VNF on-boarding using OPNFV and ONAP

Eric Debeau, Orange

October, 12 , 2017

Context

- VNF
  - Not yet full agreed VNF package outside ETSI/NFV:
    - SOL001TOSCA-based VNFD and NSD specification
    - SOL004TOSCA-based VNF Package specification
  - Variety of VNF configuration solution
  - VNF software will be released in a monthly/weekly/daily basis

- Many infrastructure implementations
  - hardware, controllers, engineering networking choices, security constraints

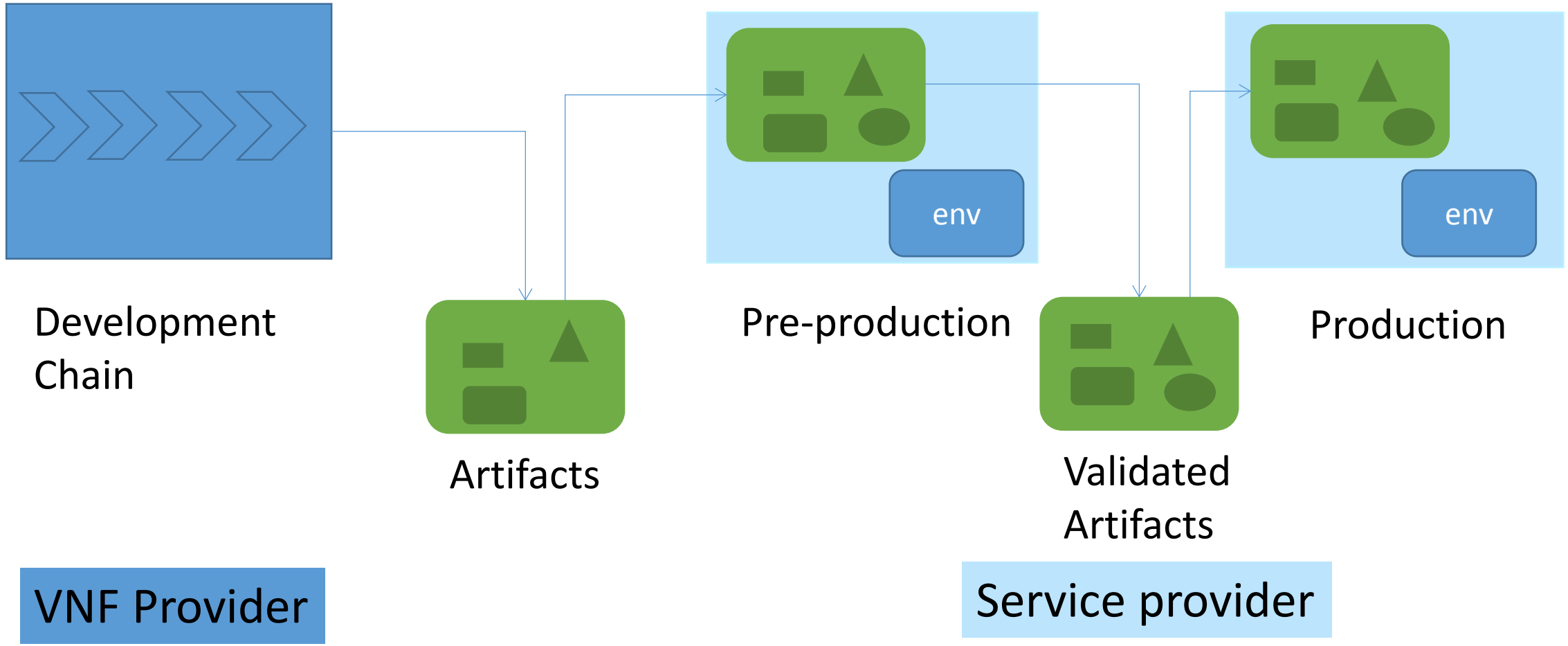
- No more possible to spend days to integrate VNF

On-boarding

- Minimum on-boarding package
  - Descriptor (using Heat, TOSCA, …)
  - Images
  - Additional artifacts (scripts)

- Global on-boarding may also include
  - License model
  - Test
  - Configuration (Netconf/Yang, Ansible Playbook…)
  - Policy rules
  - Security rules
  - Workflows
  - Environment parameters

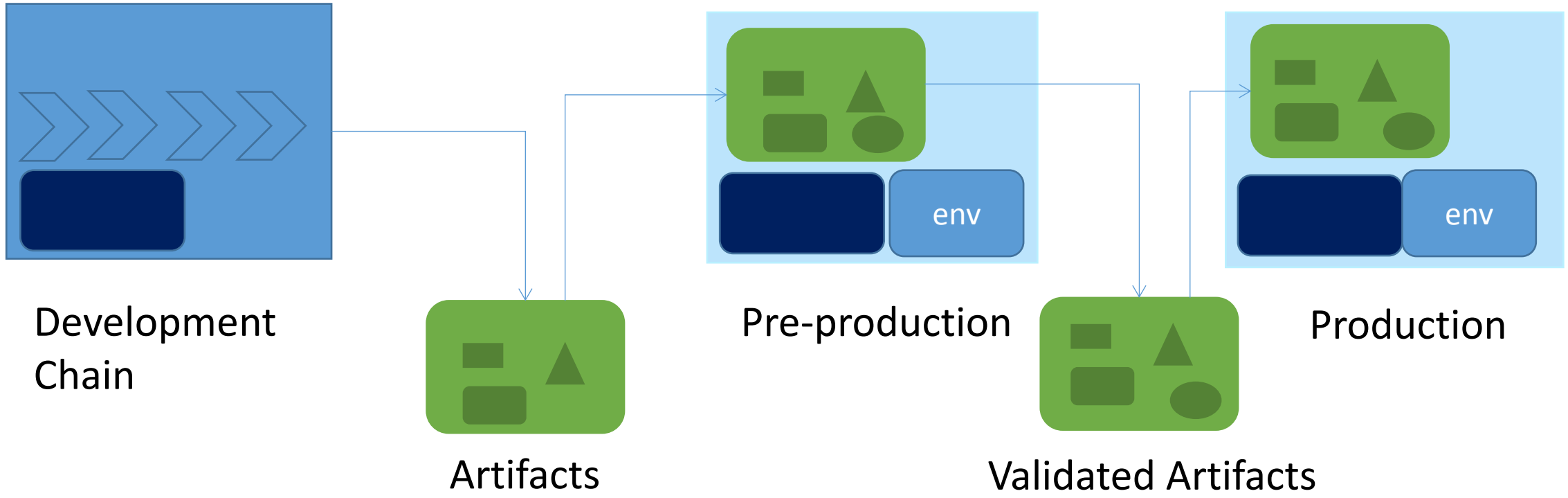
Simplified VNF Pipeline



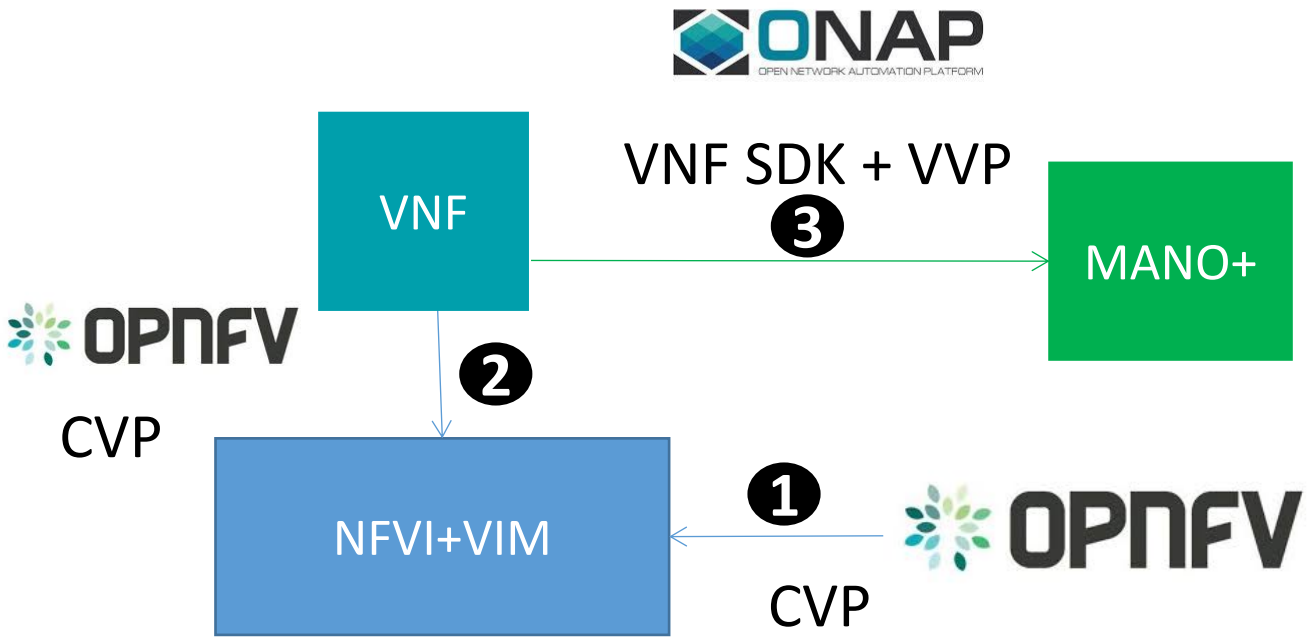
Sharing the tools along the VNF journey

- Open-source solutions as a catalyst for eco-system
  - Reference implementation
  - Tooling to be shared by all actors involved in the NFV chain

VNF 'checking' tools

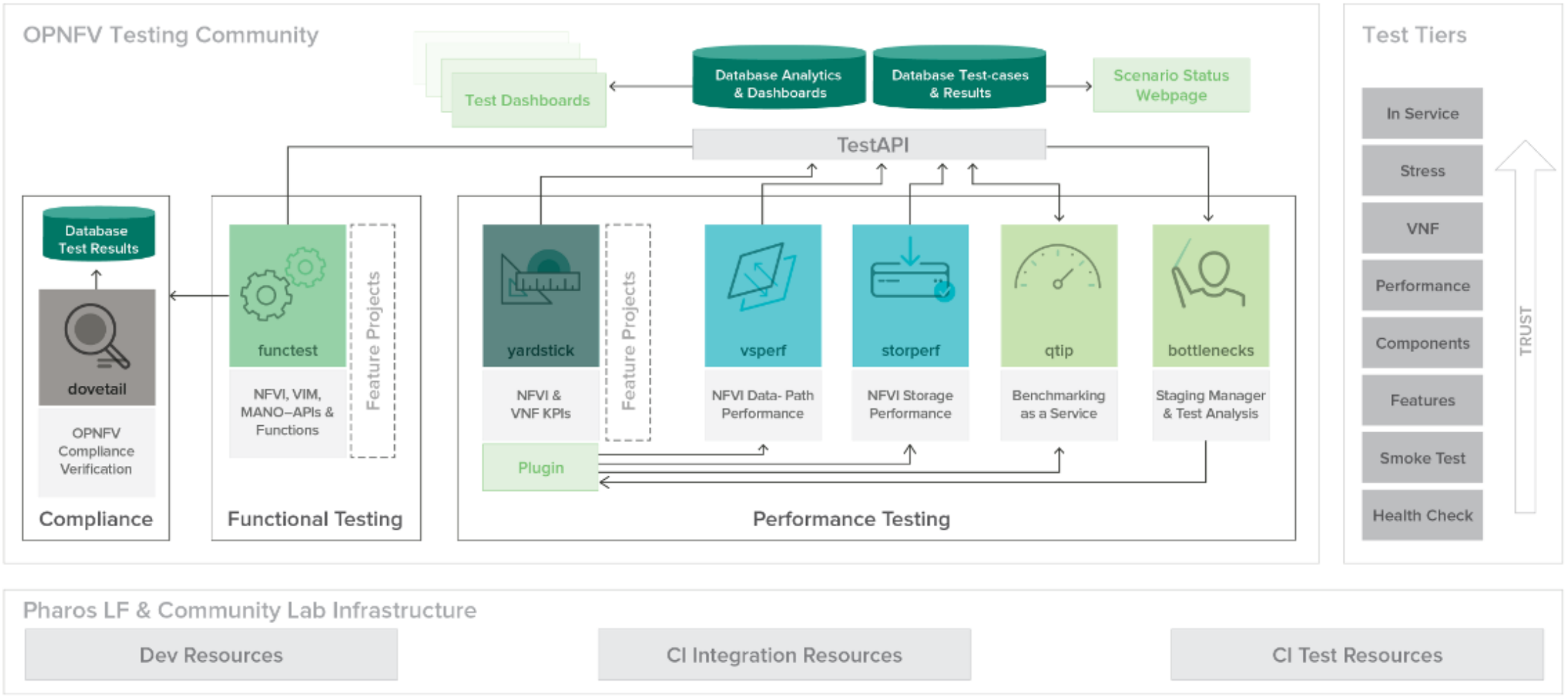


OPNFV & ONAP to “certify” VNF



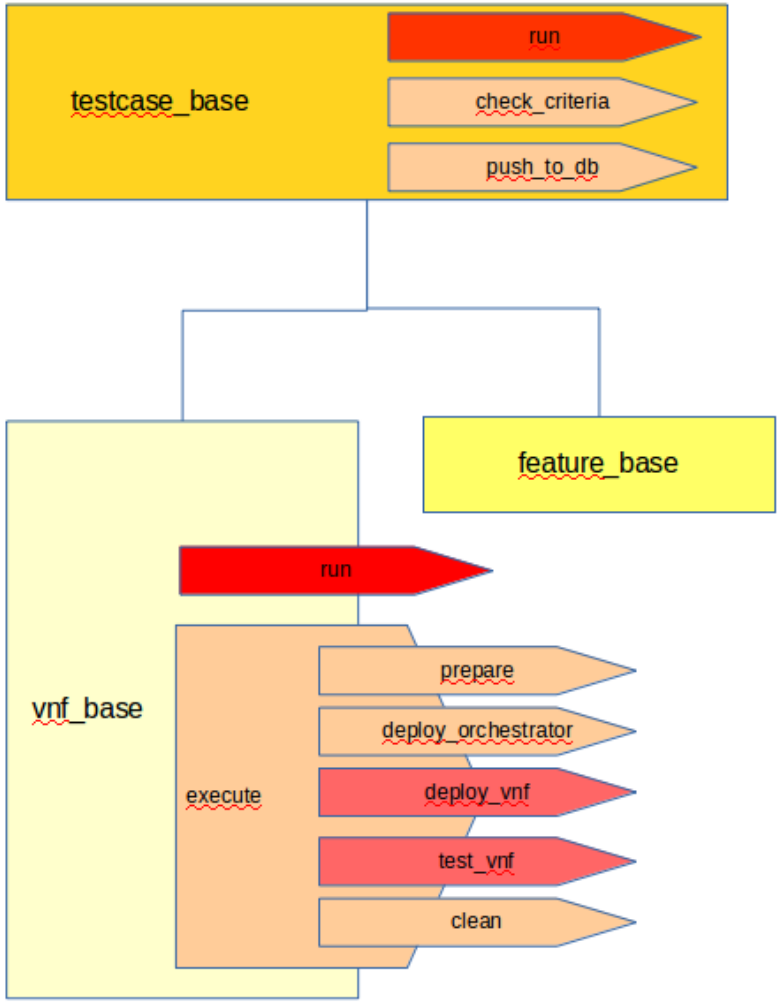
CVP: OPNFV Compliance and Verification Program
VVP: ONAP Verification and Validation Program

OPNFV tests toolkit for Compliance, Functional and Performance



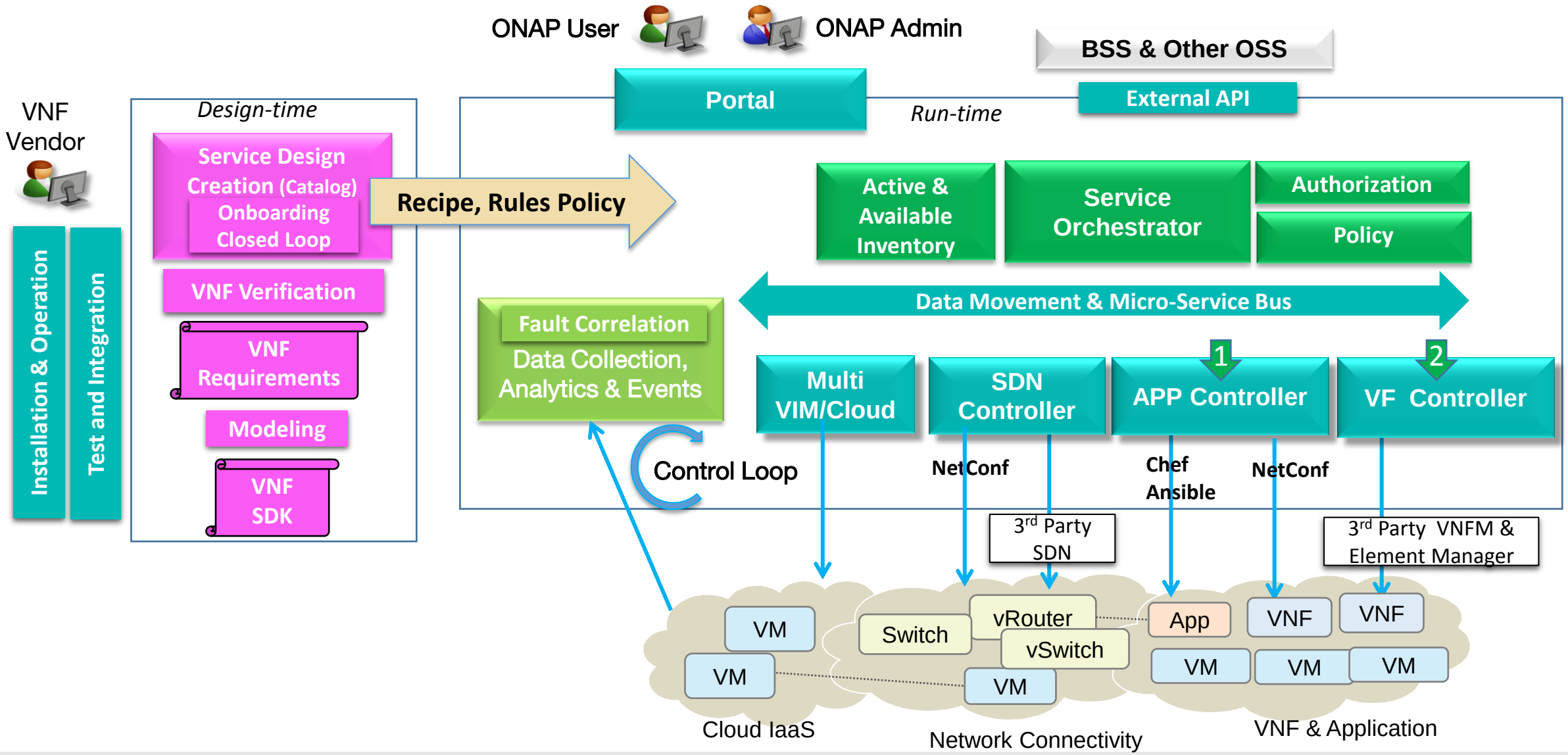
https://wiki.opnfv.org/display/testing

OPNFV: Functest



https://wiki.opnfv.org/pages/viewpage.action?pageId=8687767

ONAP Release Amsterdam: high level architecture



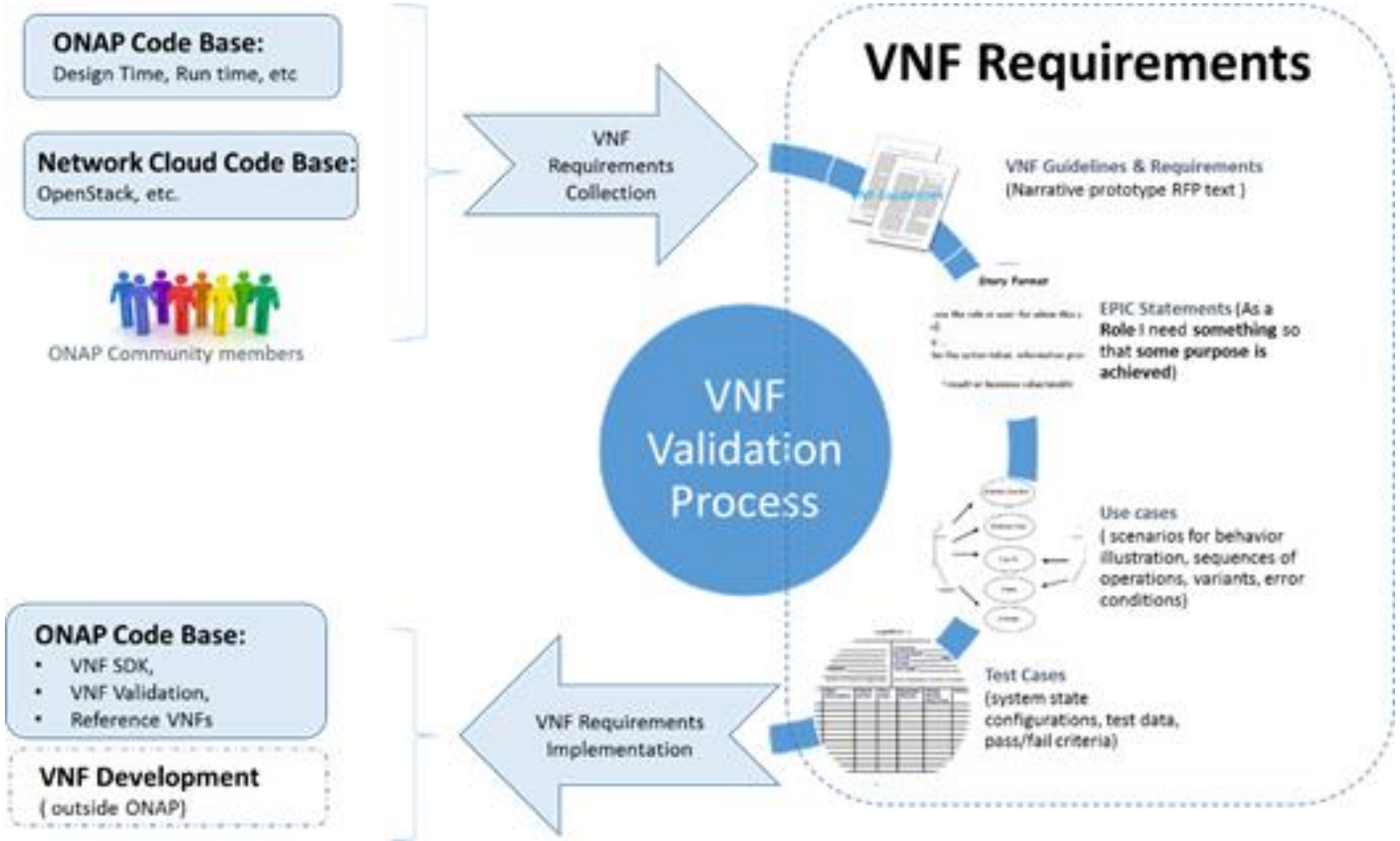
ONAP VNF Requirements

- A set of requirements including
  - Design
  - Resiliency
  - Security
  - Modularity
  - Devops
  - Management
    - Packaging
    - Closed loop
    - Configuration
    - License
- Modeling requirements
  - Heat and TOSCA templates

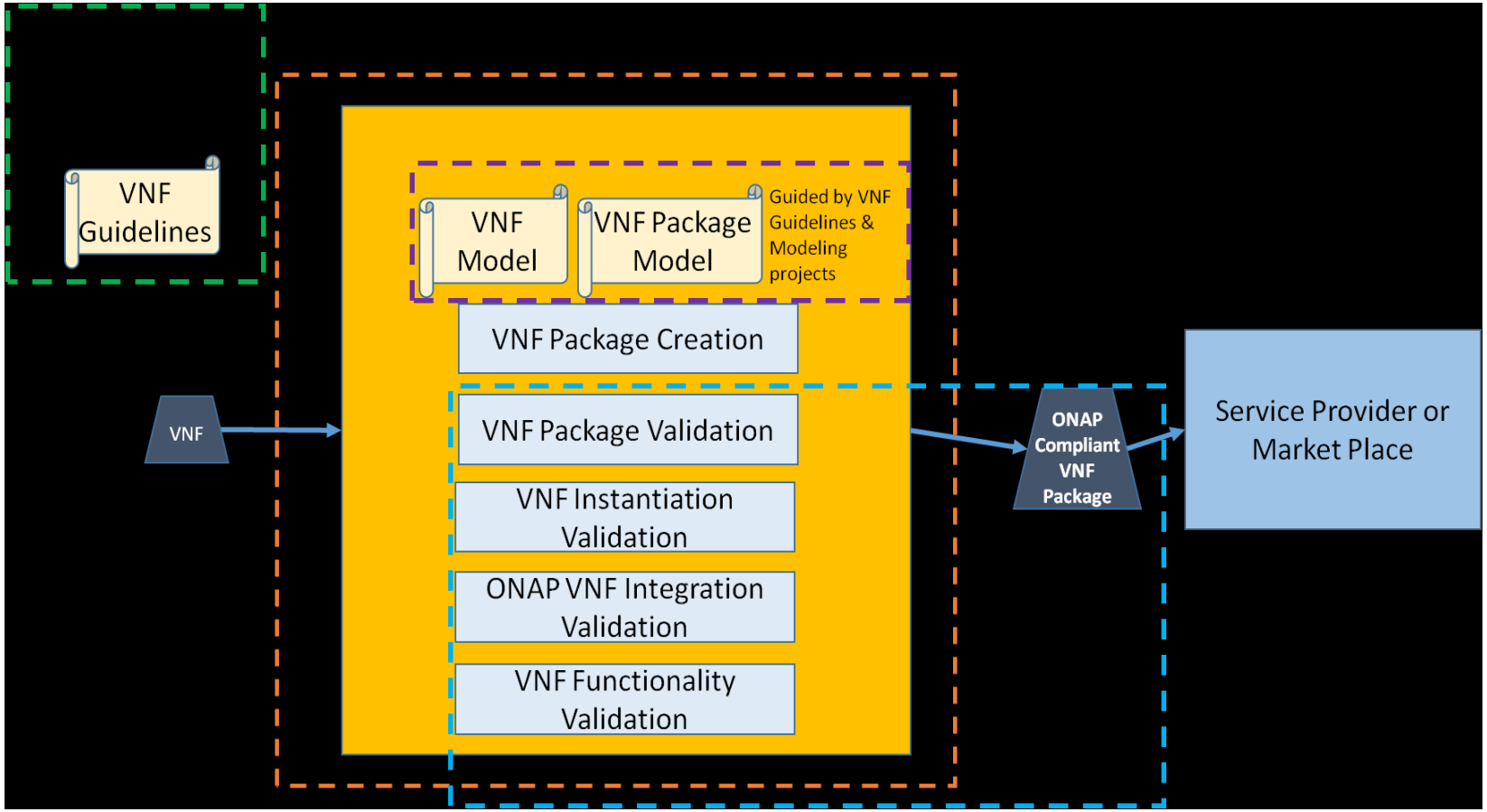
R-46960 The VNF **MUST** utilize only the Guest OS versions that are supported by the NCSP's Network Cloud. [1]
R-23475 The VNF **SHOULD** utilize only NCSP provided Guest OS images. [2]
R-09467 The VNF **MUST** utilize only NCSP standard compute flavors. [2]
R-02997 The VNF **MUST** preserve their persistent data. Running VMs will not be backed up in the Network Cloud infrastructure.
R-29760 The VNFC **MUST** be installed on non-root file systems, unless software is specifically included with the operating system distribution of the guest image.
R-20860 The VNF **MUST** be agnostic to the underlying infrastructure (such as hardware, host OS, Hypervisor), any requirements should be provided as specification to be fulfilled by any hardware.
R-89800 The VNF **MUST NOT** require Hypervisor-level customization from the cloud provider.
R-86758 The VNF **SHOULD** provide an automated test suite to validate every new version of the software on the target environment(s). The tests should be of sufficient granularity to independently test various representative VNF use cases throughout its lifecycle. Operations might choose to invoke these tests either on a scheduled basis or on demand to support various operations functions including test, turn-up and troubleshooting.
R-39650 The VNF **SHOULD** provide the ability to test incremental growth of the VNF.
R-14853 The VNF **MUST** respond to a "move traffic" [3] command against a specific VNFC, moving all existing session elsewhere with minimal disruption if a VNF provides a load balancing function across multiple instances of its VNFCs. Note: Individual VNF performance aspects (e.g., move duration or disruption scope) may require further constraints.
R-06327 The VNF **MUST** respond to a "drain VNFC" [2] command against a specific VNFC, preventing new session from reaching the targeted VNFC, with no disruption to active sessions on the impacted VNFC, if a VNF provides a load balancing function across multiple instances of its VNFCs. This is used to support scenarios such as proactive maintenance with no user impact,

http://onap.readthedocs.io/en/latest/submodules/vnfrqts/requirements.git/docs/index.html

ONAP: various tools to produce the VNF package



ONAP VNF Requirements SDK and Validation projects



Various ONAP projects

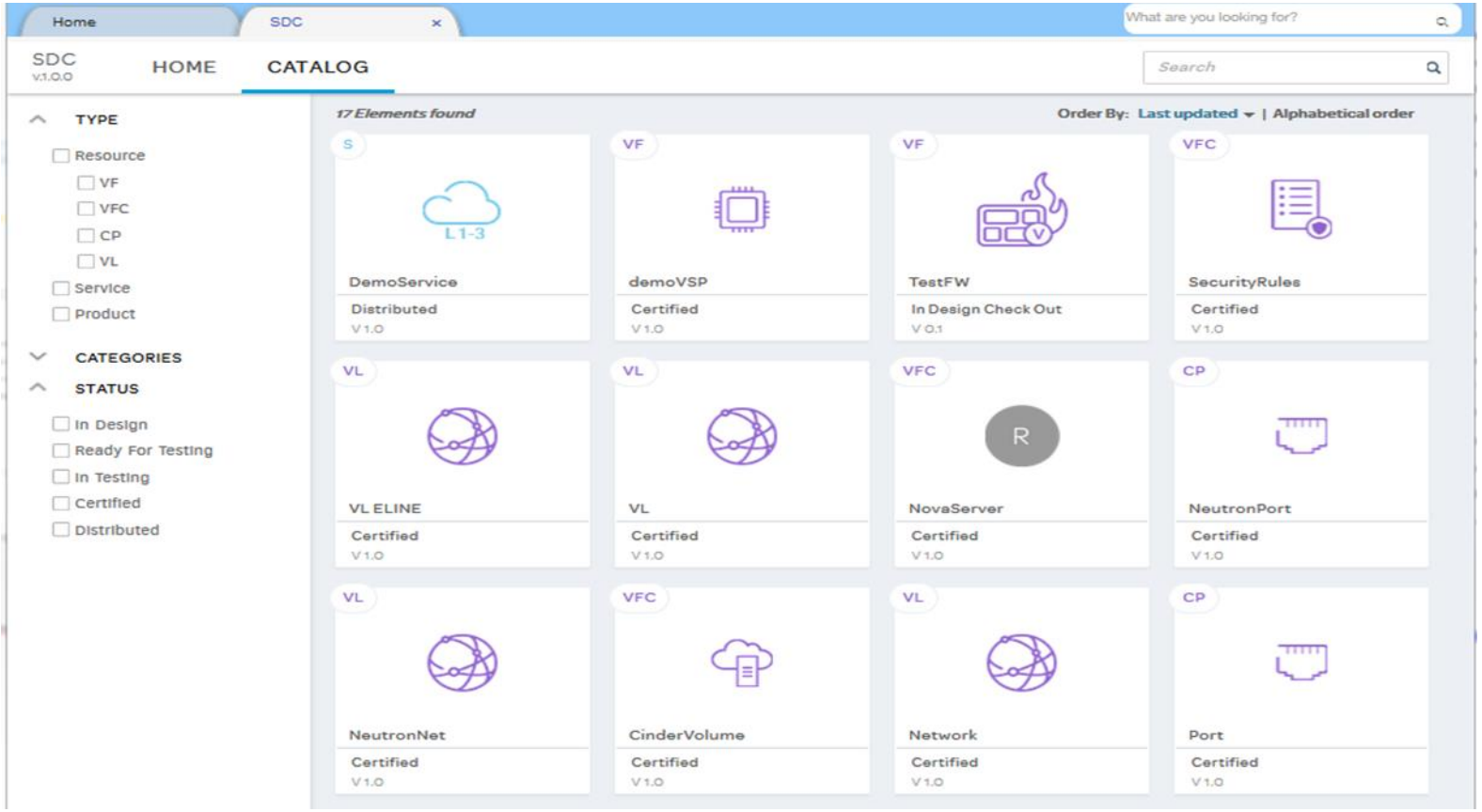
RACI Chart (Roles and Responsibilities Matrix)

	VNF Requirements	VNF SDK	VNF Validation	Reference VNFs	Documentation
Develop VNF Requirements as prototype RFP text	R/A	I	I	I	I
Develop Requirements for VNF Package	R/A	C	I	I	I
Develop Requirements for VNF Validation	R/A	I	C	I	I
Develop Requirements for VNF functionality during service design	R	I	I	I	I
Develop Requirements for VNF functionality during operation	R/A	I	I	I	I
Develop Requirements for VNF operation during in service ONAP and network cloud CI/CD tool chain operations	R	I	I	I	I
Develop VNF requirements reflecting VNF validation processes	R	I	A	I	I

R = Responsible, A = Accountable, C = Consulted, I = Informed

On-boarding example using SDC

http://{{sdc_ip}}:8080/onboarding-api/v1.0/vendor-software-products/{{vsp_id}}/upload



Next steps

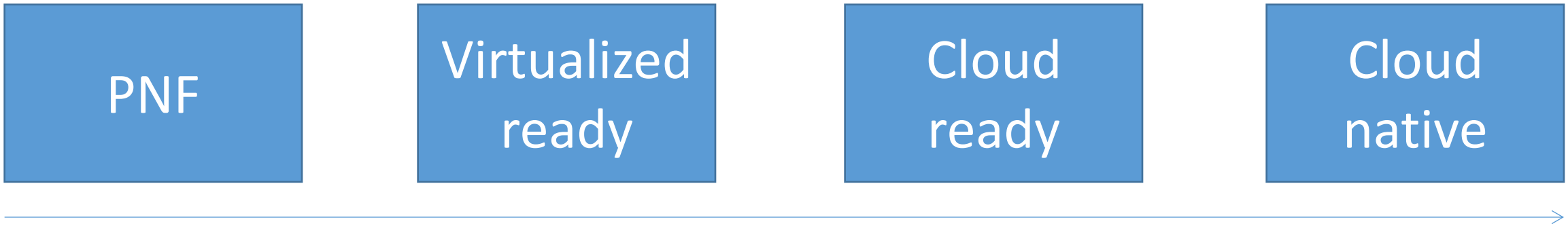
- RE-use OPNFV best practice
  - Functest project to test VNF
  - Tests (functional, perf, security…) should be part of the global VNF package

- Consolidate the VNF Requirements, VNF SDK & VVP
  - To cover both Heat & TOSCA “based VNF”
  - To be extended towards the full VNF on-boarding package
  - How to optimize the validation of new package
  - Every VNF Requirement should be automatically tested

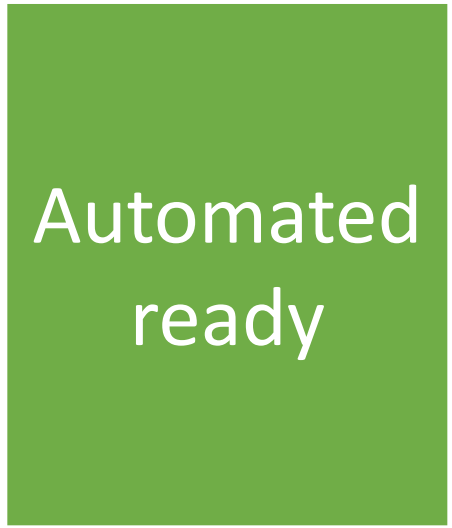
- On-boarding automation, a key step for Zero-Touch automation

Towards ‘Automated-ready’ VNF

- Classical path for VNF design



- Required path



- No more hardcoded parameters
- No more ‘hidden’ data
- {{Templating}}
- Model-driven
- Associated tests

Not far from Amsterdam, the Columbus Day !