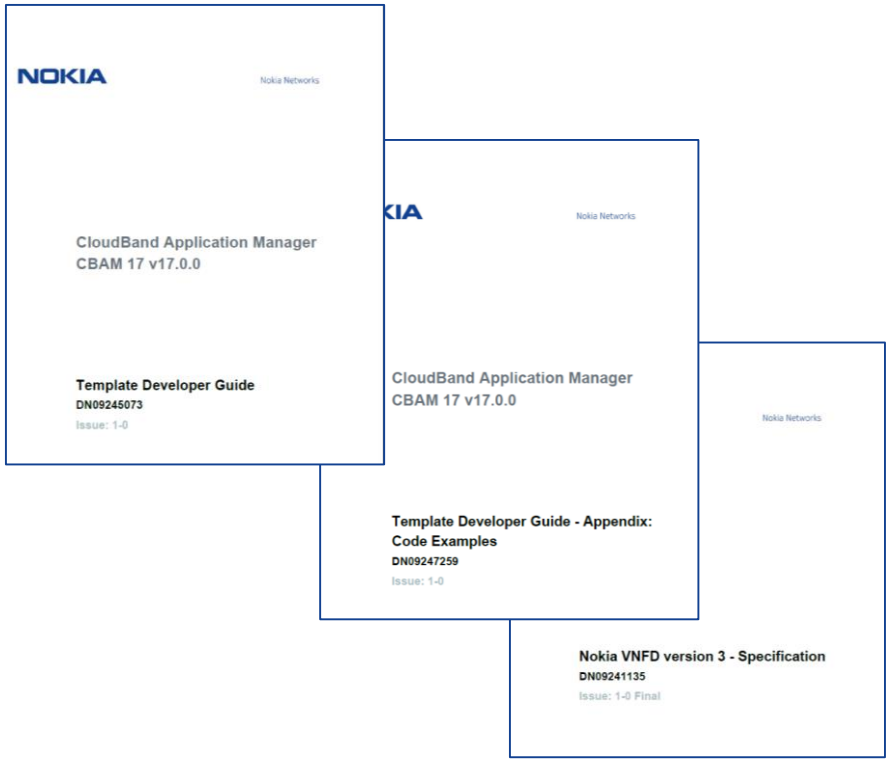
```
heat_template_version: 2015-04-30

parameters:
  cbam:
    type: json

resources:
  port:
    type: OS::Neutron::Port
    properties:
      network: ..
      ..
  floating_ip:
    type: OS::Neutron::FloatingIP
    ..
  server:
    type: OS::Nova::Server
    properties:
      key_name: preservedSshKey
      image: { get_param: [ cbam, resources, server, imageId ] }
      flavor: { get_param: [ cbam, resources, server, flavorId ] }
      ..
      networks:
        - port: { get_resource: port }
```

```
{
    "vnfId": "simple",
    "resources":
    {
            "server":
            {
                "azId": "compute-a",
                "flavorId": "m1.tiny",
                "imageId": "cirros"
            }
    },
    "externalConnectionPoints":
    {
        "universal_ecp":
        {
            "networkId": "pub_net"
        }
    },
    "virtualLinks":
    {
        "internal_vl":
        {
            "networkId": "preservedNetwork"
        }
    }
}
```

provided by NFVO (grant response)

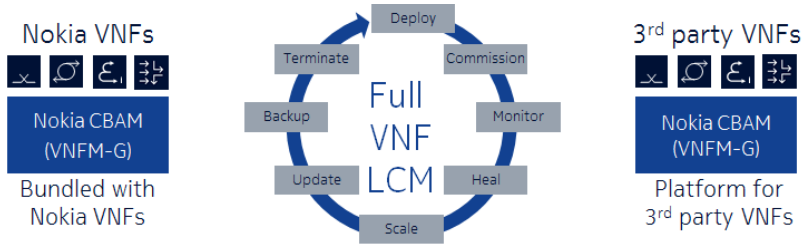
NOKIA

Our experiences with VNF open templating



Documented interfaces:

- VNFD specification, Template Developer Guide
- Lifecycle management API specification
- Lifecycle change notification API specification



NOKIA

The value of an open system for VNF templates

Solves major challenge of VNFs truly becoming part of cloud game

Brings VNFs to life by automating their lifecycle management

Fosters maximum choice of VNFs, so as to offer subscribers the best available services

A major step forward in realizing NFV ambitions

NOKIA