



ONAP Platform Integration Test Strategy with External Systems

Integration Team

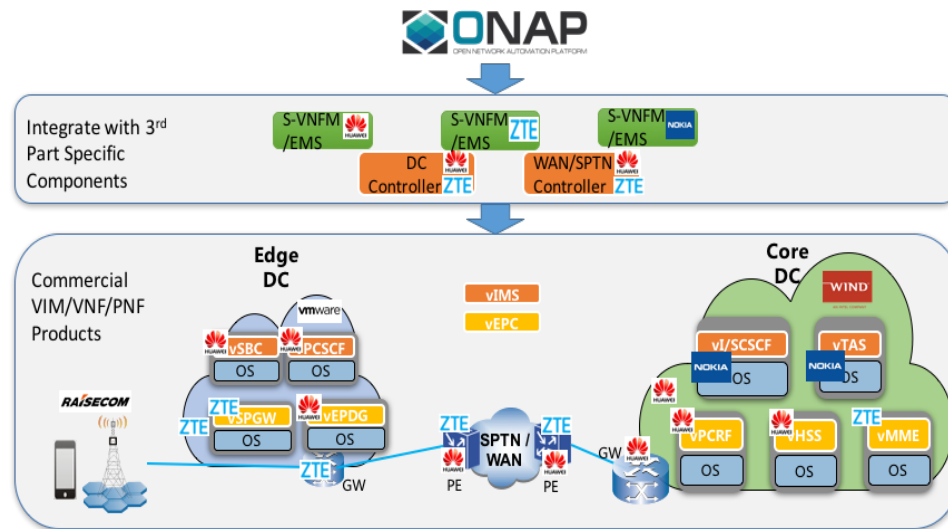
Review R1 VoLTE Use Case Testing with External Systems

The Special for VoLTE case:

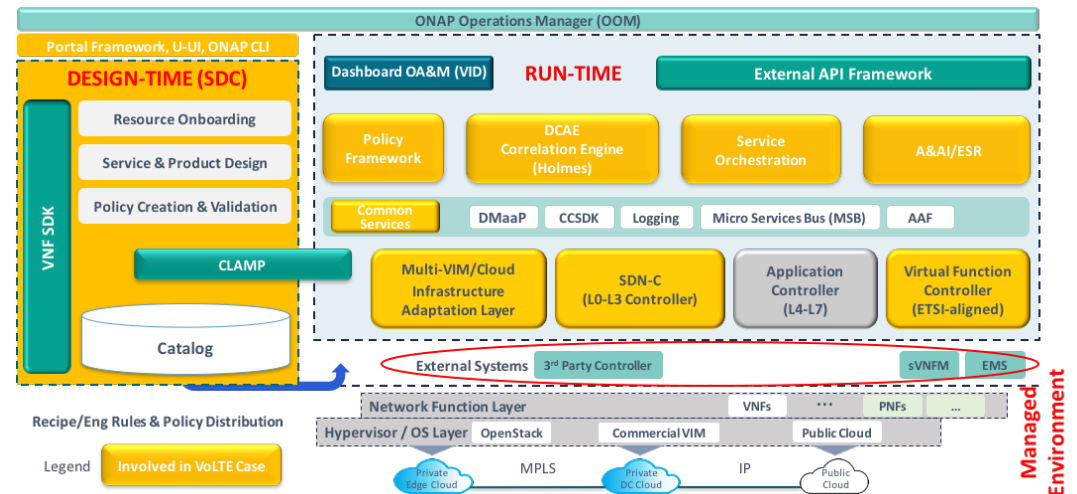
- TOSCA based Models: TOSCA based VNF package imported to SDC
- E2E Service Orchestrator: SDN+NFV
- Integrate with multi-vendors commercial products as 3rd external system, such as VNFMs/EMSs/SDN Controllers/VIMs

How we did VoLTE use case E2E test in R1:

- In integration lab tested features not involved with external systems, other features could only be partially tested due to lack of external systems
- In CMCC physical lab, focused on testing e2e features with the external systems
- All tests were done manually



ONAP Architecture for Amsterdam



VoLTE Integration Test Strategy for R2+

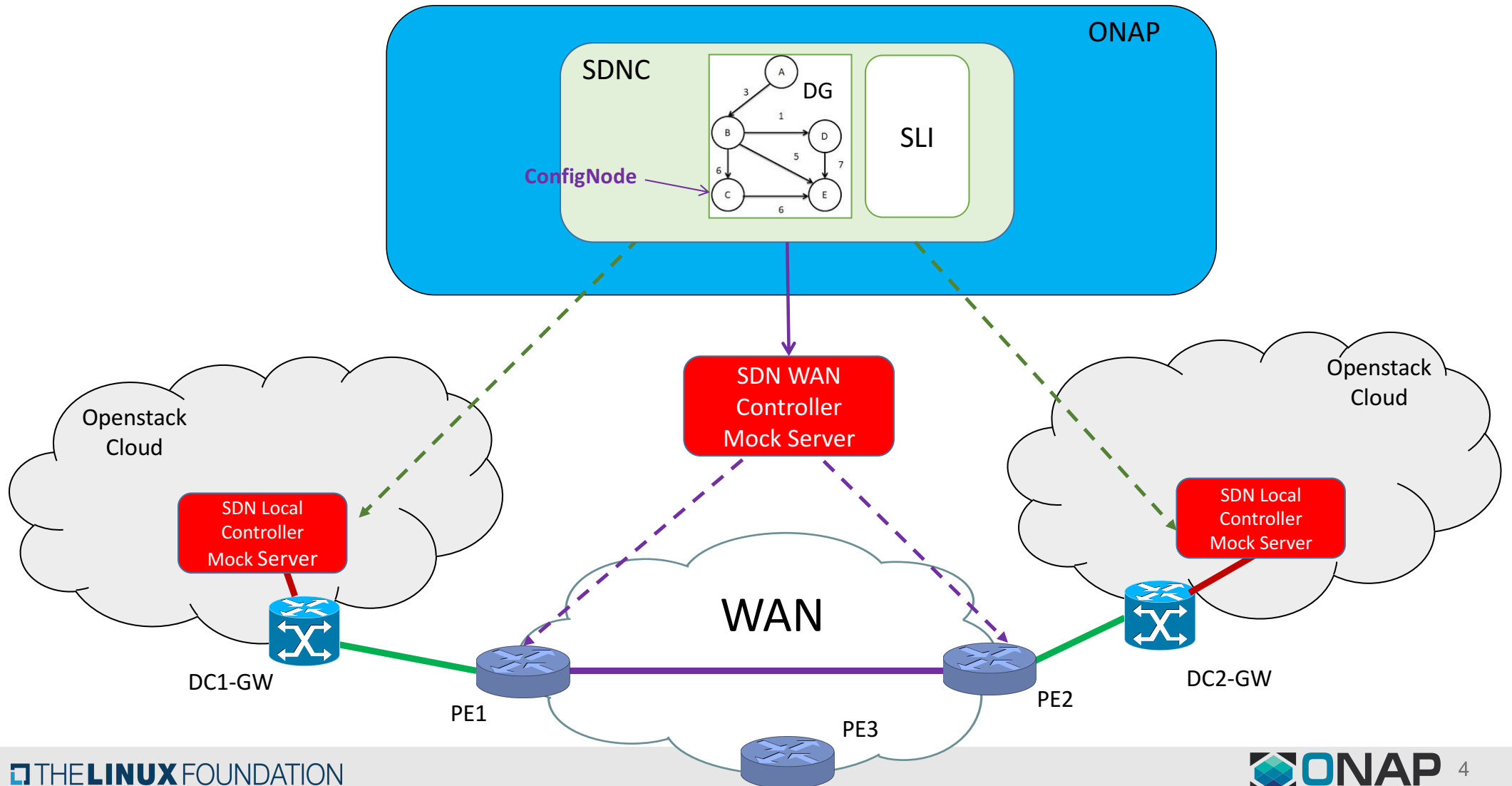
Plan for R2+ VoLTE Use Case Testing:

- Step1: Implement EMS/VNFM/SDN Controller mock servers
- Step2: Automate feature/use case level test cases with mock servers
- Step3: Hook up automated tests with mock servers into CI/CD process
- Step4: Run E2E test automatically in physical lab with real external systems

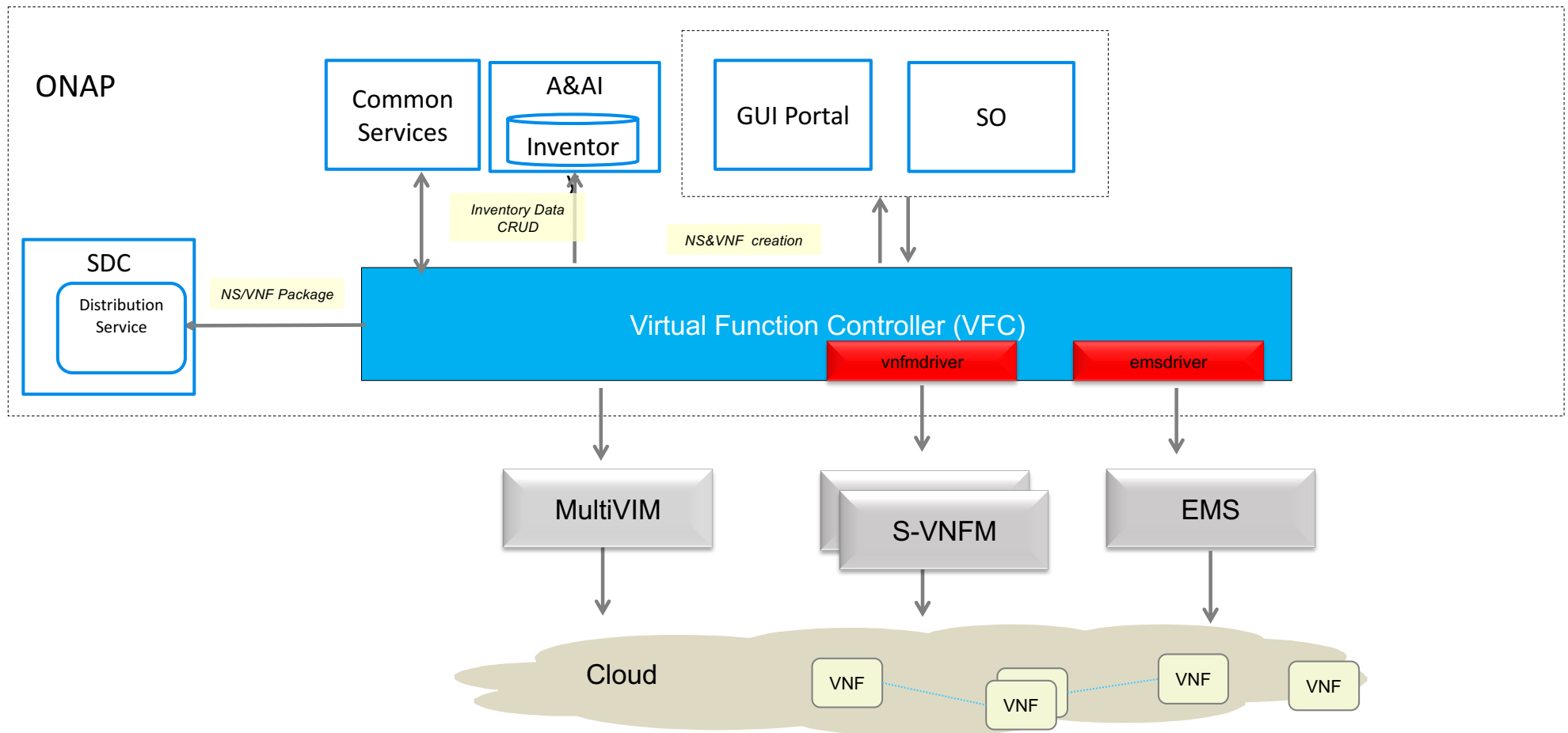
General Suggestions:

- When a new external system is introduced by use case, the external system mock server should be provided before integration test starts
- Add/Enhance automated e2e use case tests with mock server to CI/CD

Mock SDN Controllers in VoLTE

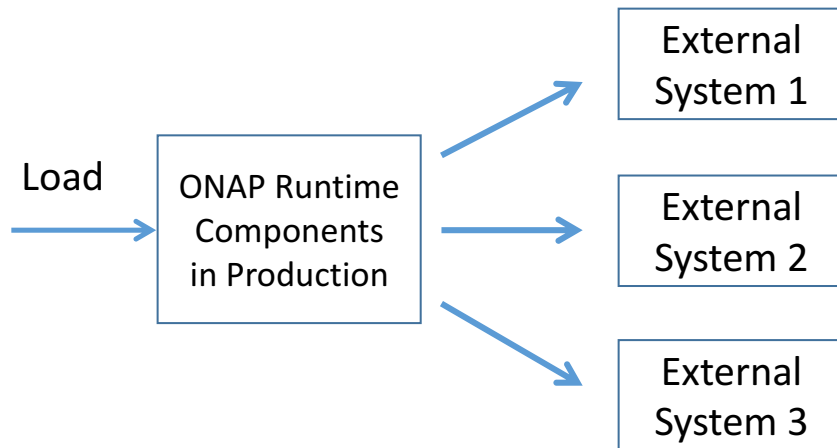


Mock ONAP Drivers in VoLTE

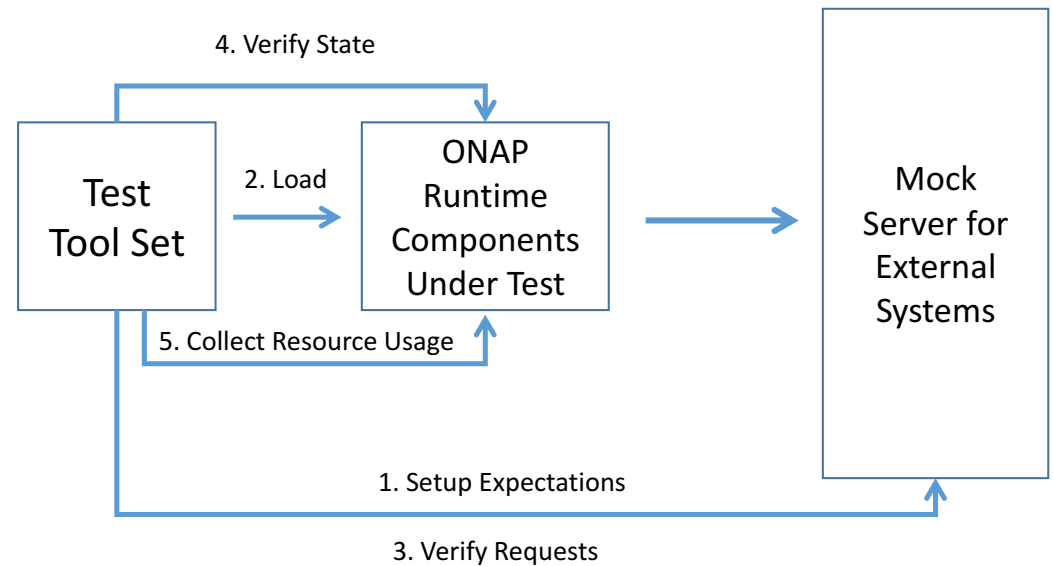


ONAP Platform S3P Test Methodology with External Systems

A ONAP runtime components with external system dependencies



MockServer is used to mock 3rd party external system dependencies



Tools Used



Locust – Load Generator

MockServer

MockServer – Simulate External Systems

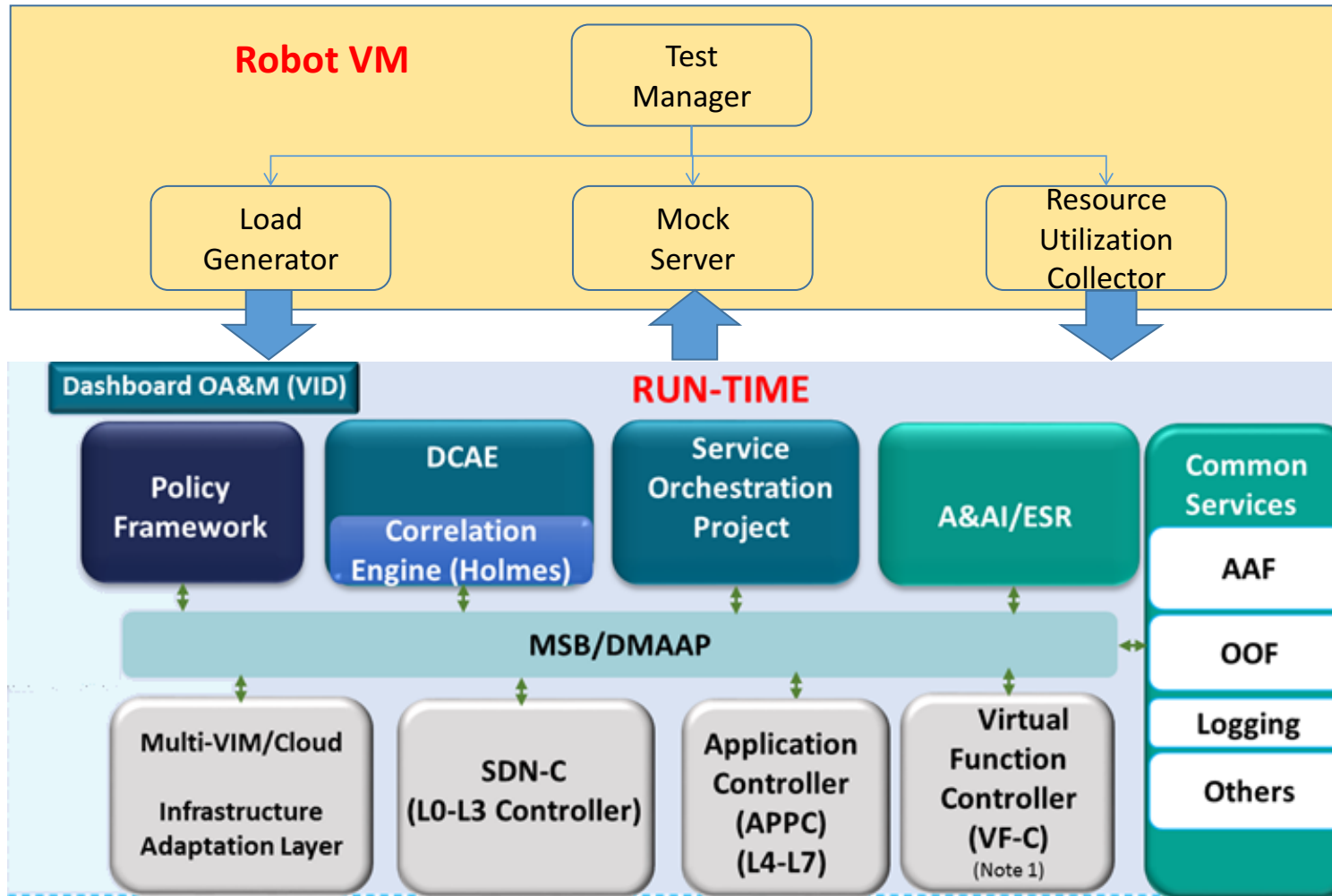


Wireshark – Packet Trace



Scripts – Create Tests & Collect Data

Testbed Setup



Register MockServer as ONAP ESR SDN Controllers

The screenshot shows the ONAP Portal interface with a 'Register sdn info' dialog box open. The dialog box contains the following fields:

- Name: IP-WAN-Controller
- Uri: http://10.0.10.1:1080 (highlighted with a red circle)
- Username: zmb@huawei.com
- Password: [masked]
- Version: Agile Controller-WAN V300R002C10SPC015T
- Vendor: HUAWEI
- Protocol: RESTCONF
- Product Name: AC-WAN
- Type: WAN

Buttons for 'save' and 'cancel' are located at the bottom right of the dialog box.

Changes to Make on ONAP MSB

ONAP MicroService Bus

API Service IUI Service Custom Service

80 HTTP / 443 HTTPS 34

Edit API Service

Source Service

Host	No	IP	PORT
	1	10.0.10.1	1080

URL * /api/huaweivnfmdriver/v1

Published as

Published URL http://openo/api/huaweivnfmdriver/v1

Name * huaweivnfmdriver

Version v1

+ API Define /api/huaweivnfmdriver/ customInput

Cancel Save

huaweivnfmdriver-Service Detail

fetching resource list: http://openo/apijson/huaweivnfmdriver/v1

image2018-2-8_15-52...png GMT20180320-1305...mp4 volte_sdnc.log service-VolteE2eServi...yaml resource-Testpnf-tem...yaml resource-Testpnf-csar.csar Show All x

Locust and Swarm of Locusts

```
Class UserBehavior(TaskSet):  
    @task(1)  
    def test(self):  
        test_create_service_instance()  
        test_closed_loop()  
        test_scale_out()  
        test_scale_in()  
        test_delete_service_instance()  
  
... ..  
  
Class User(HttpLocust):  
    task_set = UserBehavior  
    min_wait = 1000  
    max_wait = 3000
```



Start new Locust swarm

Number of users to simulate

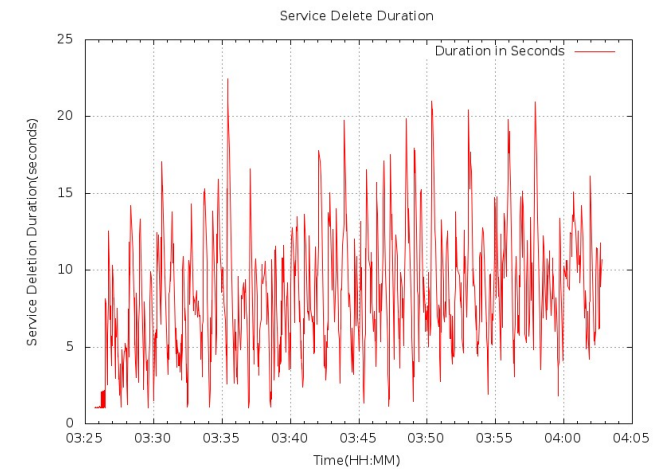
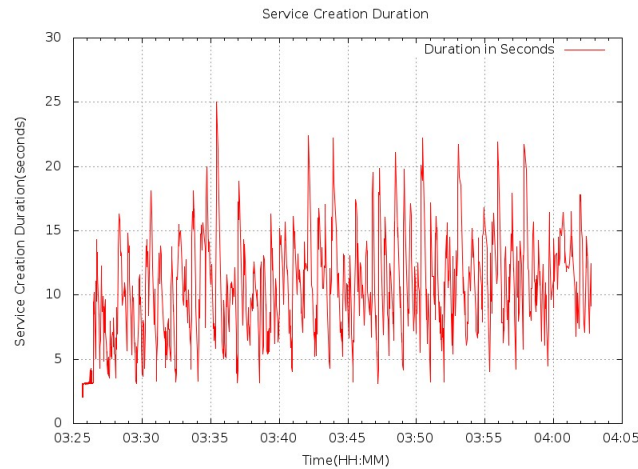
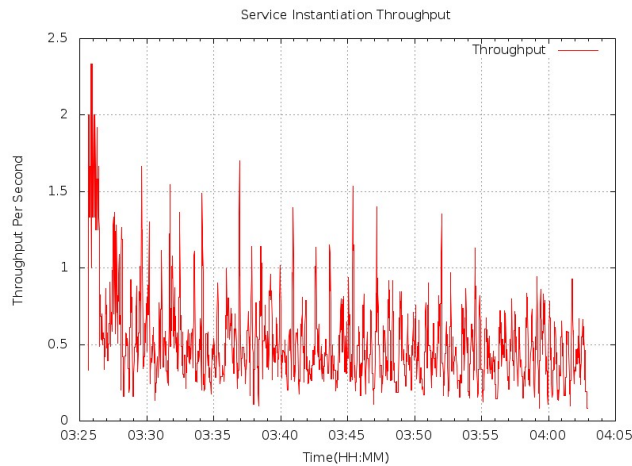
Hatch rate (users spawned/second)

Start swarming



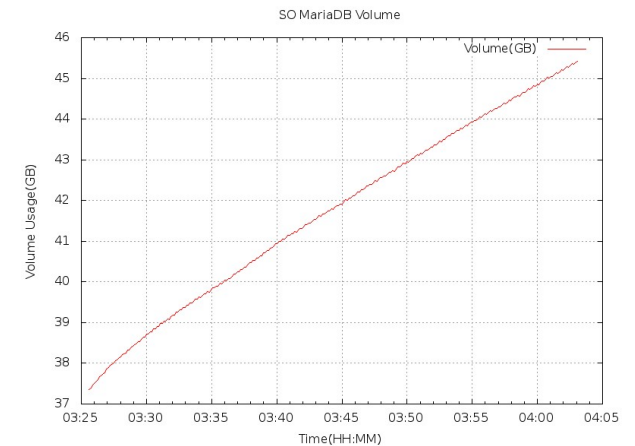
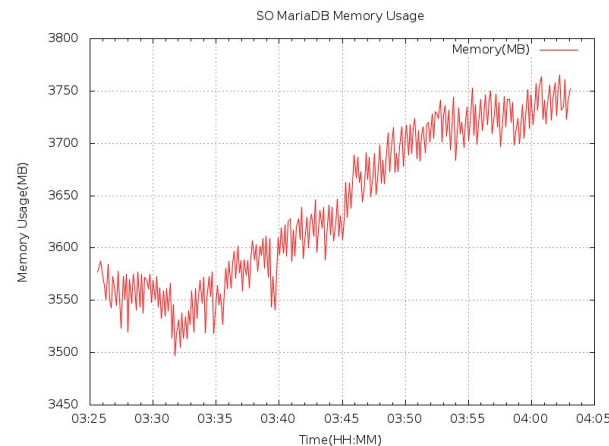
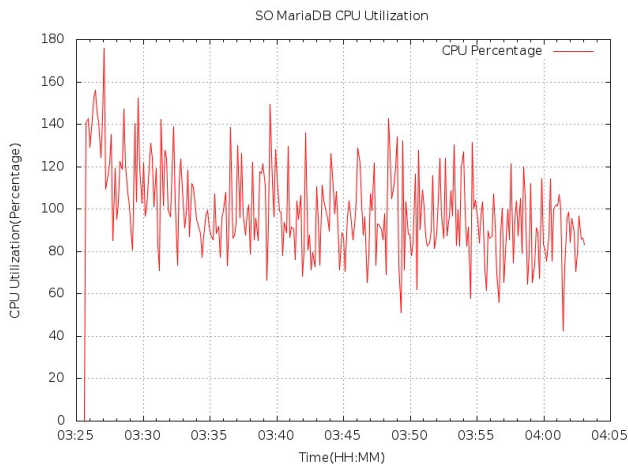
Collected Throughput and Latency Data

```
reports — ubuntu@onap-robot: ~/stats/plot — -bash — 111x24
Yangs-MacBook-Air:reports yang$ head operation.log
{"datetime": "2018-03-17T03:25:39UTC", "operation": "volte_create", "result": "success", "duration": 2.059}
{"datetime": "2018-03-17T03:25:38UTC", "operation": "volte_create", "result": "success", "duration": 3.12}
{"datetime": "2018-03-17T03:25:40UTC", "operation": "volte_create", "result": "success", "duration": 2.07}
{"datetime": "2018-03-17T03:25:41UTC", "operation": "volte_delete", "result": "success", "duration": 1.04}
{"datetime": "2018-03-17T03:25:39UTC", "operation": "volte_create", "result": "success", "duration": 3.076}
{"datetime": "2018-03-17T03:25:42UTC", "operation": "volte_delete", "result": "success", "duration": 1.038}
{"datetime": "2018-03-17T03:25:42UTC", "operation": "volte_delete", "result": "success", "duration": 1.035}
{"datetime": "2018-03-17T03:25:41UTC", "operation": "volte_create", "result": "success", "duration": 2.062}
{"datetime": "2018-03-17T03:25:40UTC", "operation": "volte_create", "result": "success", "duration": 3.074}
{"datetime": "2018-03-17T03:25:43UTC", "operation": "volte_delete", "result": "success", "duration": 1.038}
```

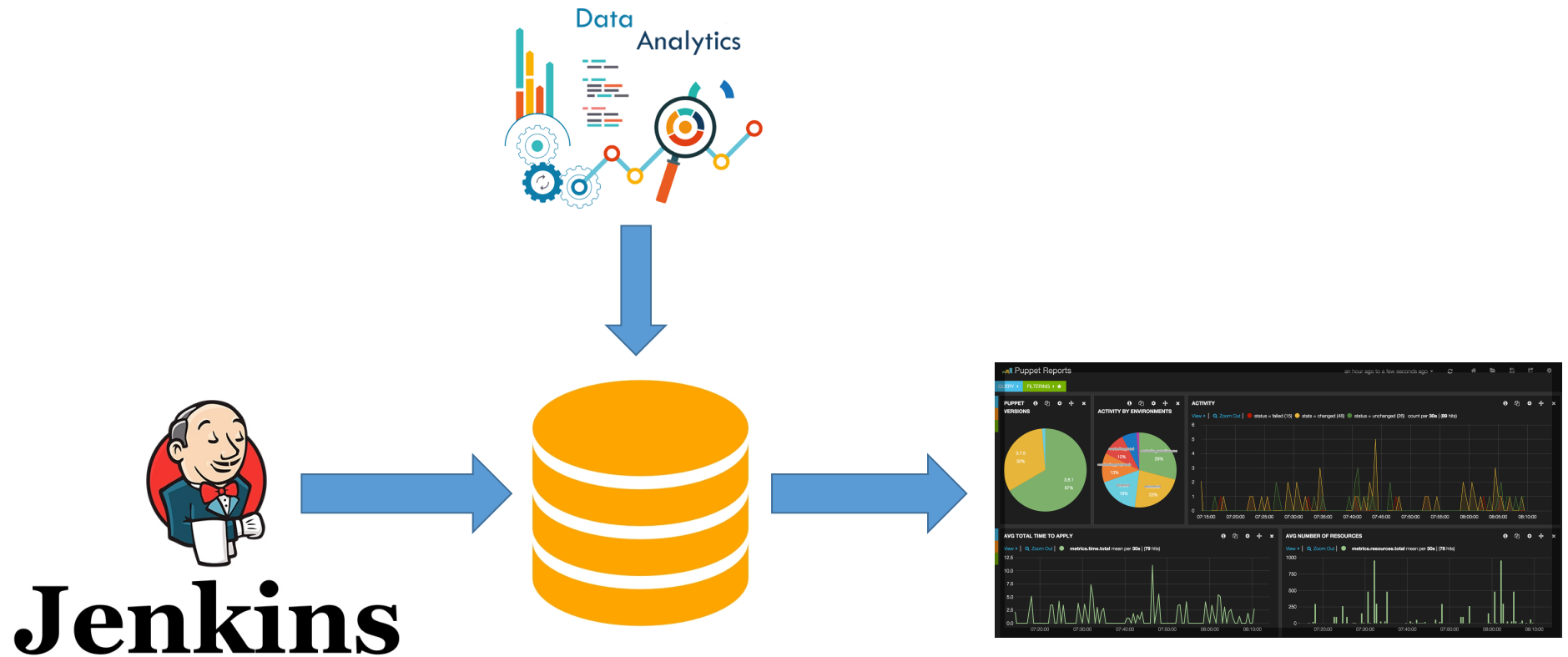


Collected Resource Usage Data

```
reports — ubuntu@onap-robot: ~/stats/plot — -bash — 184x24
Yangs-MacBook-Air:reports yang$ head resource.log
{"datetime": "2018-03-17T03:25:36UTC", "component": "SO", "container": "testlab_mso_1", "cpu": 1.0, "memory": 3237.0, "physical": 7796.0, "mem_percent": 41.52}
{"datetime": "2018-03-17T03:25:36UTC", "component": "SO", "container": "testlab_mariadb_1", "cpu": 0.55, "memory": 3577.0, "physical": 7796.0, "mem_percent": 45.88, "volume": 37.35}
{"datetime": "2018-03-17T03:25:39UTC", "component": "SDNC", "container": "sdnc_ueblistener_container", "cpu": 0.05, "memory": 476.0, "physical": 7798.0, "mem_percent": 5.96}
{"datetime": "2018-03-17T03:25:39UTC", "component": "SDNC", "container": "sdnc_dmaaplistener_container", "cpu": 0.04, "memory": 215.0, "physical": 7798.0, "mem_percent": 2.69}
{"datetime": "2018-03-17T03:25:40UTC", "component": "SDNC", "container": "sdnc_dgbuilder_container", "cpu": 0.0, "memory": 50.5, "physical": 7798.0, "mem_percent": 0.63}
{"datetime": "2018-03-17T03:25:40UTC", "component": "SDNC", "container": "sdnc_portal_container", "cpu": 0.0, "memory": 45.2, "physical": 7798.0, "mem_percent": 0.57}
{"datetime": "2018-03-17T03:25:40UTC", "component": "SDNC", "container": "sdnc_controller_container", "cpu": 1.55, "memory": 5515.0, "physical": 7798.0, "mem_percent": 70.73}
{"datetime": "2018-03-17T03:25:41UTC", "component": "SDNC", "container": "sdnc_db_container", "cpu": 0.03, "memory": 507.9, "physical": 7798.0, "mem_percent": 6.36, "volume": 0.2}
{"datetime": "2018-03-17T03:25:44UTC", "component": "SO", "container": "testlab_mso_1", "cpu": 97.85, "memory": 3239.0, "physical": 7796.0, "mem_percent": 41.55}
{"datetime": "2018-03-17T03:25:44UTC", "component": "SO", "container": "testlab_mariadb_1", "cpu": 140.97, "memory": 3582.0, "physical": 7796.0, "mem_percent": 45.94, "volume": 37.4}
```



Next Step: Automate Platform Non-Functional Tests In CI/CD





Q&A

Call for contributors

Please contact with integration team if you are interested in