



OPNFV SUMMIT

JUNE 2017 | BEIJING, CHINA

ONAP and OPNFV collaboration: from infrastructure to machine-learning

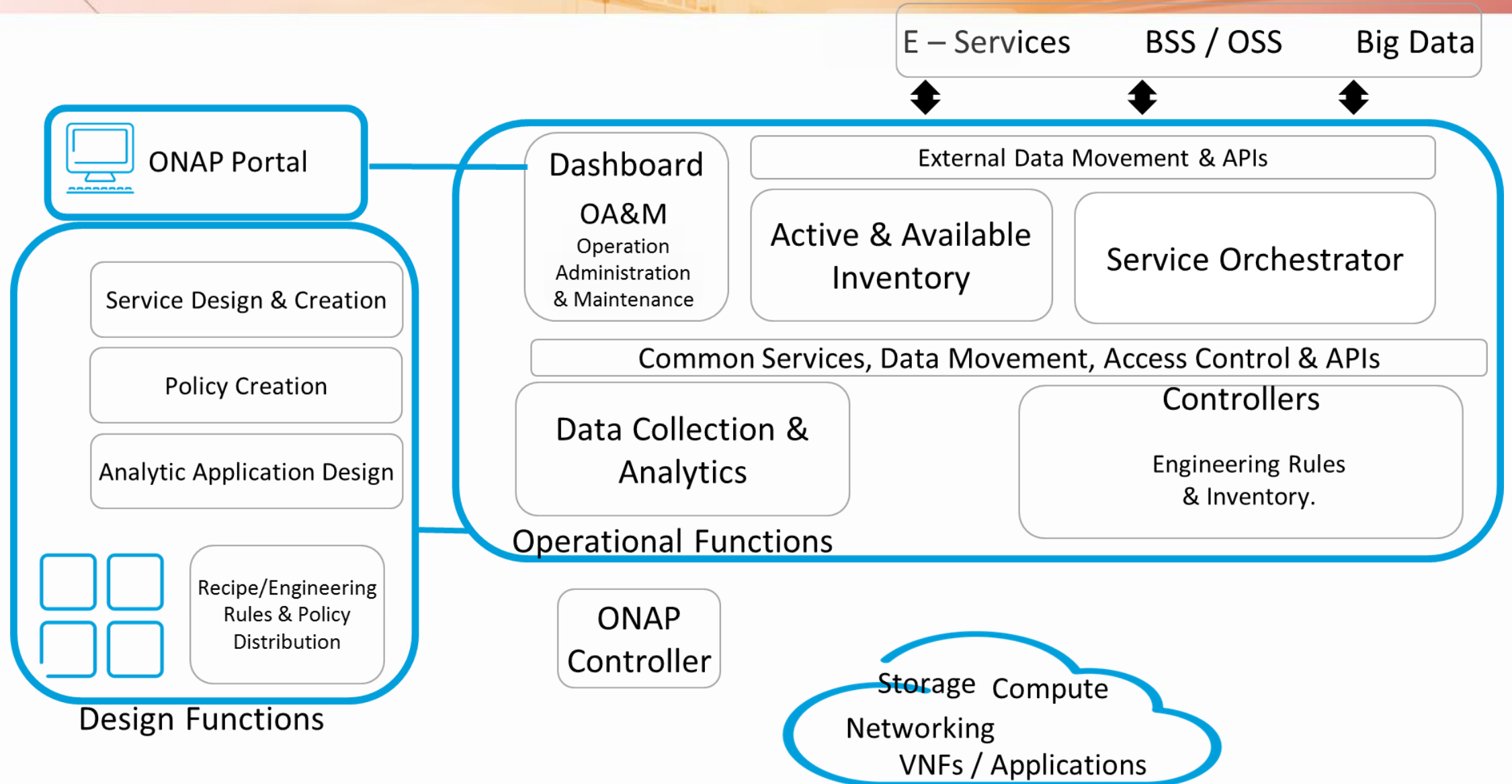
Eric Debeau, Orange
Alassane Samba, Orange

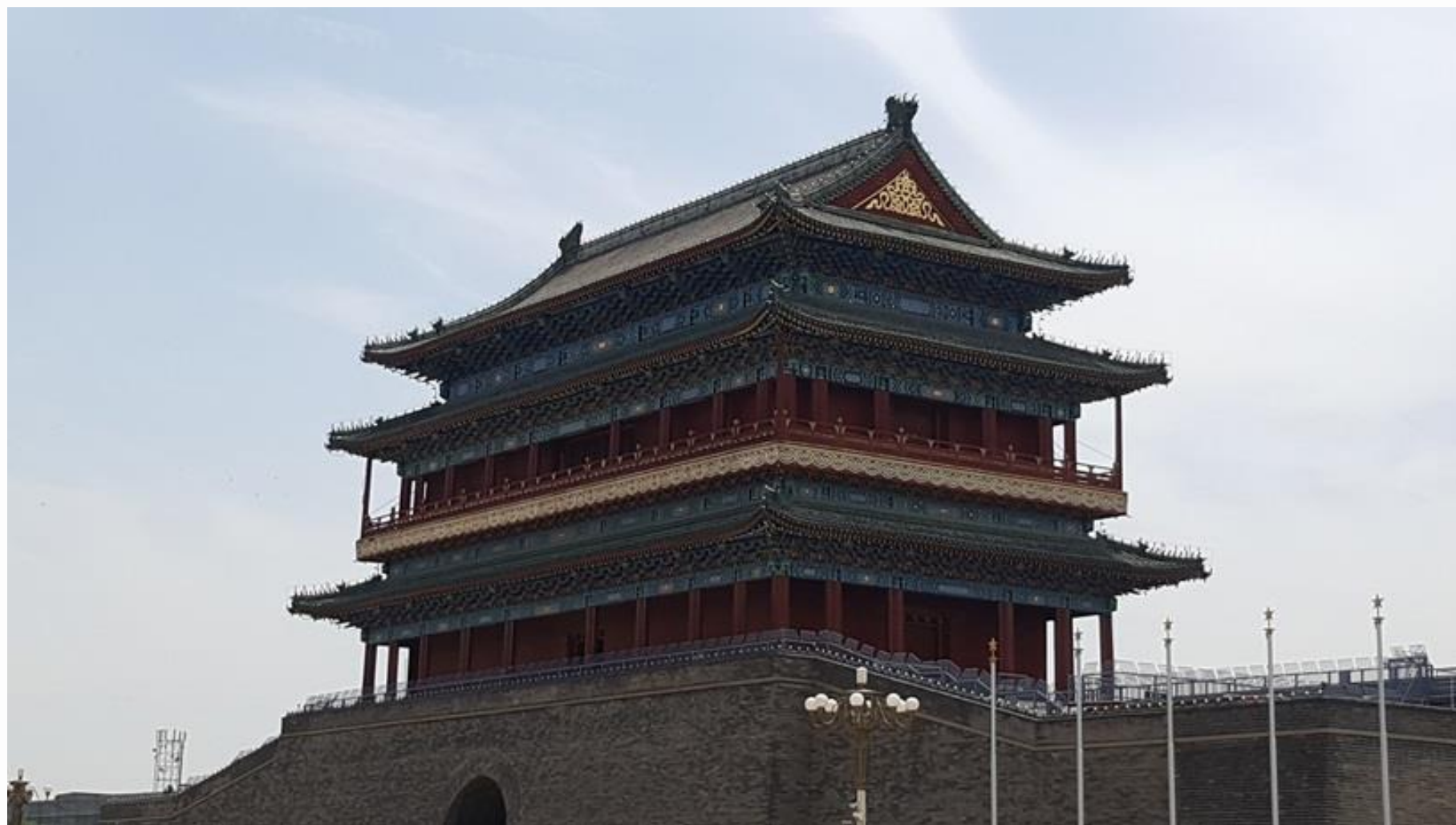
The Orange logo, consisting of a solid orange square above the word "orange" in a white, lowercase, sans-serif font, with a trademark symbol (TM) to the right.

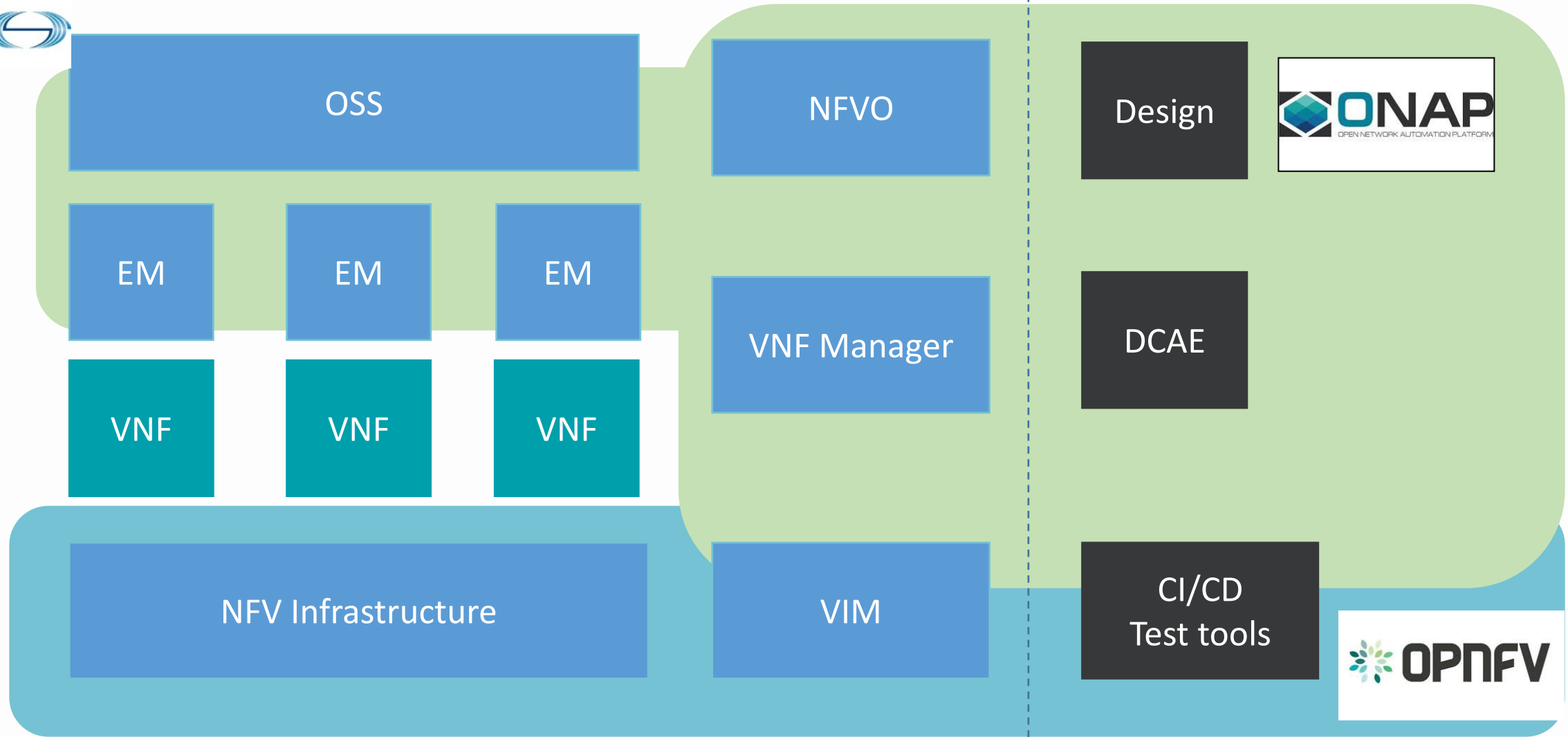
orange™

Agenda

- ONAP context
- Infrastructure
- Reference VNFs
- Certification
- Machine-Learning
- Conclusion

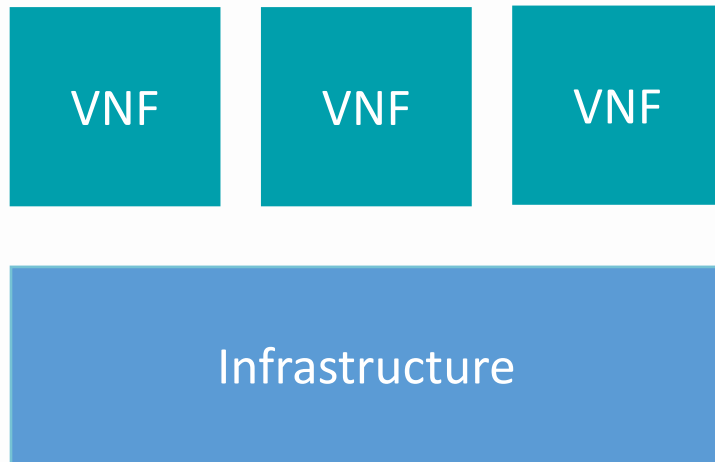






ONAP requires cloud platform for

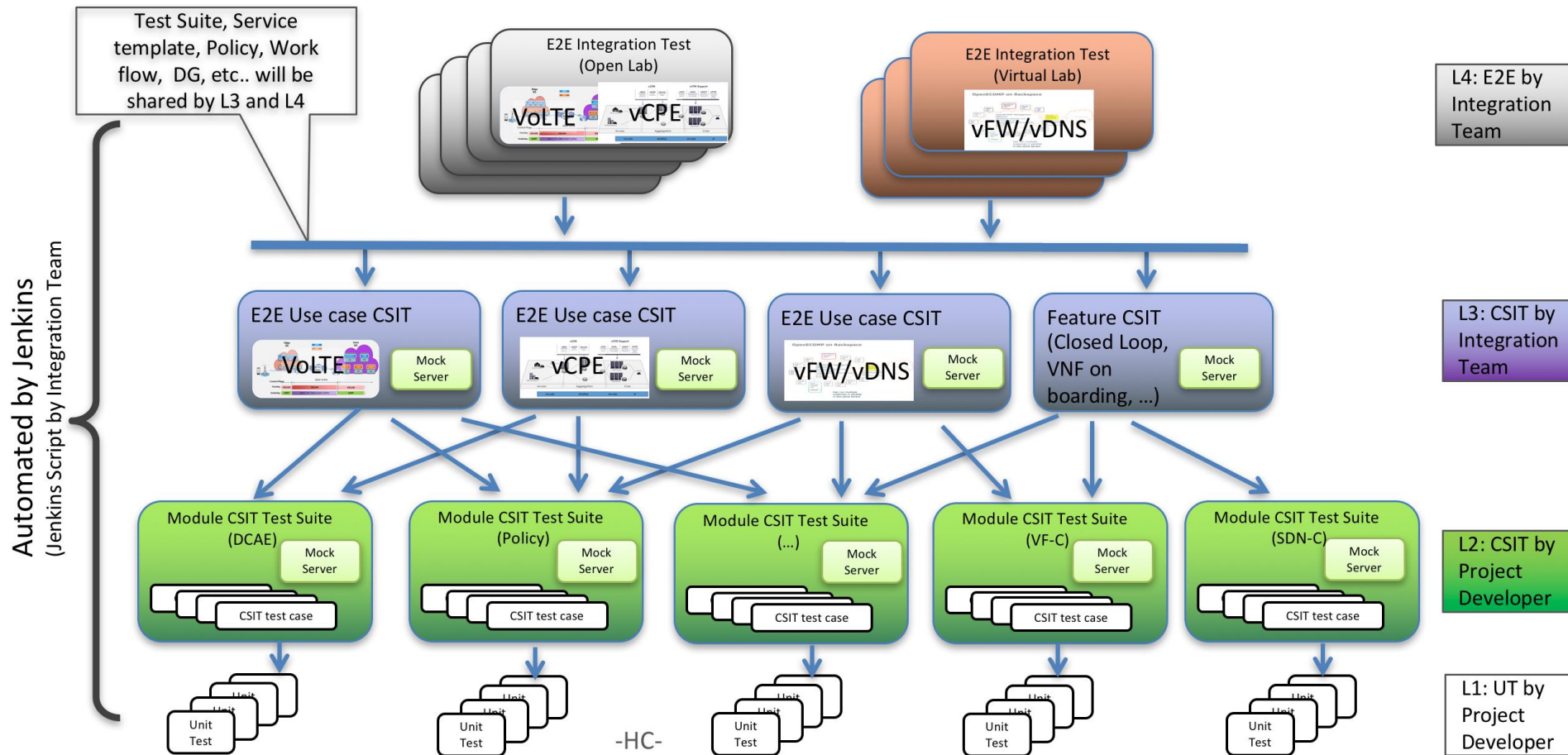
- ONAP components
- VNF deployments



Lab requirements

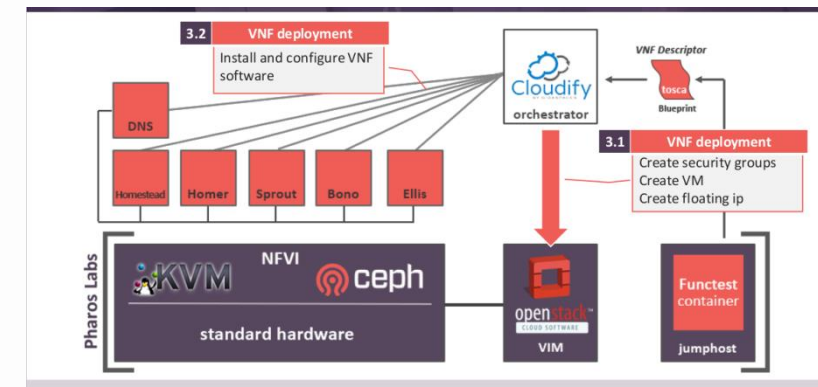
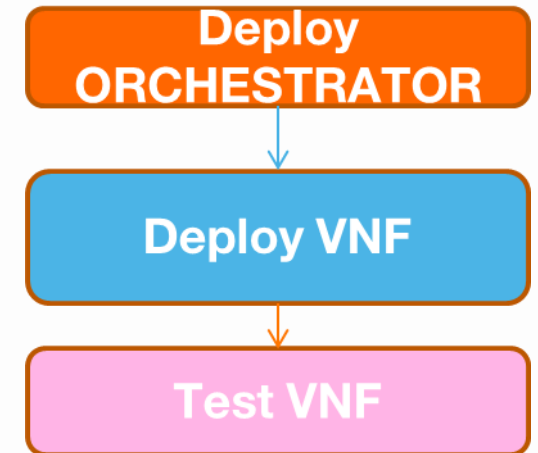
- Development Labs: to support the code management (write code, perform unit tests...)
- Open Labs: to support the needs of consistent, reproducible lab setup for demo and POC purposes.
- Demo labs: to support specific use-cases using vendor VNF or specific network equipment

ONAP 4-Levels CI / CD Architecture



Reference VNFs in OPNFV

- FuncTest project to « verify » infrastructure
 - vIMS Clearwater, openIMS, vRouter Vyatta
 - Reference VNF catalog
 - <https://wiki.opnfv.org/display/DEV/VNF+Catalogue>
- Sample _VNF project to « benchmark » infrastructure
 - CG-NAT (Carrier Grade Network Address Translation) VNF
 - Firewall (vFW) VNF
 - Provider Edge router (vPE) VNF
 - Access Control List (vACL) VNF
 - vEPC-SAE-GW VNF

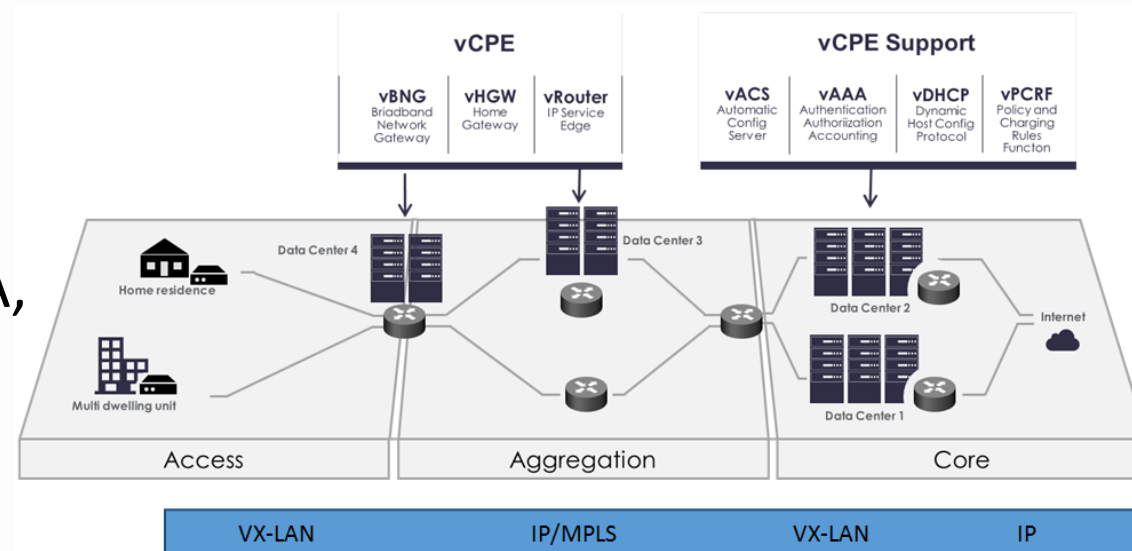


ONAP Reference VNFs

- To be used by any developer (ie open-source)
- To be used to show how the ONAP platform manages VNFs installation and lifetime management.
- To test the platform: verify whether VNFs on-boarding, deployment, and ONAP closed-loop operations.
- To demonstrate and document VNF Requirement compliance
- Existing Reference VNFs
 - vFireWall
 - vLoadBalancer
 - + more to be added

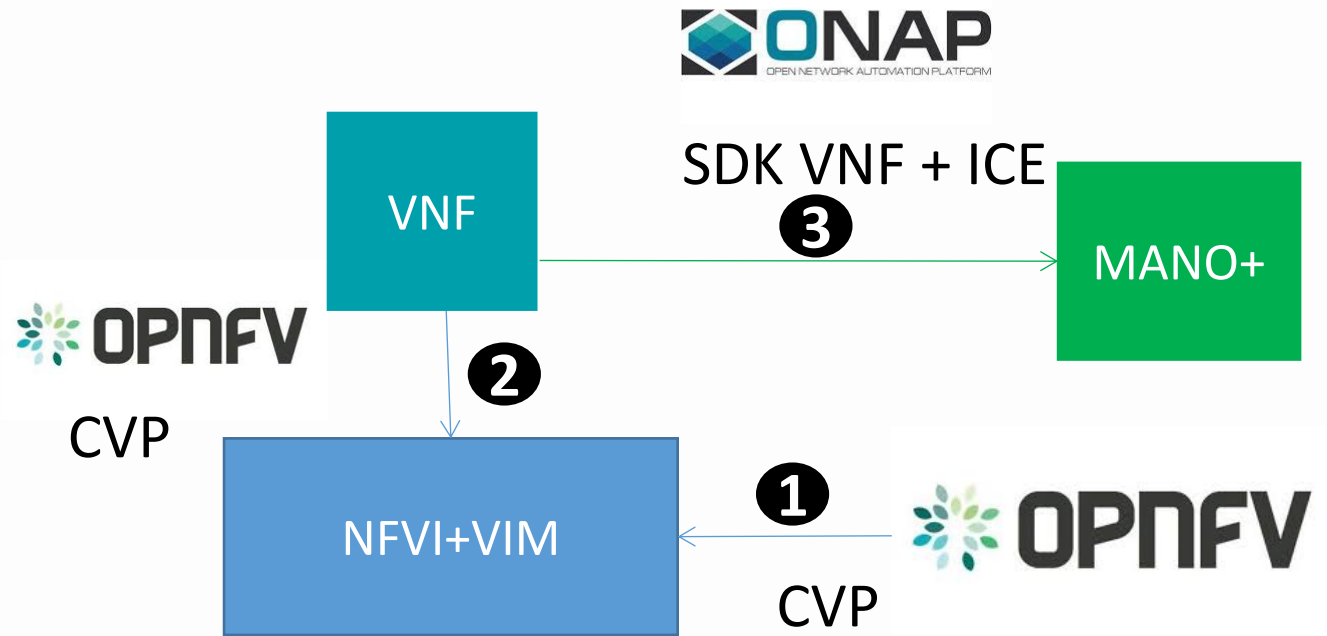
ONAP & OPNFV common interest

- OPNFV: More reference VNF for the OPNFV platform to cover various types of VNF
 - Candidate VNF: vCDN (storage oriented VNF)
- ONAP : More reference VNF for use-case
 - Candidate VNF for use-case VoLTE/vEPC: vIMS, vEPC...
 - Candidate VNF for use-case vCPE: vAAA, vACS, vDHCP, vBNG, vRouter, vHGW
 - May require some adaptations to become ONAP compliant

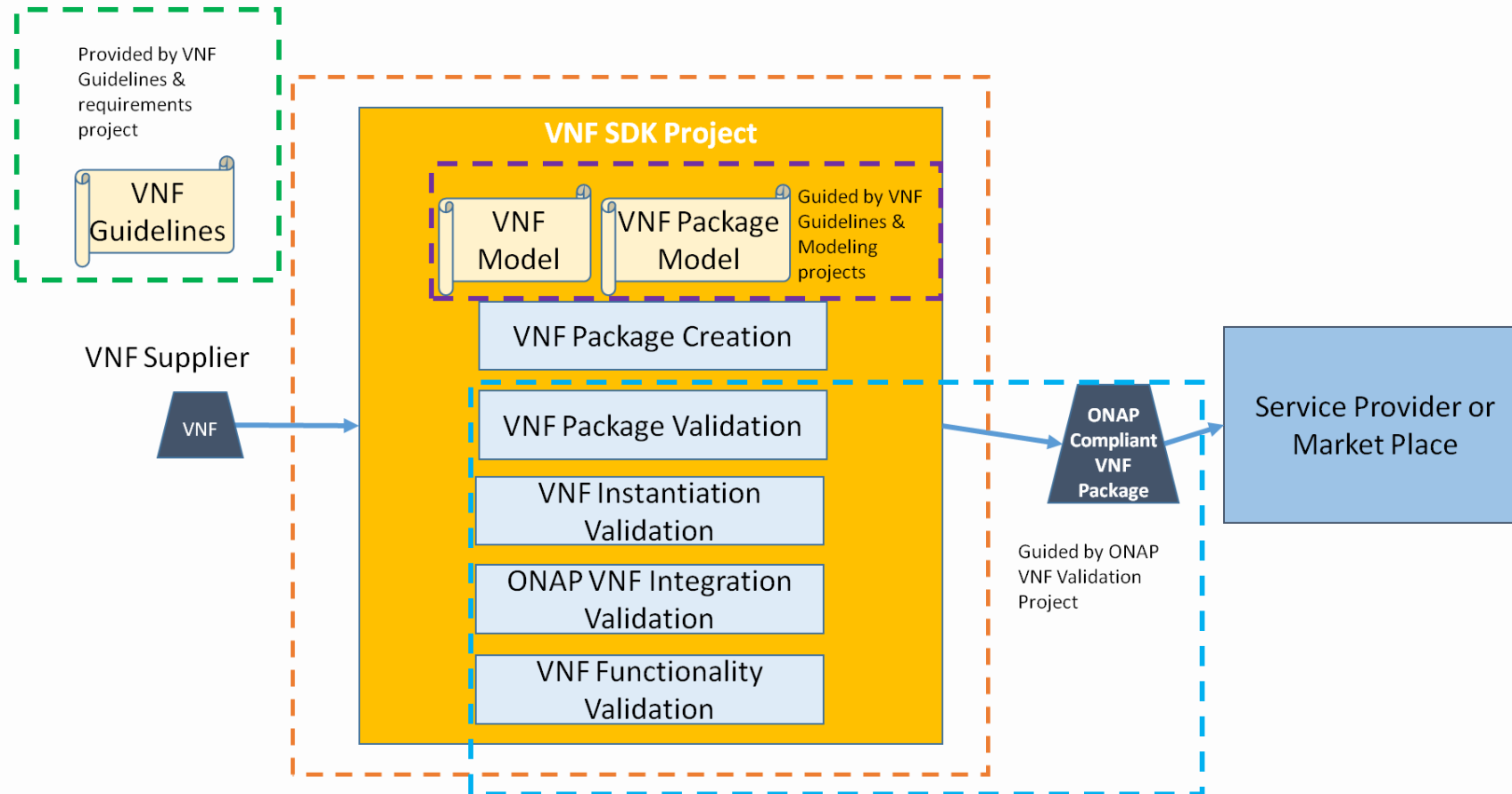


Certification process

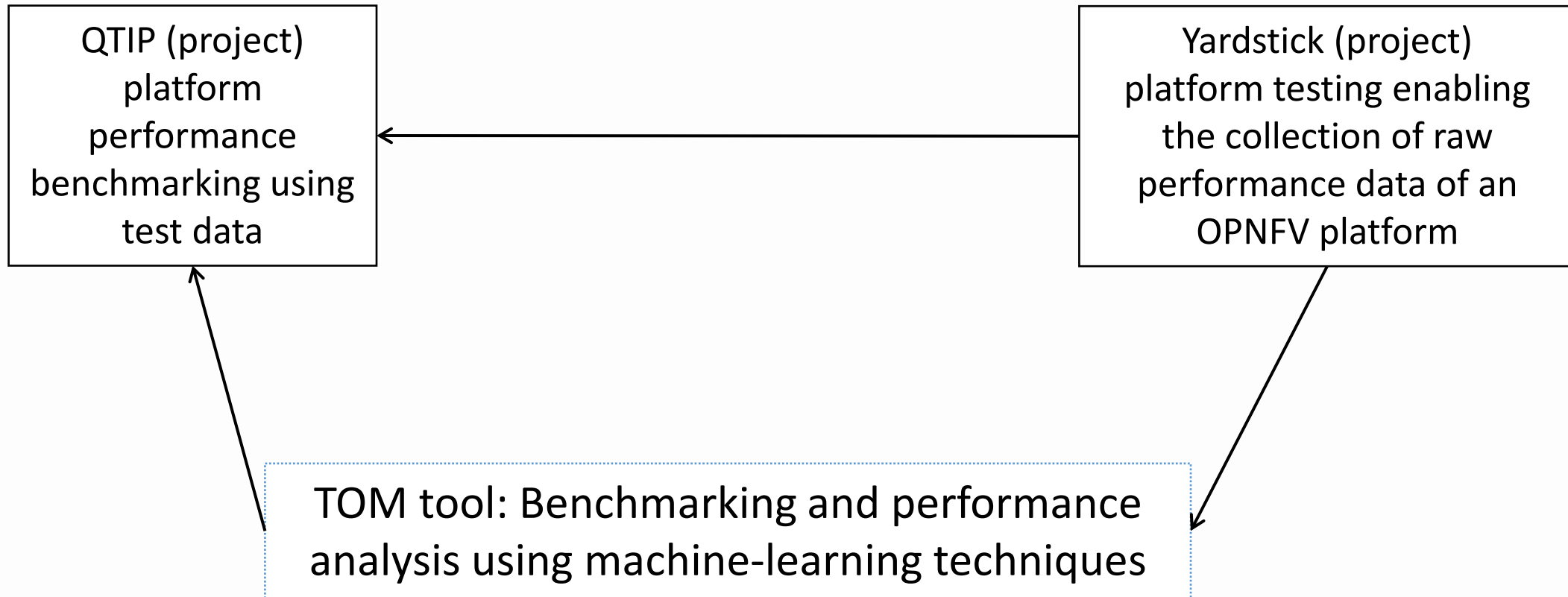
- 1: NFVI/VIM certification
- 2: Functional VNF
- 3: VNF Integration with ONAP



ONAP VNF SDK Project



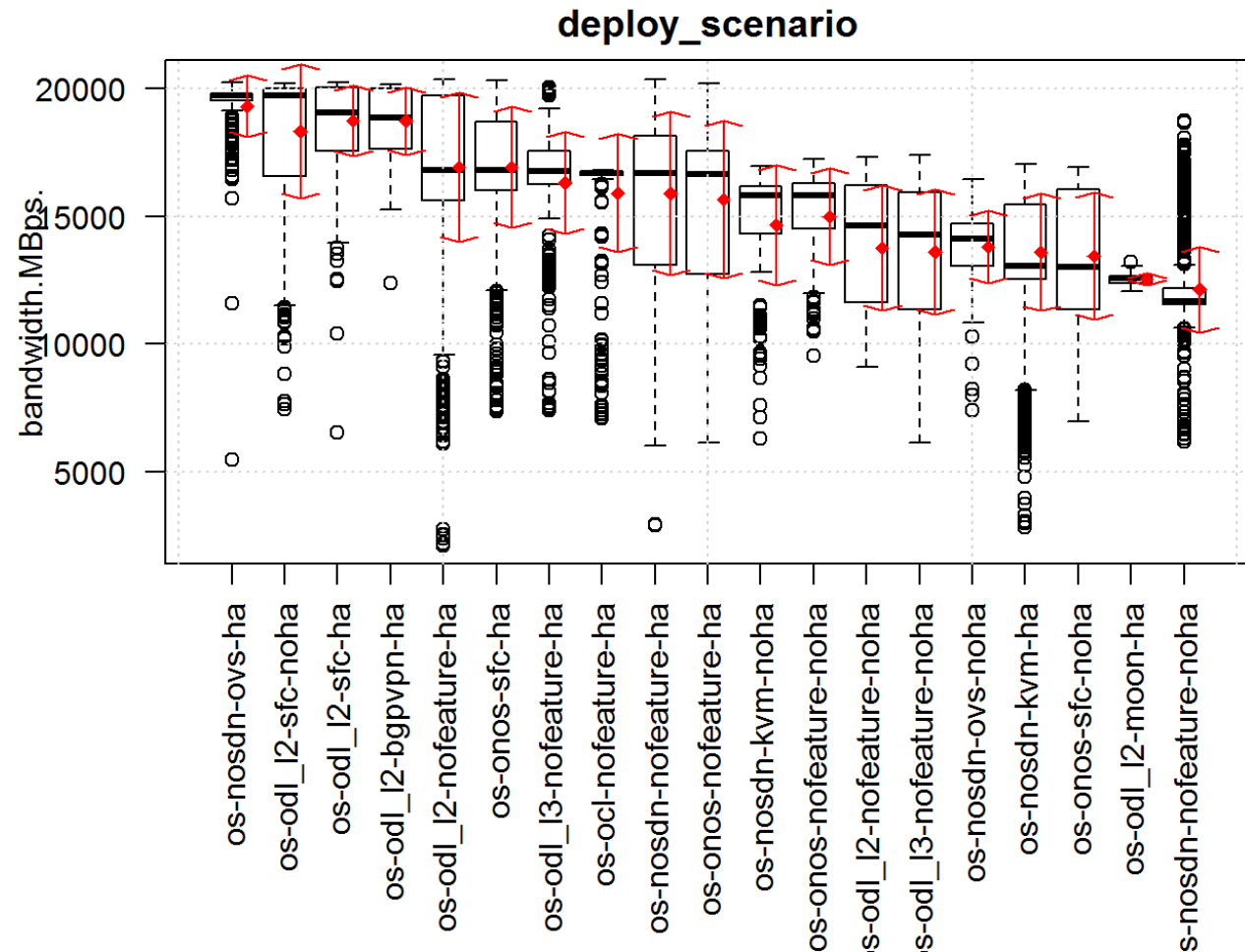
Benchmarking of different test configurations and result analysis



- Components:
 - Benchmarker: configuration comparator
 - rank configurations according to median test result value
 - visualize configuration differences using boxplots
 - Analyzer: determine how far a set of configuration parameters impacts the result of test
 - evaluation of the test result predictability
 - according to each parameter
 - according to a combination of parameters
 - algorithm of to find the best combination
- First version of the code written in R is committed in [QTIP project repo](#)
- Usage example on [OPNFV online wiki page](#)

TOM/Benchmarker

- Rank configurations using one parameter: `deploy_scenario`
- Test case: memory bandwidth



TOM/Analyzer

- How impacting is a context parameter or a combination of context parameter for the result of the test?

- Use of an indicator of influence for the analysis : r^2

$$r^2 = \frac{ESS}{TSS} \quad \text{Where:}$$

$$ESS = \sum_{j=1}^p n_j (\bar{x}^j - \bar{x})^2 : \text{ explained sum of squares}$$

$$TSS = \sum_{i=1}^n (x_i - \bar{x})^2 : \text{ total sum of squares}$$

x_i : the value of the KPI for test $i, i = 1, \dots, n$

\bar{x}^j : the average value of the KPI for the n_j tests performed in context $j, j = 1, \dots, p$

\bar{x} : the average value of the KPI

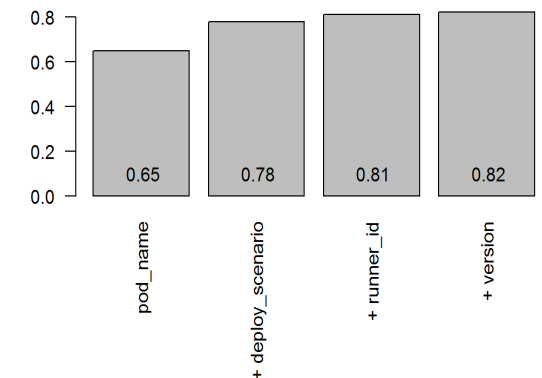
- j is the index of the different *contexts* got from one or a combination of context parameters
- A *context* is described by 1 or several context parameters
- The r^2 is between 0 and 1
- A context parameter or a combination of context parameters is all the more impacting the results as its r^2 is close to 1

TOM API usage example

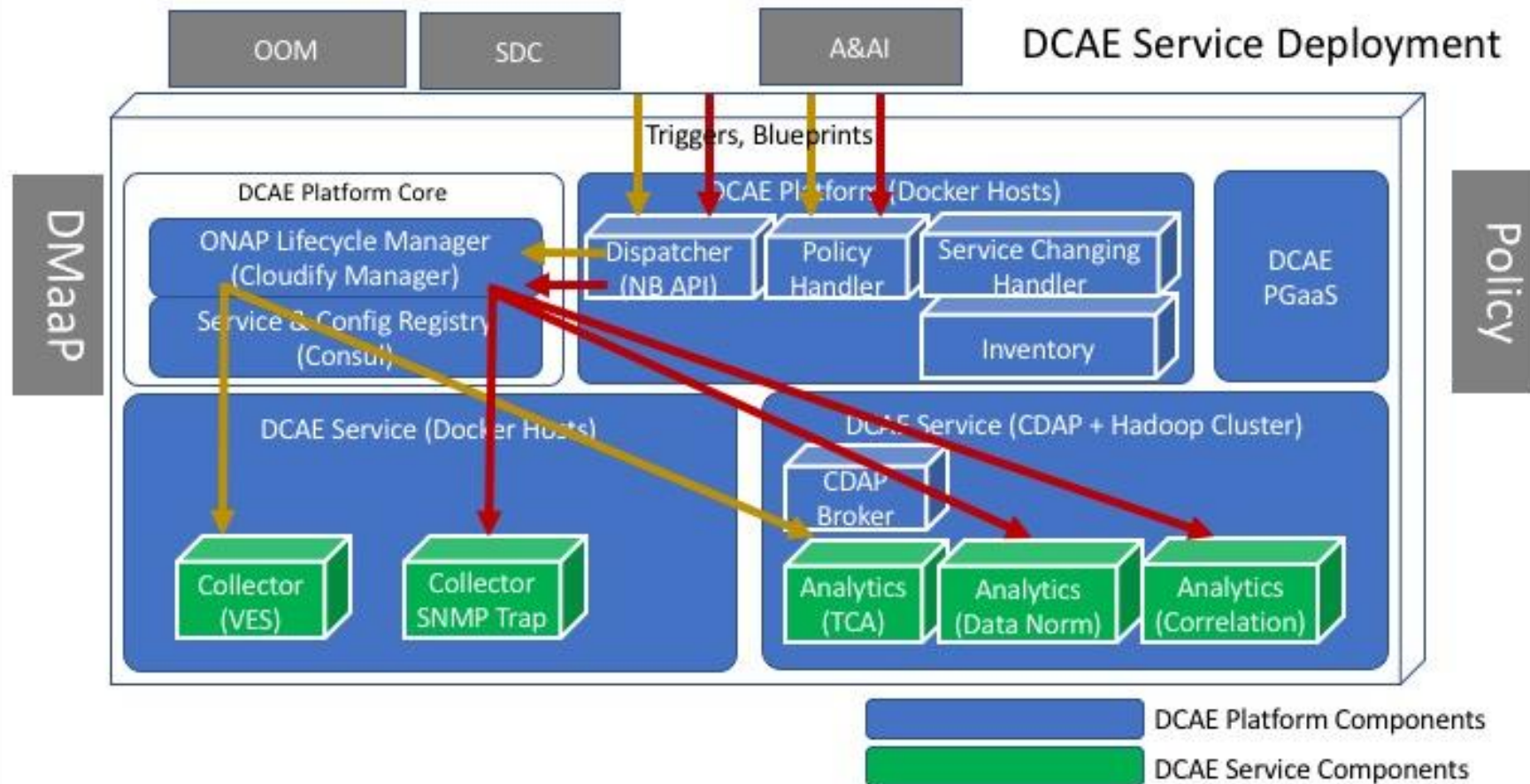
```
curl http://<host>:<port>/read
with arguments, then
curl http://<host>:<port>/analyze
with arguments, then
curl http://<host>:<port>/correlation
outputs:
```

```
{
  "deploy_scenario": [ 0.22 ],
  "version": [ 0.13 ],
  "pod_name": [ 0.65 ],
  "runner_id": [ 0.00 ]
}
```

```
curl http://<host>:<port>/explainGraph
outputs:
```



DCAE Architecture

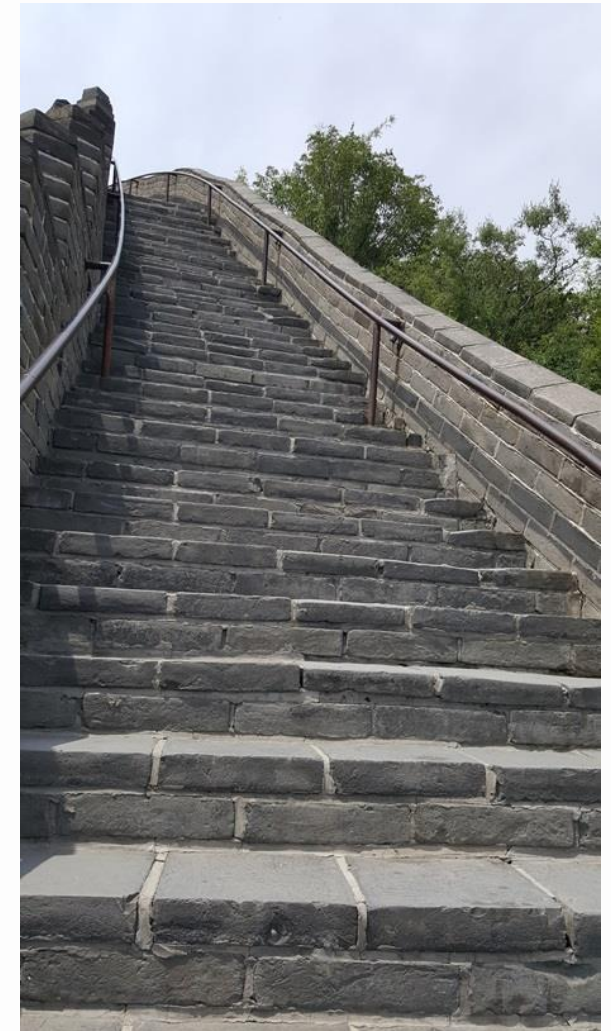


Many topics for collaboration

- Infrastructure
- CI/CD tooling
- Reference VNFs
- Certification process
- Data collection, analytics

And more

- Modeling
- Multi-VIM/Cloud



謝謝

