Critical Design Review

Containerized, Intelligent Network Functions on Hosts

Project Instructor: Dr. Kevin Gifford
Project Advisor: Dr. Levi Perigo

Team 06:

Afure Martha Oyibo
Kiran Yeshwanth
Manesh Yadav
Prarthana Shedge
Soumya Velamala

Agenda

Topic	Presenter	Slide No.
Problem with Traditional Networking	Soumya	3
Benefits of SDN-NFV using Containers	Soumya	4
Project Objectives	Soumya	5
FBD and CONOPS	Manesh	6 - 7
Levels of Success	Manesh	8
Critical Project Elements	Manesh	9 - 11
Design Requirements	Manesh	12 - 17
Baseline Design Summary	Kiran	18 - 21
Verification & Validation	Kiran	22 - 34
Risk Factors & their Impact	Prarthana	35 – 41
Project Risks	Prarthana	42 - 44
Revised CONOPS	Soumya	45
Organizational Chart	Soumya	46
Work Breakdown Structure	Soumya	47
Cost Plan	Martha	48
Hardware Requirements	Martha	49
Project Targets & Key Deadlines	Martha	50

Problem – Traditional Networking

Difficult to deploy and scale

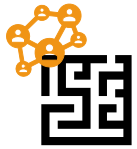
Lacks flexibility

Large CAPEX and OPEX

Failures and packet losses

Administrative overhead

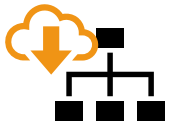
Error-prone



Network complexity

Short product life cycle

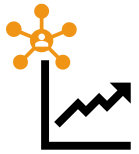
Benefits- SDN NFV using Containers



Quick deployment

Flexibility and scalability

Reduced CAPEX and OPEX



Proactive failure detection

Reduced kernel & administrative overhead

NSOT

Simplified network architecture

Timely updates

Project Objectives

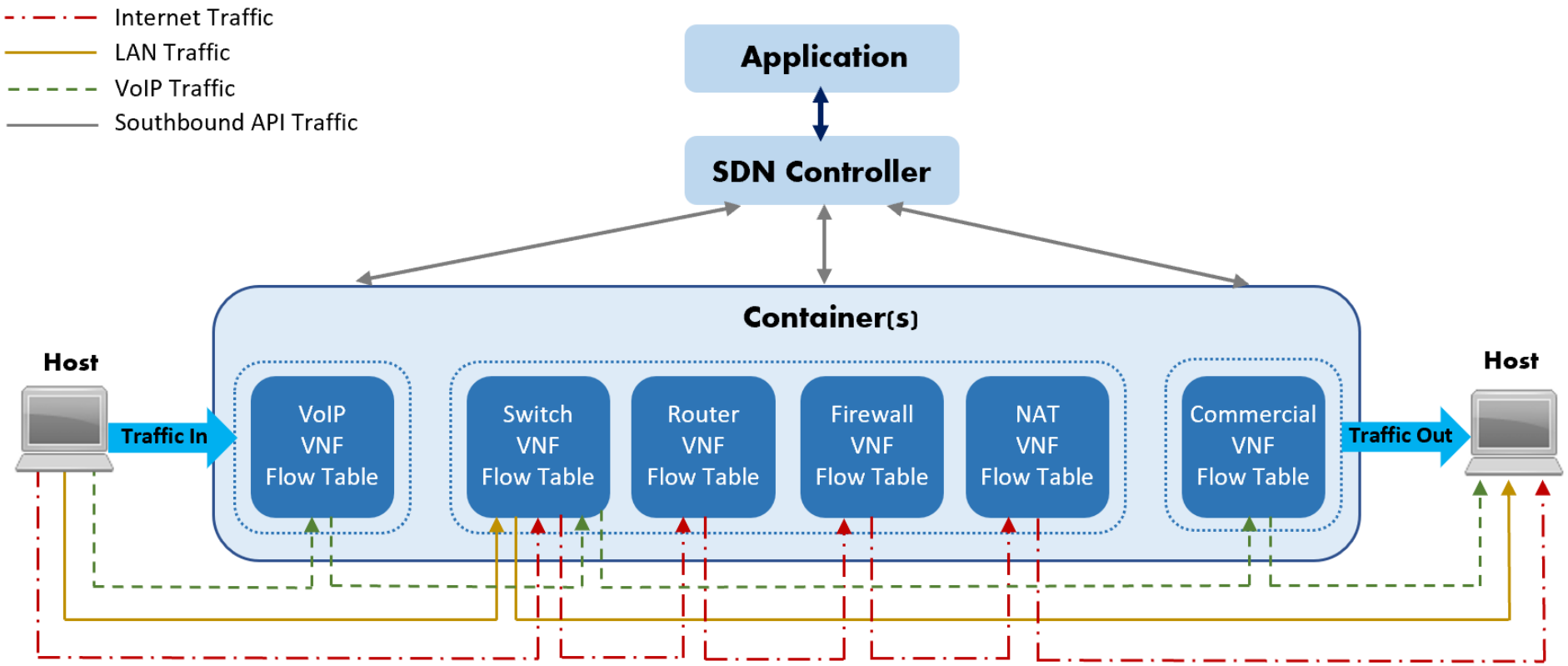
Objective

- Create a low-cost solution for the deployment of SDN based network services such as routers, firewalls, VoIP, etc. on the host devices through NFV:
  - NFV based services will be deployed through container based VNFs
  - Enhanced performance by bringing SDN based intelligence closer to the end hosts
  - Rapid deployment of network services

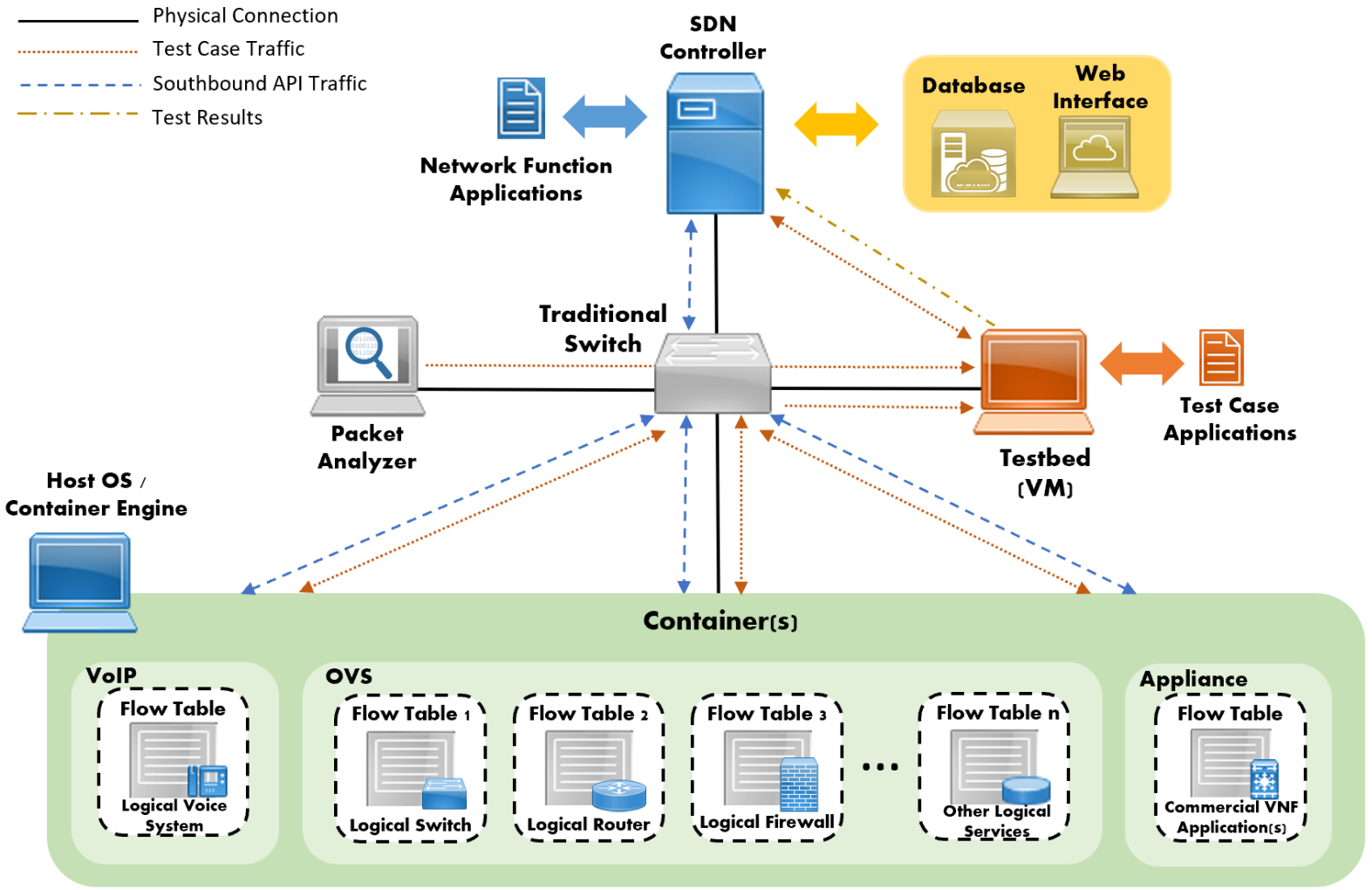
Stretch Goal

- Detection of network failovers and implementation of redundant solutions:
  - Reduces the risk of packet losses
  - Seamless end user experience

Functional Block Diagram (FBD)



Concept of Operations (CONOPS)



Levels of Success

Level 1: Configuration Tests

- Test the deployment of VNFs
- Test the service chain creation
- Test the reachability between network devices

Level 2: Performance Tests

- Test network performance for different traffic types
- Distinct service types, transmission rates and packet sizes
- Evaluate QoS and throughput

Level 3: Failover Tests

- Test the network redundancy
- Device failure, link failures, network congestion, security breach
- Test the network's ability to detect and allocate redundant resources

Critical Project Elements - Technical

CPE 1.1: VNF Creation

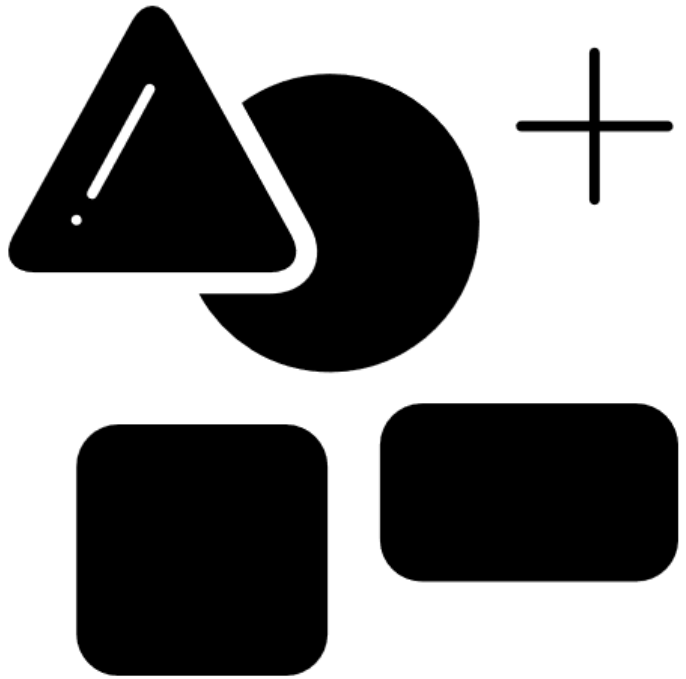
- Deploy OVS Switch using containers
- Create VNFs corresponding to network services on OVS Switches

CPE 1.2: Configure SDN Infrastructure

- Deploy SDN Controller to configure and manage flows in VNFs
- Deploy overlay network for connectivity
- Deploy packet analyzer for analysis

CPE 1.3: Test VM Creation

- Deploy Linux based Test VM
- Emulate test environment on Test VM
- Perform operational, performance and configuration tests



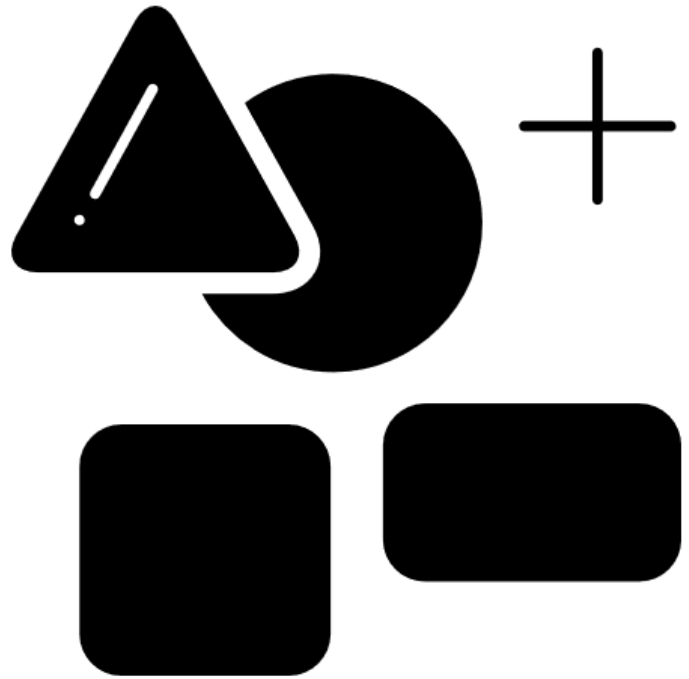
Critical Project Elements – Technical (continued)

CPE 1.4: Service Chain Creation

- Create multiple service chains
- Simulate and test different traffic scenarios

CPE 1.5: Test results (Storage and display)

- Database to store and parse the output of test results
- Web-interface to display the output of test results



Critical Project Elements - Logistical

CPE 2.1: Hardware Devices

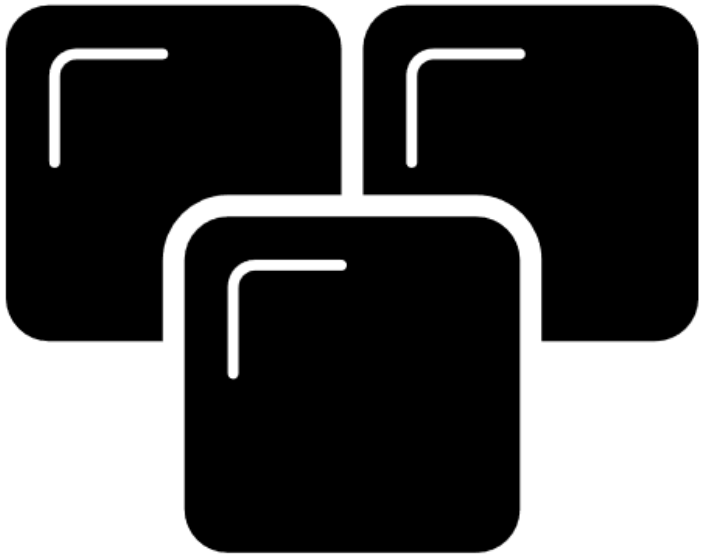
- Using servers and traditional switches available in Telecommunications lab
- Laptops with Windows, Linux and MacOS as host devices

CPE 2.2: Containers/ Controllers

- Using open-source software for topology creation
- Dockers as containers, ONOS as SDN controller

CPE 2.3: Knowledge and concepts

- Understand the functionality of containers
- Determine web GUI framework and learn the functionality

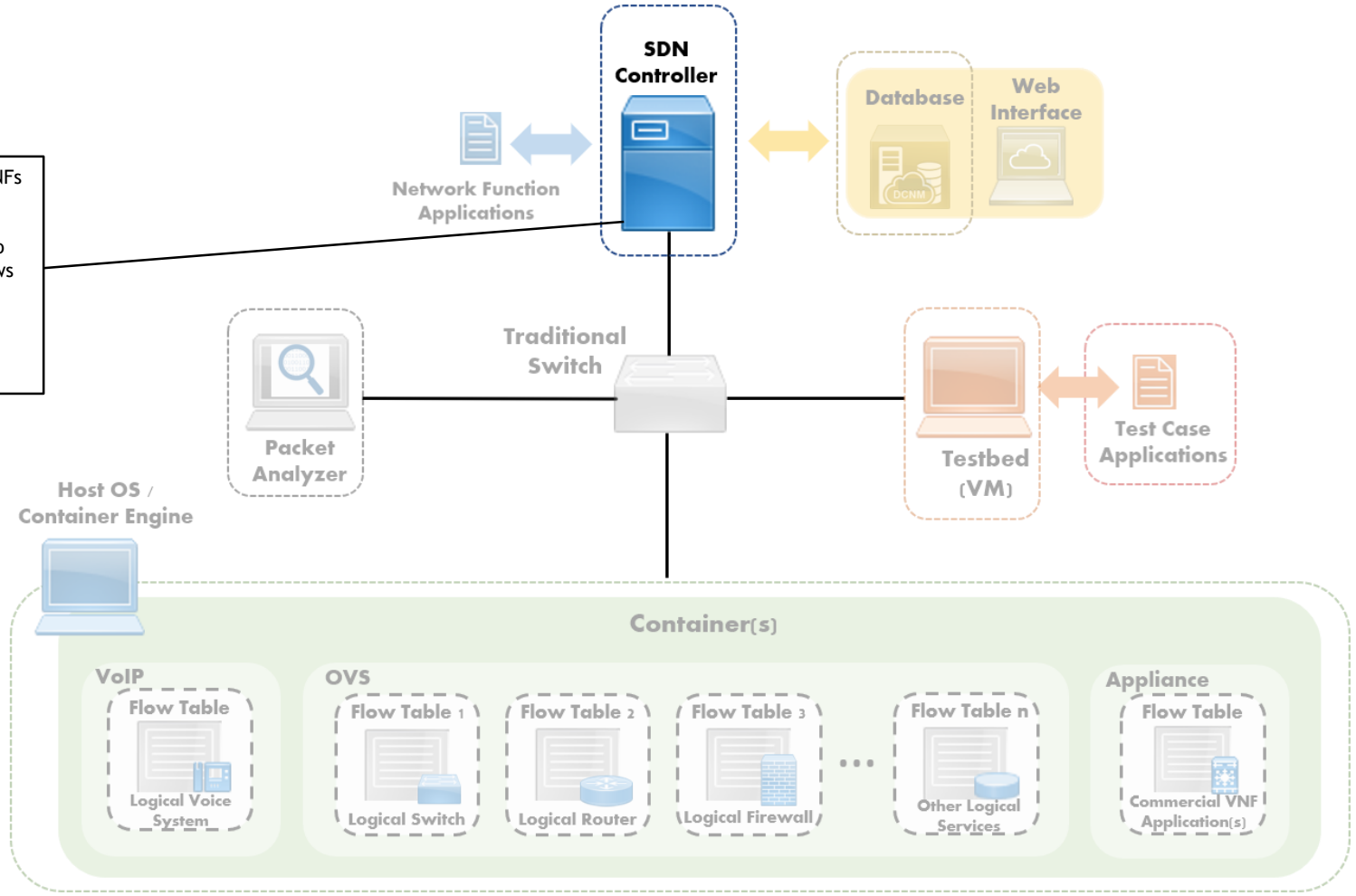


Design Requirements

Controller has a web-interface to install VNFs on the hosts with push of a button

Controller supports NBI-application layer to push intents to controller and SBI-push flows as per the intents to the hosts

Controller sends notifications to test VM to perform test cases on the hosts



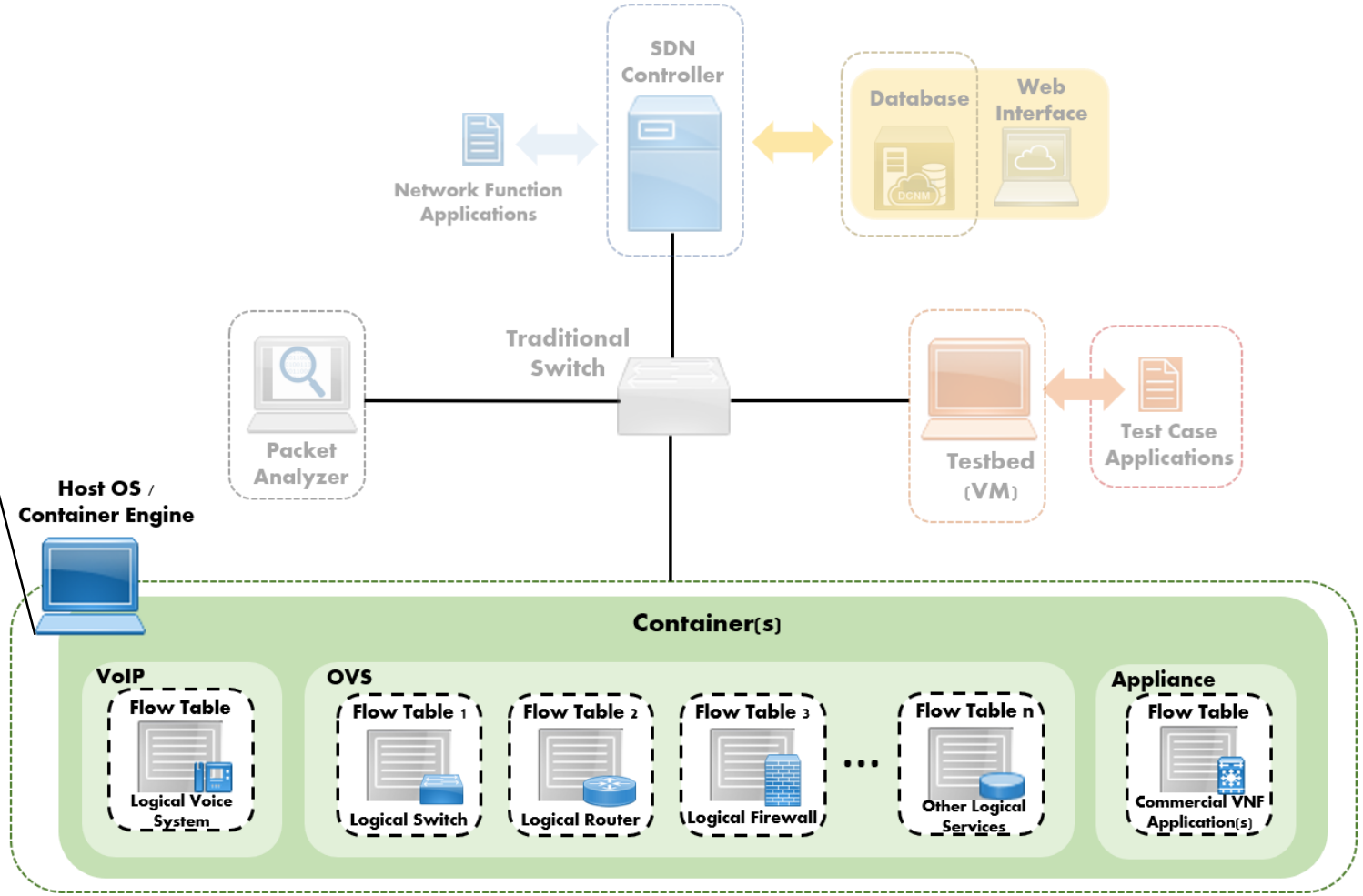
Design Requirements (continued)

Containers are hosted on Windows and Linux OS

Containers are used to spin up VNFs

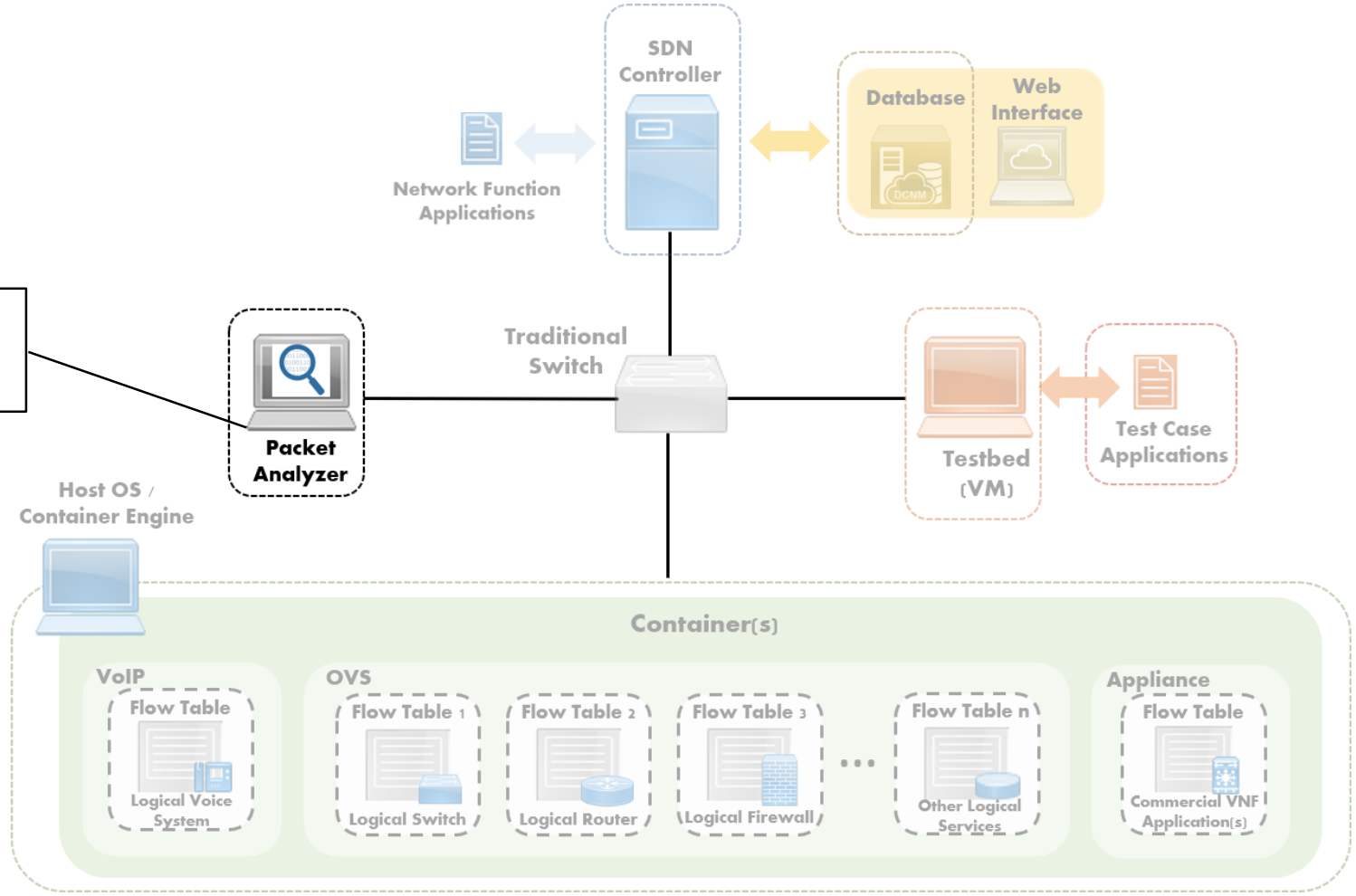
OVS images are used to create VNFs

VNFs are created for individual components in the service chain

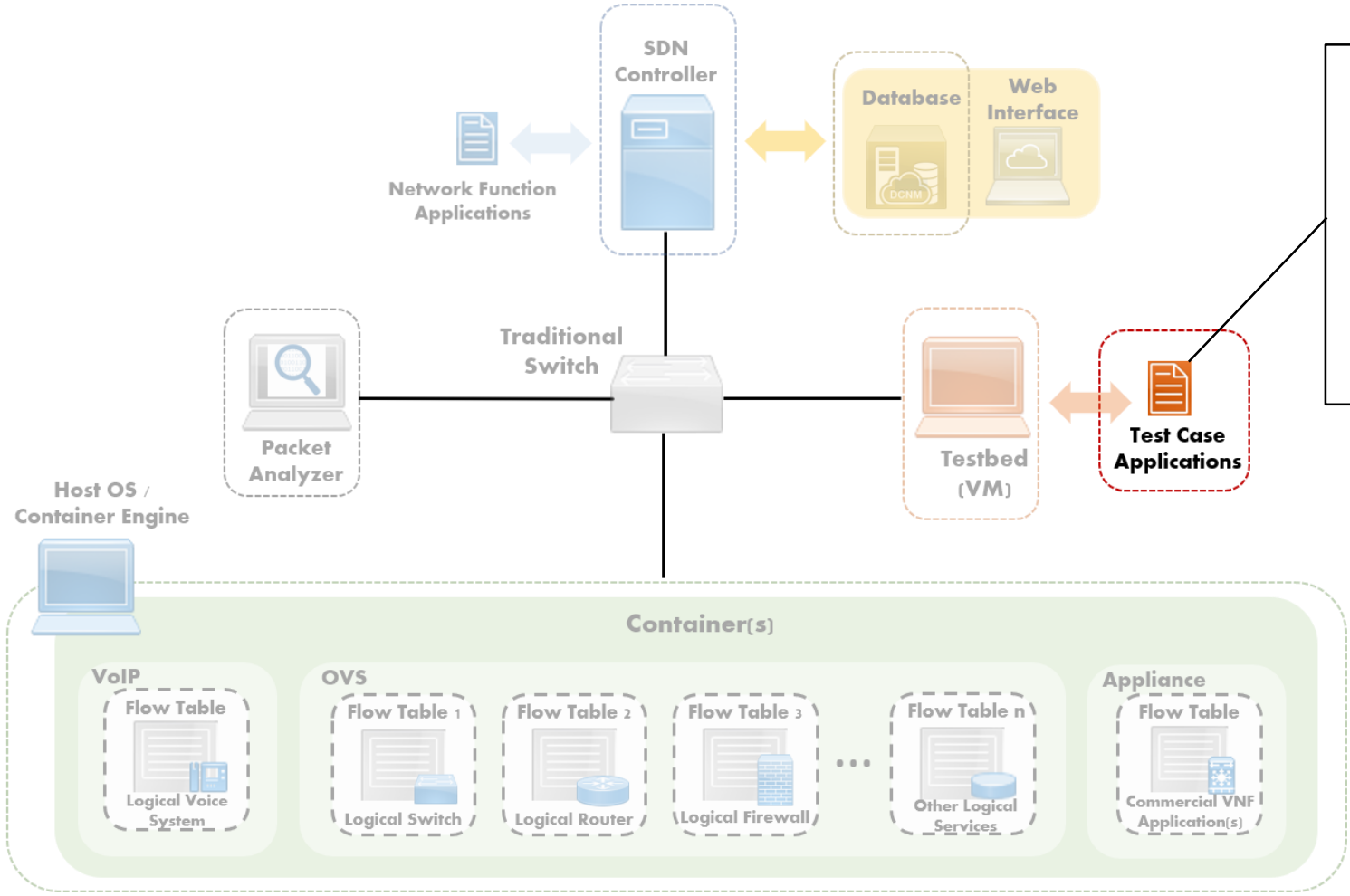


Design Requirements (continued)

Packet Analyzer is used to capture south-bound traffic, used to analyze performance and design test cases



Design Requirements (continued)

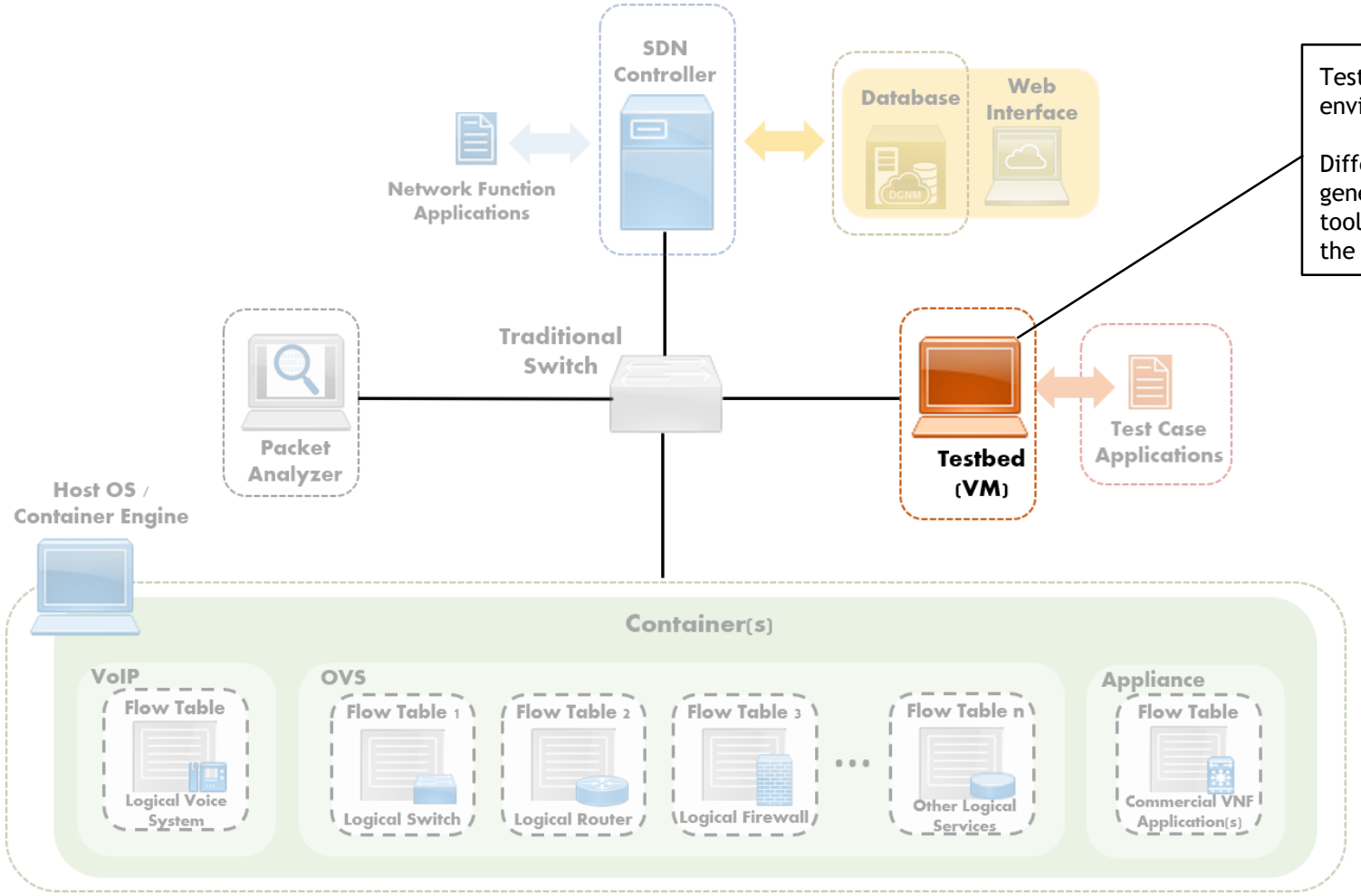


Test cases are written to verify the configuration, performance and failover mechanisms for the service chain

Test cases are written for various traffic parameters such as TCP, UDP, ICMP, payload, encrypted and unencrypted traffic

Test cases check VNFs for performance parameters such as latency, throughput, jitters, delay, losses, memory utilization, and redundancy

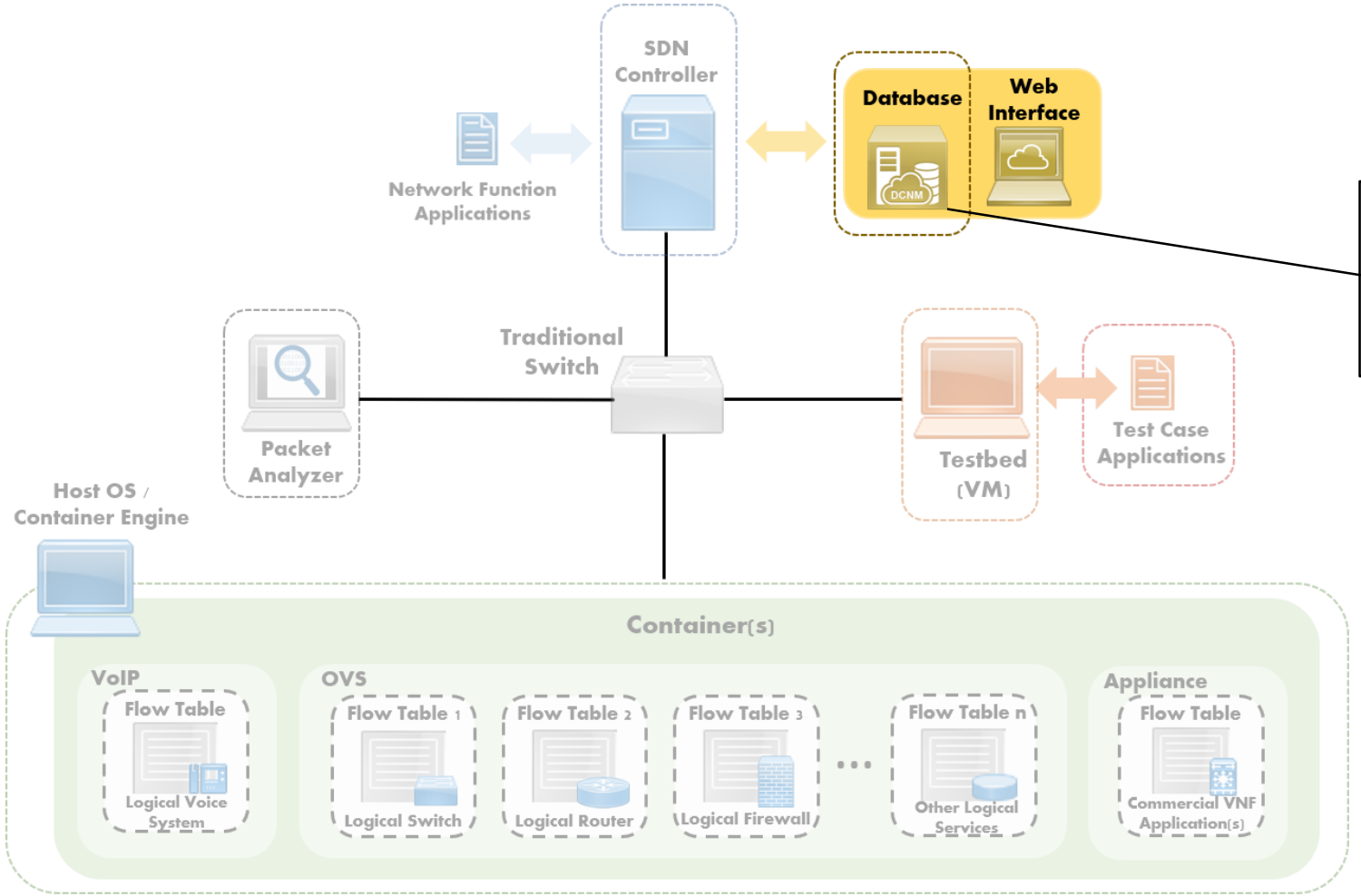
Design Requirements (continued)



Test VM(s) will host a test environment to run test cases

Different traffic types will be generated by the traffic generator tools to test the functionality of the VNFs

Design Requirements (continued)



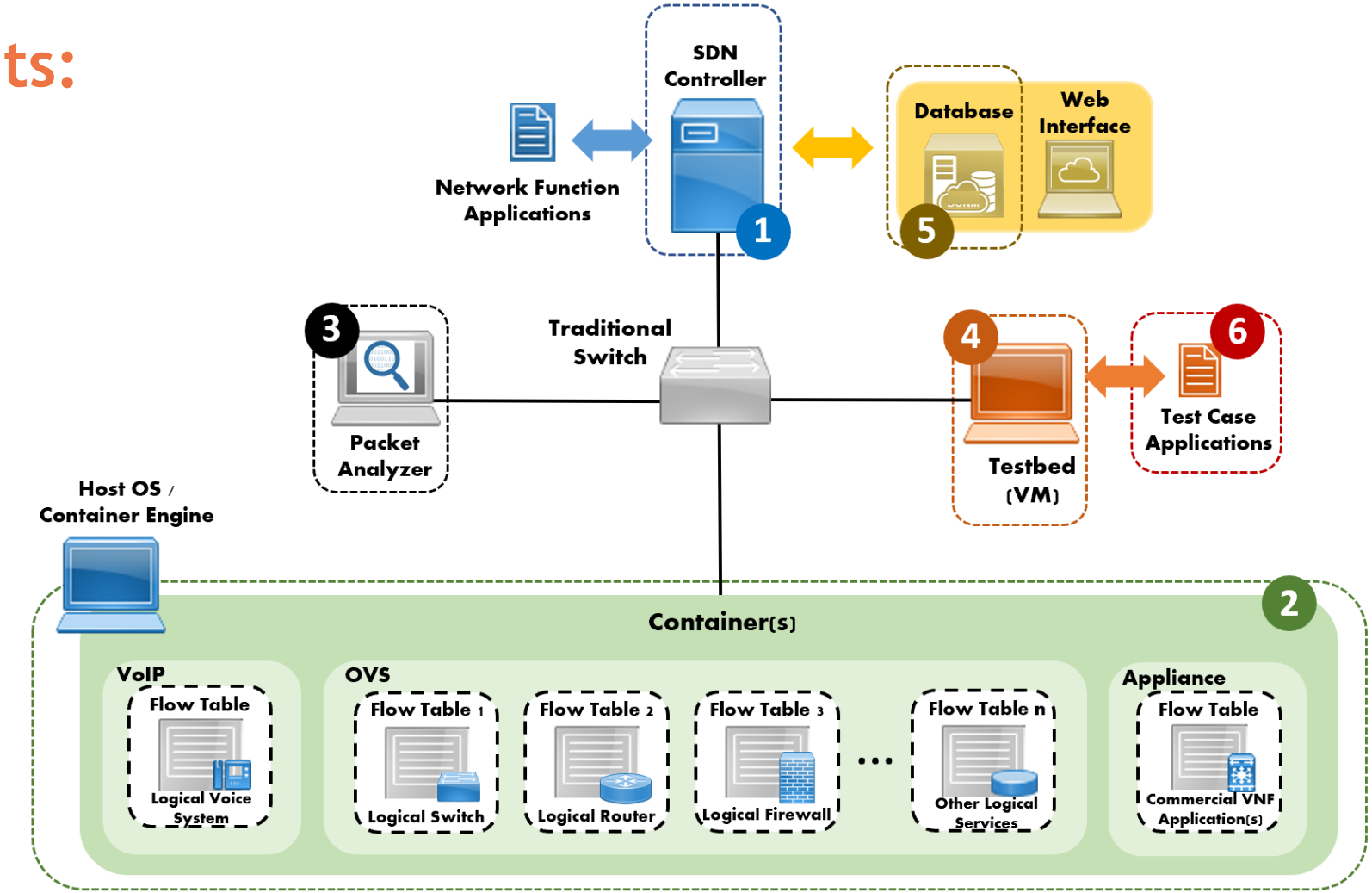
Database manager stores and processes the test results

Output of the test cases are stored and displayed in real-time on the web interface

Baseline Design from CDD

Key Design Elements:

1 SDN Controller
2 Containers
3 Packet Analyzer
4 Hypervisor
5 Database manager
6 Traffic generator



Baseline Design Summary

Project Elements	Selected Element	Rationale
SDN Controller	ONOS	Provides northbound abstraction using REST Supports multiple southbound protocols Well documented Interactive GUI
Container	Docker	Lightweight Can be deployed on Linux and Windows OS Suitable for continuous deployment and integration
Hypervisor	KVM	Enhanced VM security and isolation Bare metal speed Better network virtualization support

Baseline Design Summary (continued)

Project Elements	Selected Element	Rationale
Packet Analyzer	Wireshark	Lightweight and user-friendly GUI Available for all flavors of OS Supports many protocols Allows the capture on multiple interfaces at once
Database Manager	SQLite	Support for multiple connections Lightweight
Traffic Generator	Iperf	Easy to install and implement across multi-vendor platforms Supports both IPv4 and IPv6 traffic The results are saved in easily readable log files Supports multithreading

Baseline Design Summary (continued)

Project Elements	Selected Element	Rationale
Southbound Protocol	OpenFlow	Standard SDN Protocol Supported by most existing hardware Supports TCP & TLS connection
Northbound Protocol	REST API	Supported by most devices Easy to implement using curl & requests.post method Popular in industry

Verification and Validation

LEVEL 6 Multi-environment Testing

LEVEL 5 Stress Testing

LEVEL 4 Performance Testing

LEVEL 3 Compatibility Testing

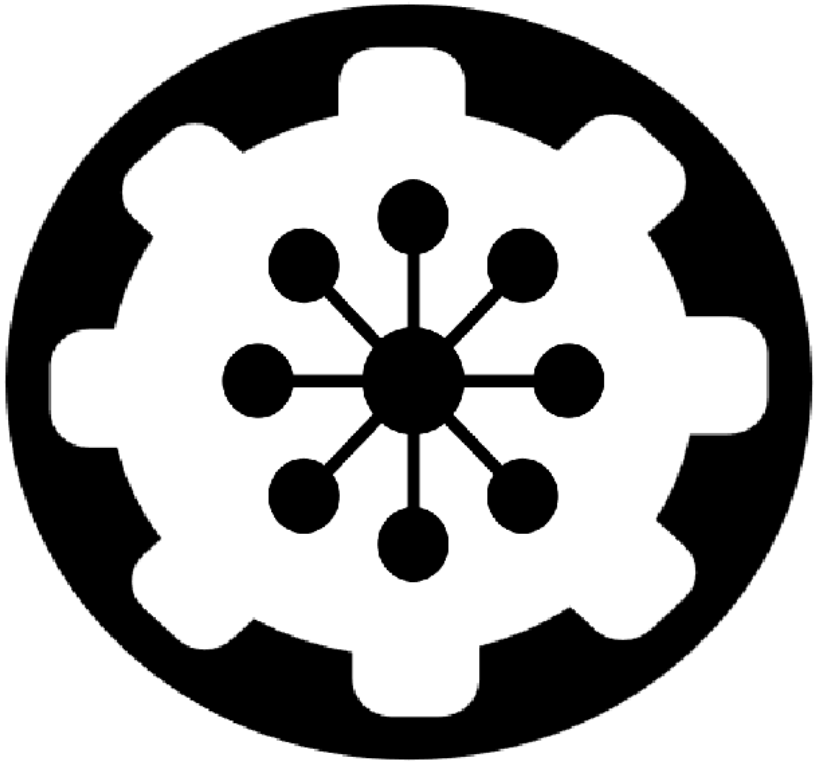
LEVEL 2 Integration Testing

LEVEL 1 Deployment Testing

Level 1: Deployment of Network Infrastructure

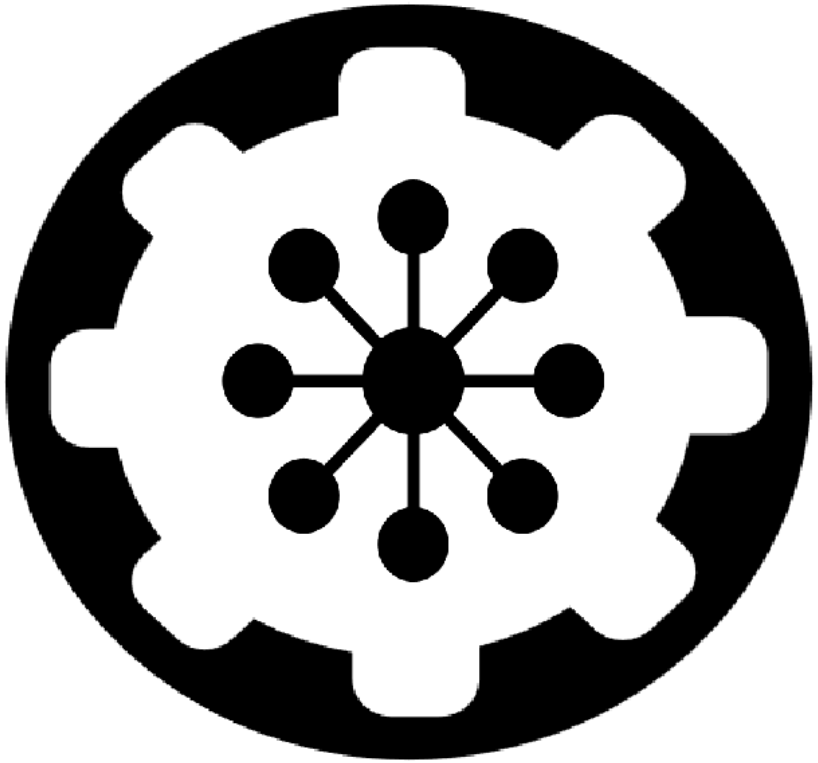
Test the deployment of network devices & software tools

Containers	• Verify the installation of OVS switches on the host devices using containers. Validate the deployment and configuration of VNFs on OVS switches
SDN Controller	• Validate the deployment and configuration of Controller VMs using containers. Verify installation of SDN Controller application in the Controller VMs



Level 1: Deployment of Network Infrastructure (continued)

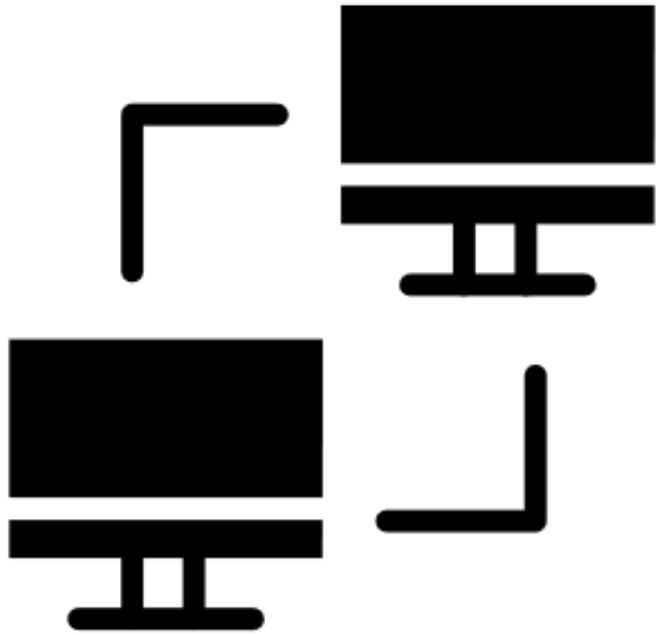
Storage & Output	• Validate deployment of storage instances and web interface
Test VM	• Verify the creation of test environment and installation of traffic generator and packet analyzer in Test VM
Traditional Devices	• Validate deployment of intermediate network devices using python script



Level 2: Integration of Network Infrastructure

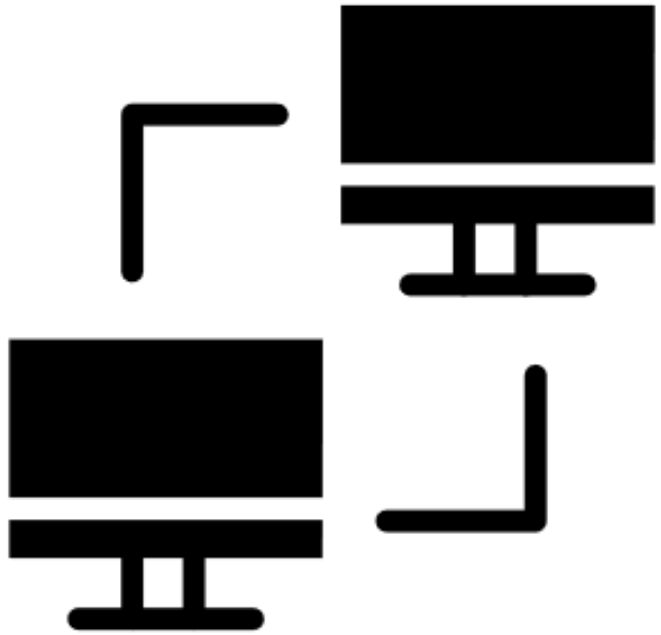
Test the reachability between network devices & VNFs

Containers	• Verify the configuration, interface status of the OVS switches. Validate the creation of service chaining between different VNFs
SDN Controller	• Ensure the reachability between the Controller VMs and the intermediate network devices. Verify the connectivity from Controller VMs to SDN Controller application, storage instances and web interfaces



Level 2: Integration of Network Infrastructure (continued)

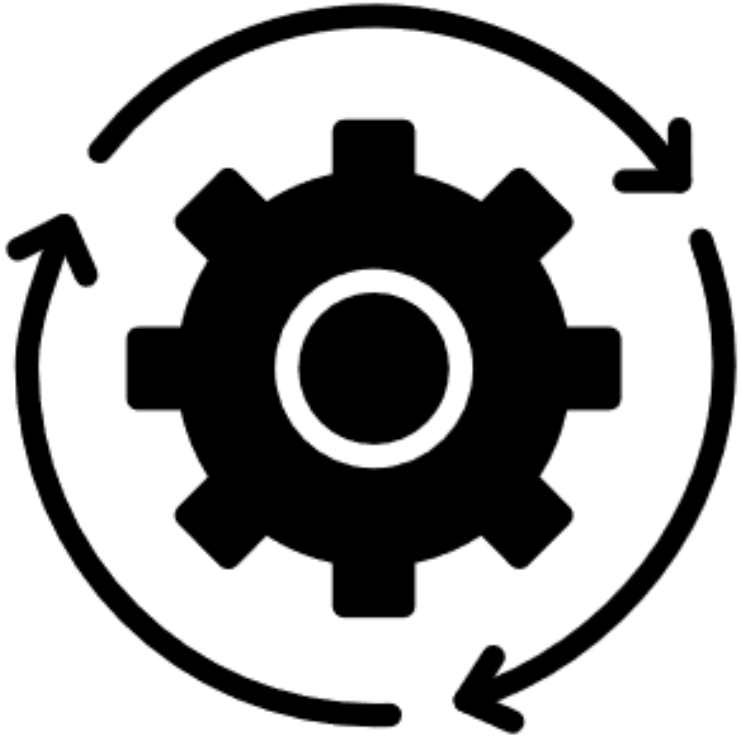
Storage & Output	• Validate the connectivity between storage devices and the web interfaces
Test VM	• Perform the health check of the Test VM interfaces and ensure the connectivity of the TEST VM to the intermediate devices
Traditional Devices	• Verify the status of interfaces connected to the SDN infrastructure devices



Level 3: Compatibility between Network Devices

Verify the seamless functionality of network protocols between network devices

Containers	• Verify installation of flow entries in multiple flow tables in the OVS switches
SDN Controller	• Validate the NBI operation between Controller VM and application layer, the SBI operation between Controller VM and network devices
Storage & Output	• Verify the storage and display of results using mock input values. HTTP Test Automation Hooks to check the web interface functionality



Level 4: Performance Test for VNFs

Test the functionality of VNFs on host devices

Containers

- Verify the functionality of VNFs with respect to expected output
- Evaluate the performance based on latency, throughput, jitters, utilization
- Multiple traffic scenarios generated through traffic generator will be used for testing

Level 5: Stress Test for Network Devices

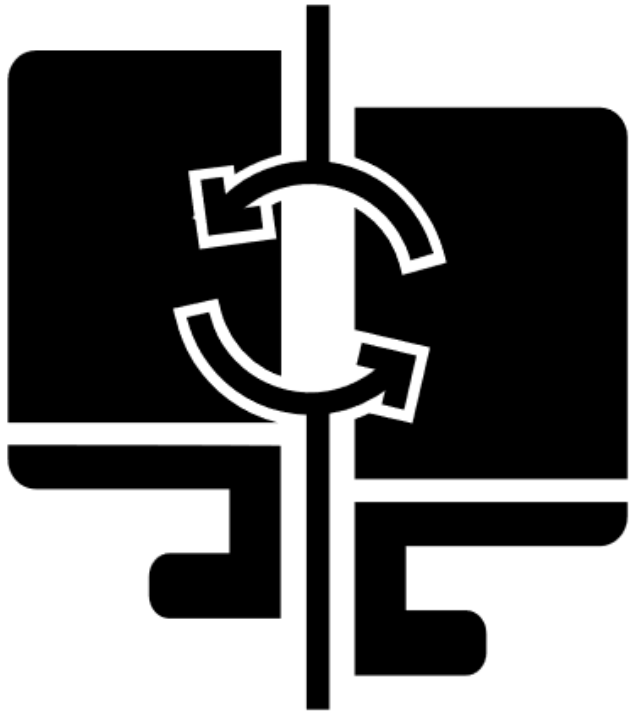
Test the backup scenarios in case of failovers

SDN Controllers

- Verify the seamless functionality provided by the secondary SDN Controller in case the primary controller fails

Traditional Devices

- Verify the reachability of the host devices and Test VM with the SDN Controller through the Access point link in case the switch or links goes down



Level 6: Multi-environment Network Deployment

Test the deployment of network services and functionality on different Operating Systems

Containers

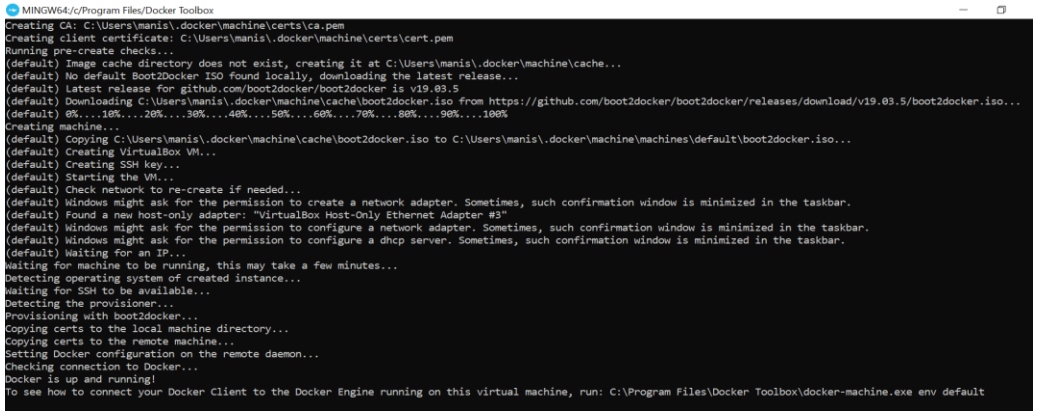
- Deploy network services using containers on different OS (Windows, Linux, MacOS) and test the functionality on different environment

SDN Controller

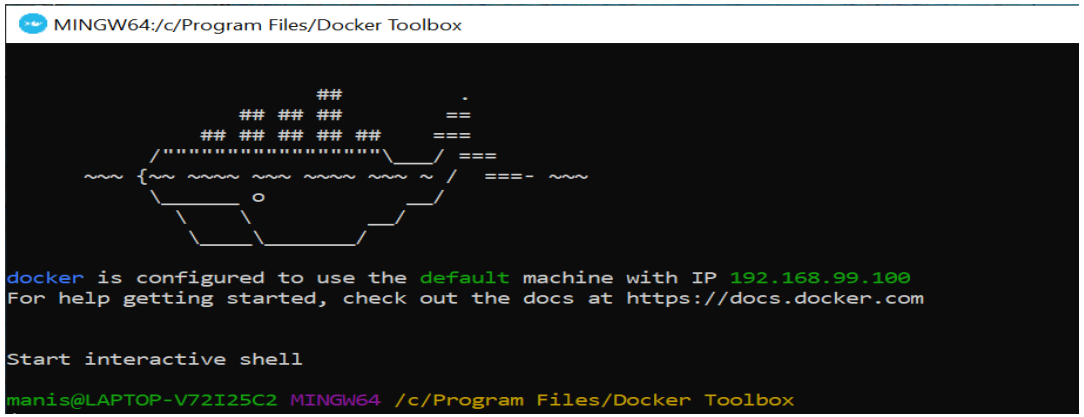
- Deploy SDN Controller using VMs and bare metal servers and test the functionality on different environment

Validation and Verification for Docker

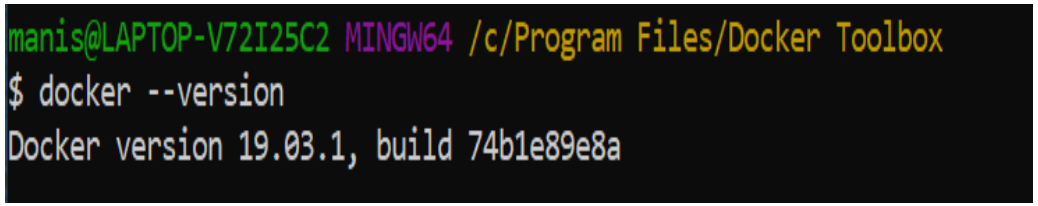
1. Docker installation



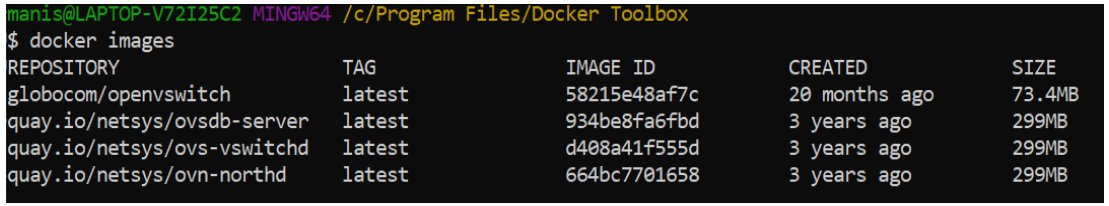
2. Docker CLI



3. Docker version

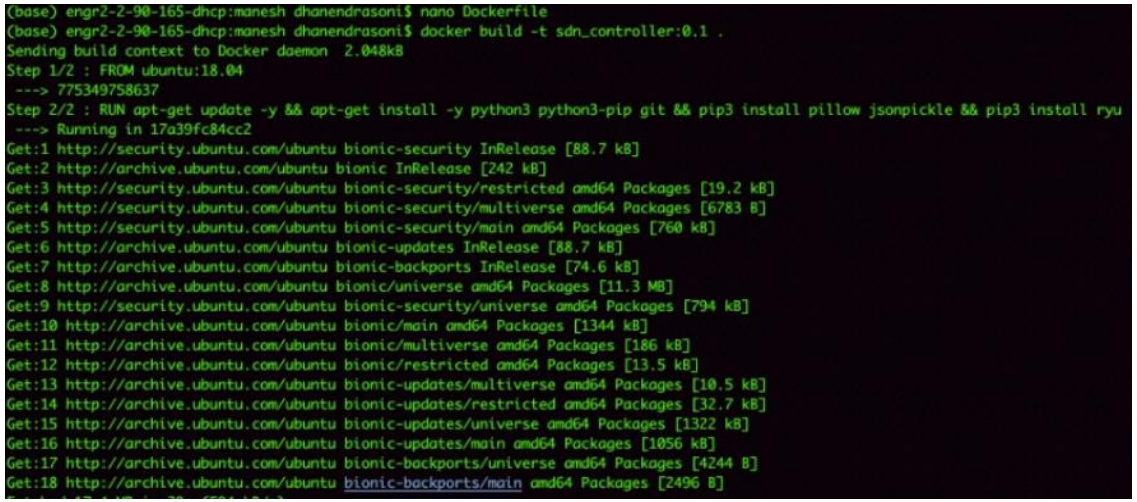


4. Docker images



Validation and Verification for SDN Controller

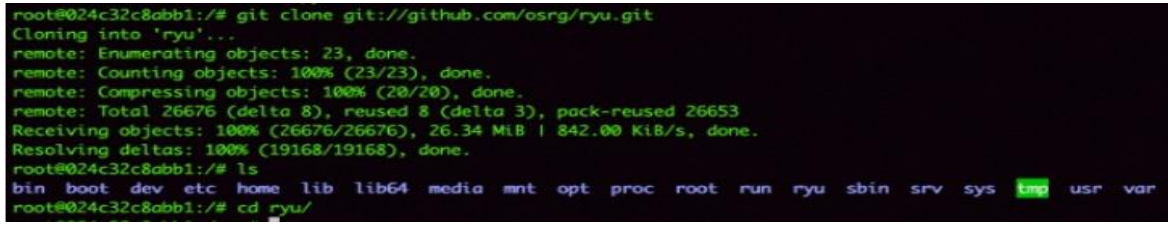
1. VM installation



2. VM configuration

3. VM image running status

4. SDN Controller application installation



Validation and Verification for OpenvSwitch

1. OVS installation

```
(base) engr2-2-90-165-dhcp:manesh dhanendrasoni$ docker run  -i -t --rm --name vswitch --cap-add=NET_ADMIN -d globocom/openvswitch
Unable to find image 'globocom/openvswitch:latest' locally
latest: Pulling from globocom/openvswitch
ff3a5c916c92: Pull complete
a1e2db7c9db6: Downloading [==========>                                        ]  7.669MB/23.96MB
a477371faa9b: Download complete
1ff1c88b1fca: Download complete
b04e3c06a506: Download complete
24a45206440d: Download complete
6bd0e8a2022e: Download complete
31354961aae9: Download complete
```

2. OVS version

```
(base) engr2-2-90-165-dhcp:manesh dhanendrasoni$ docker exec -it ba65efb65cb6 /bin/sh
/ # ovs-vsctl -V
ovs-vsctl (Open vSwitch) 2.8.1
DB Schema 7.15.0
/ #
```

3. OVS images- multiple vendors

```
$ docker ps
CONTAINER ID        IMAGE                          COMMAND                  CREATED             STATUS
NAMES
c08dac5c8af9        quay.io/netsys/ovn-northd      "ovn-northd --log-fi…"   10 minutes ago      Up 10 minutes
ovn-northd
2f8bbe766887        quay.io/netsys/ovs-vswitchd    "ovs-vswitchd --pidf…"   13 minutes ago      Up 13 minutes
ovs-vswitchd
2eda63755692        quay.io/netsys/ovsdb-server    "ovsdb-server --remo…"   13 minutes ago      Up 13 minutes
ovsdb-server
03ab2178b5c5        58215e48af7c                   "/bin/sh -c /usr/bin…"   41 minutes ago      Up 41 minutes
p                   gallant_tharp
2ba8b4adad4b        58215e48af7c                   "/bin/sh -c /usr/bin…"   About an hour ago   Up About an hour
p                   determined_feynman
f73f37f7179e        globocom/openvswitch           "/bin/sh -c /usr/bin…"   About an hour ago   Up About an hour
```

4. OVS bridge and controller setup

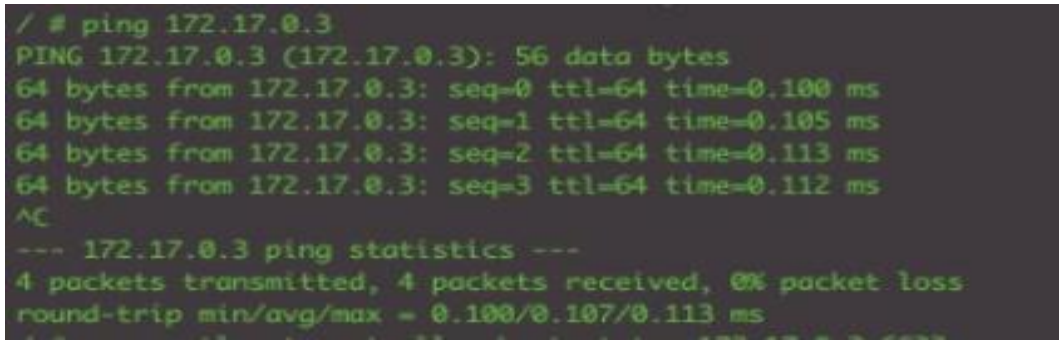
```
/ # ovs-vsctl add-br br-test
/ #
$ docker exec -it 2eda63755692 /bin/sh
# ovs-vsctl add-br br0
/ # ovs-vsctl set-controller br-test tcp:172.17.0.2:6633
/ # ovs-vsctl set-controller br-test tcp:172.17.0.2:6633
```

5. OpenFlow status

```
$ docker exec -it 2eda63755692 /bin/sh
# ovs-vsctl show
7f0a7a09-bba0-4058-9685-83f5fcd9aa13
    Bridge "br0"
        Port "br0"
            Interface "br0"
                type: internal
    Bridge br-test
        Port br-test
            Interface br-test
```

Validation and Verification: Integration & Compatibility

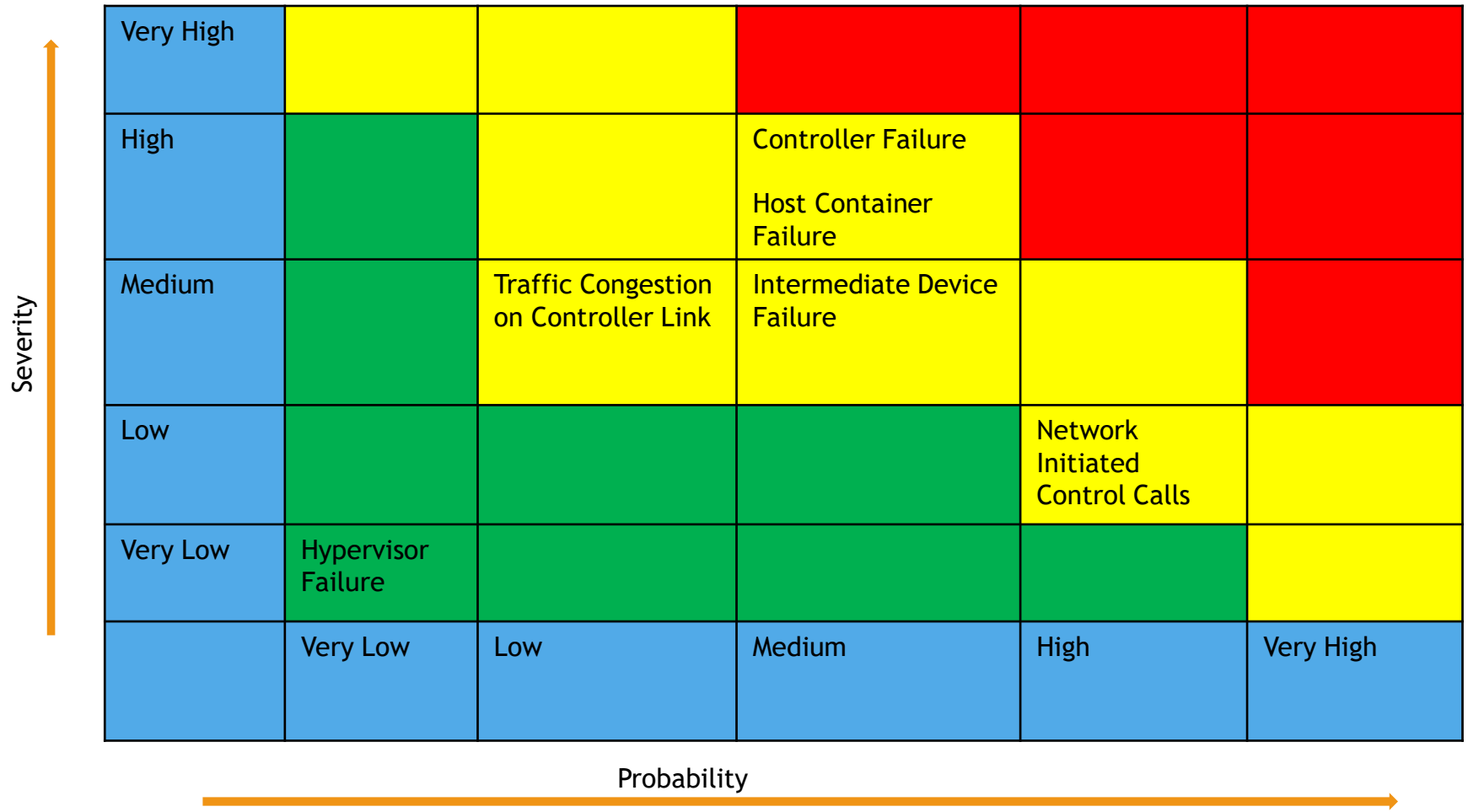
1. Reachability between controller and OVS



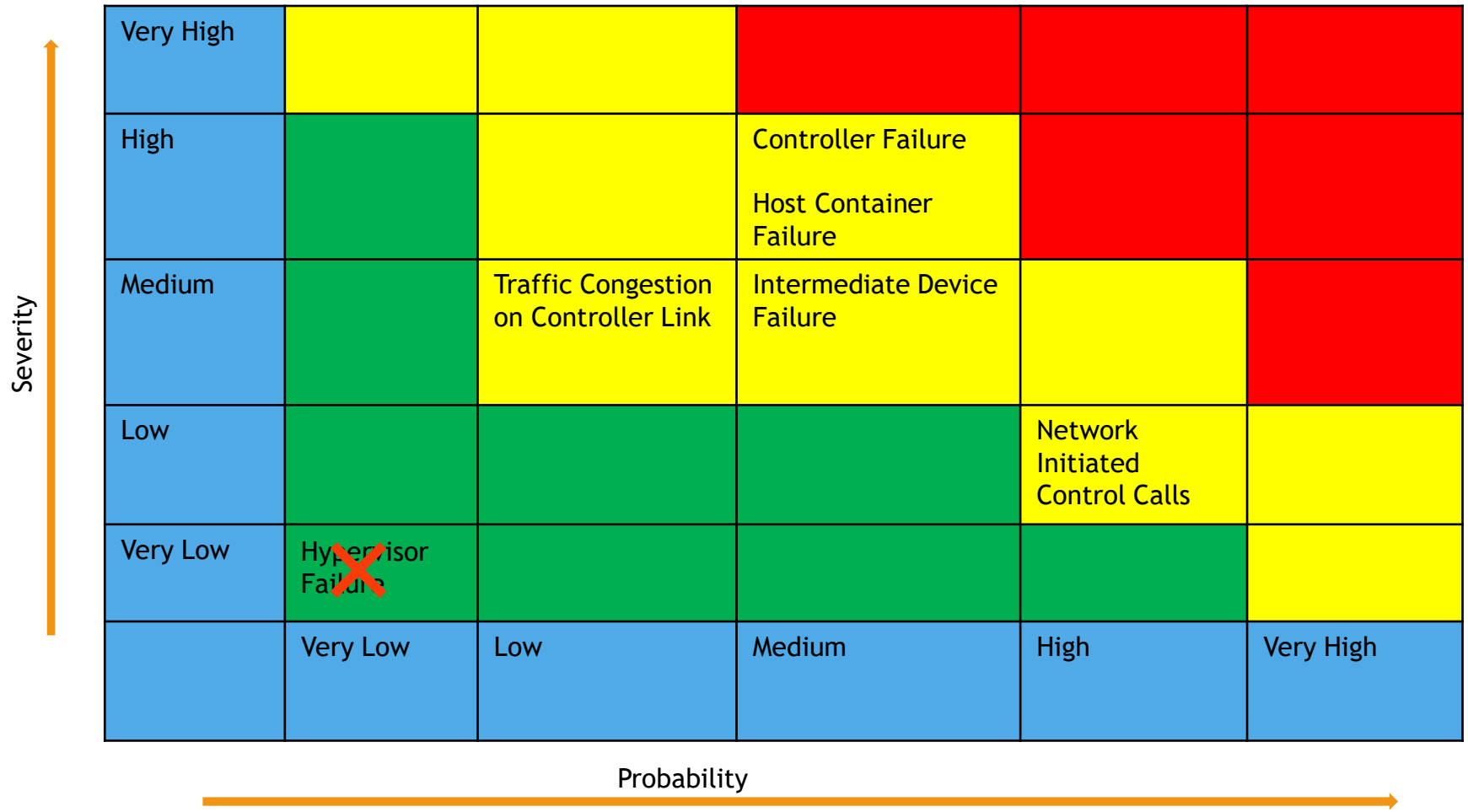
2. SDN controller initiation

3. Southbound protocol connectivity

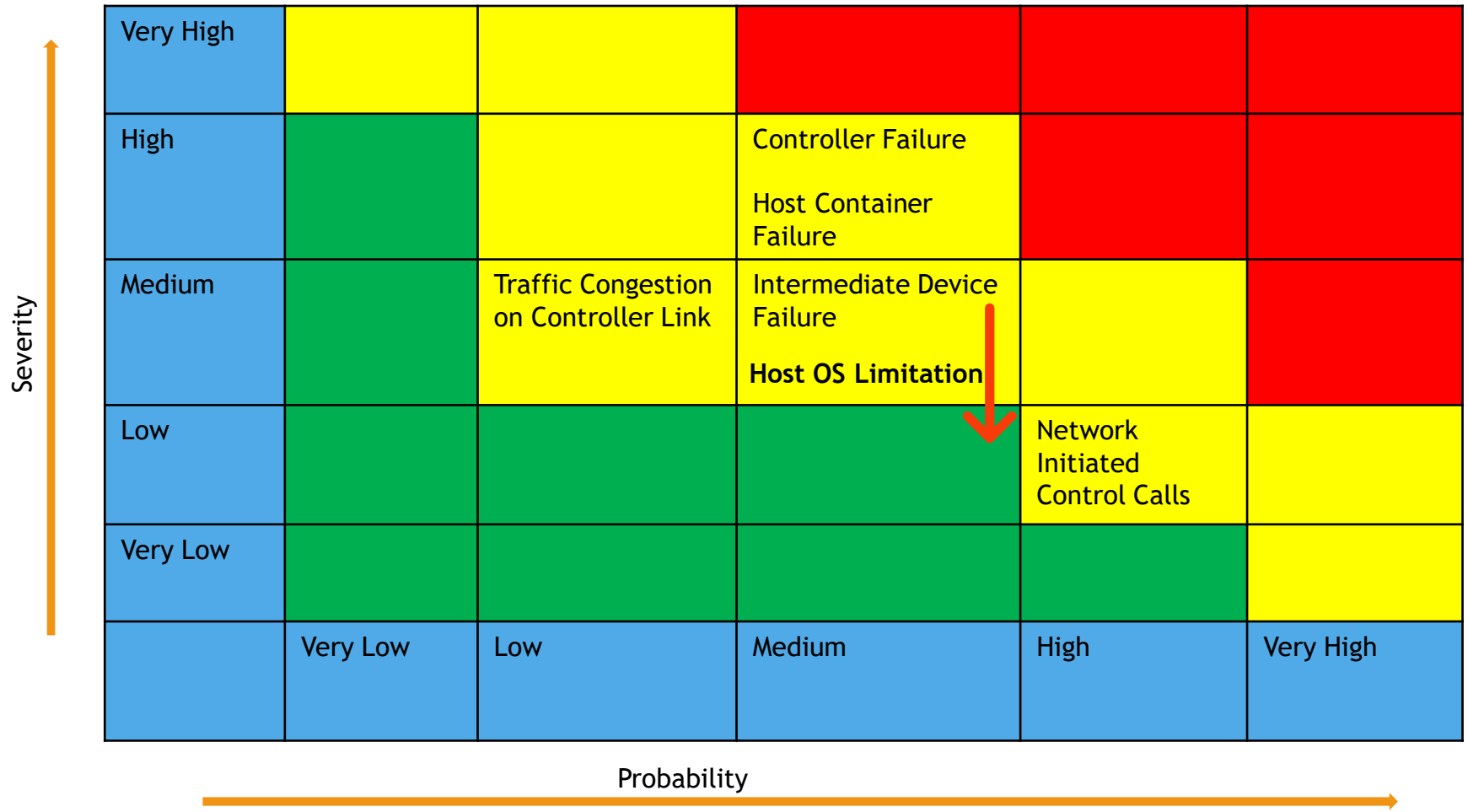
Risk Factors and their Impact - PDR



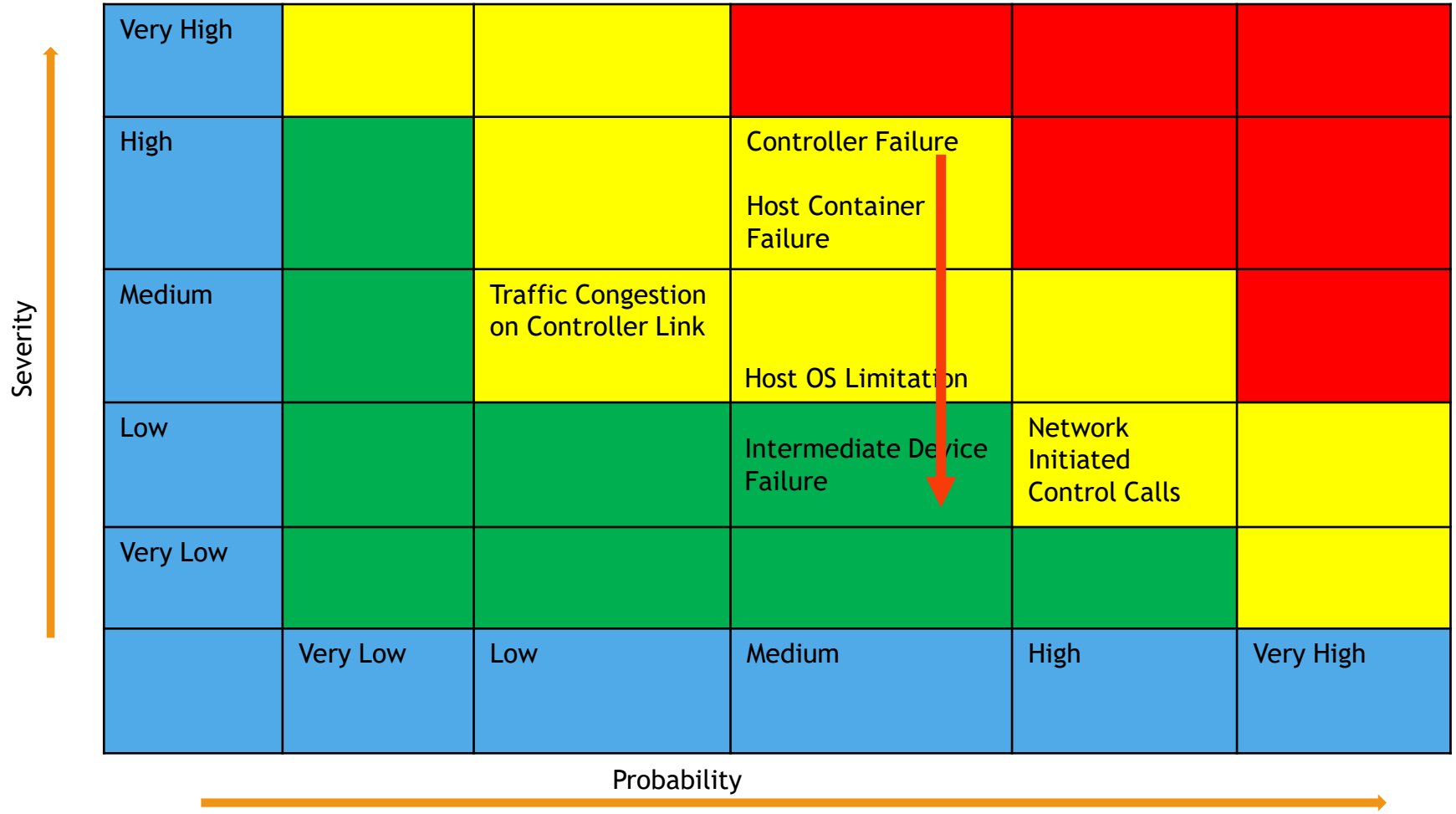
Risk Factors and their Impact - CDR



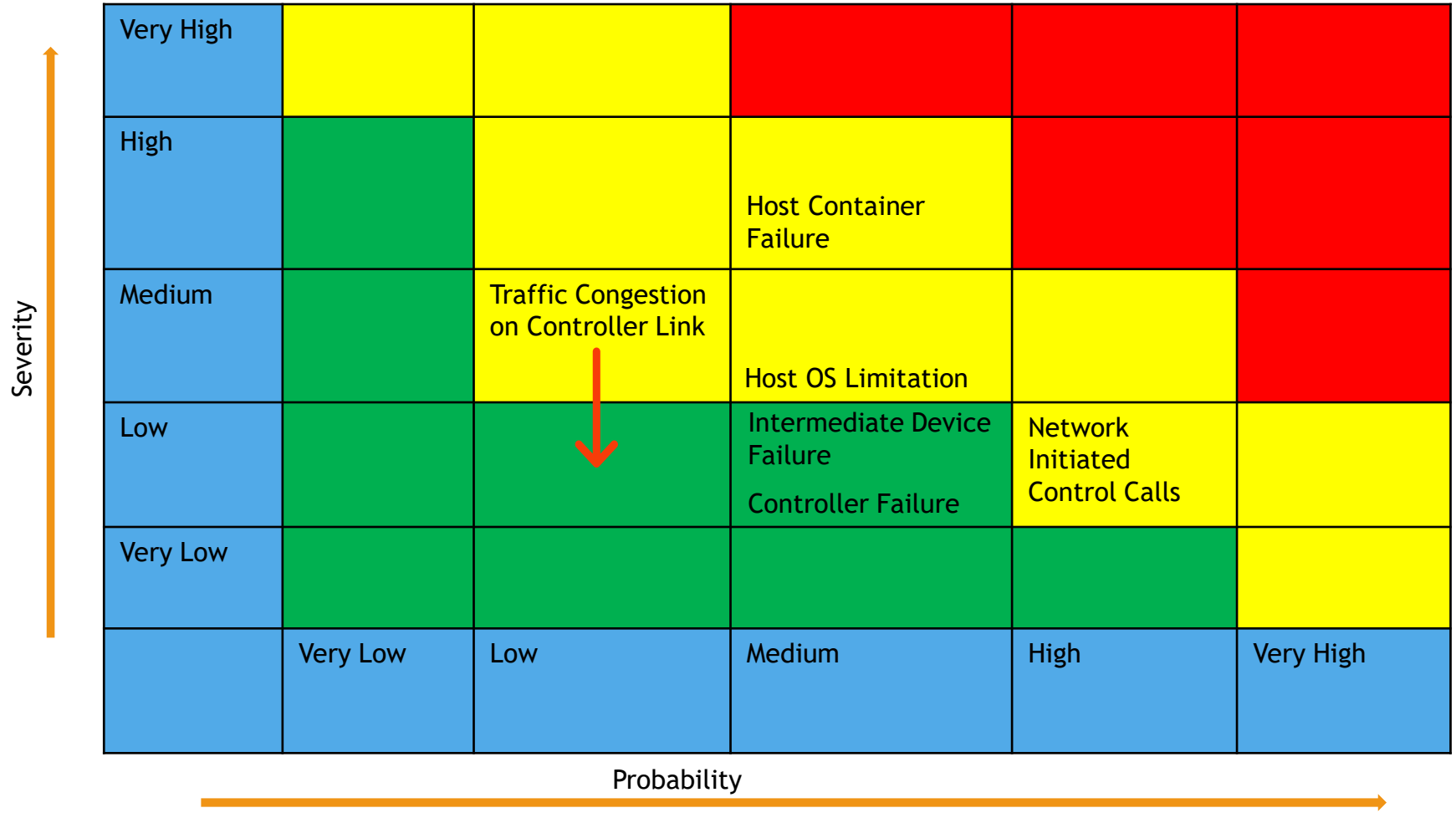
Risk Factors and their Impact - CDR



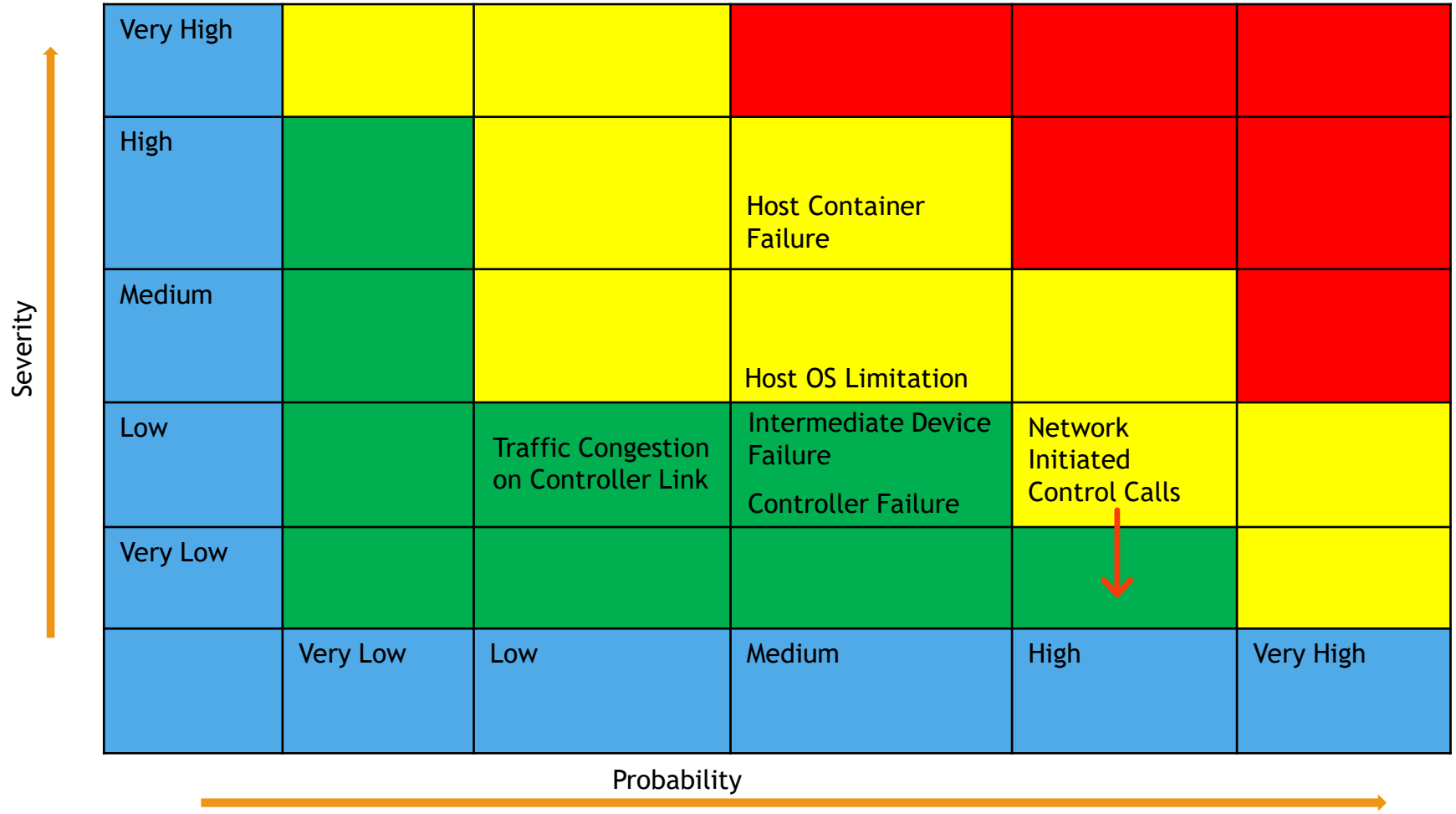
Risk Factors and their Impact - CDR



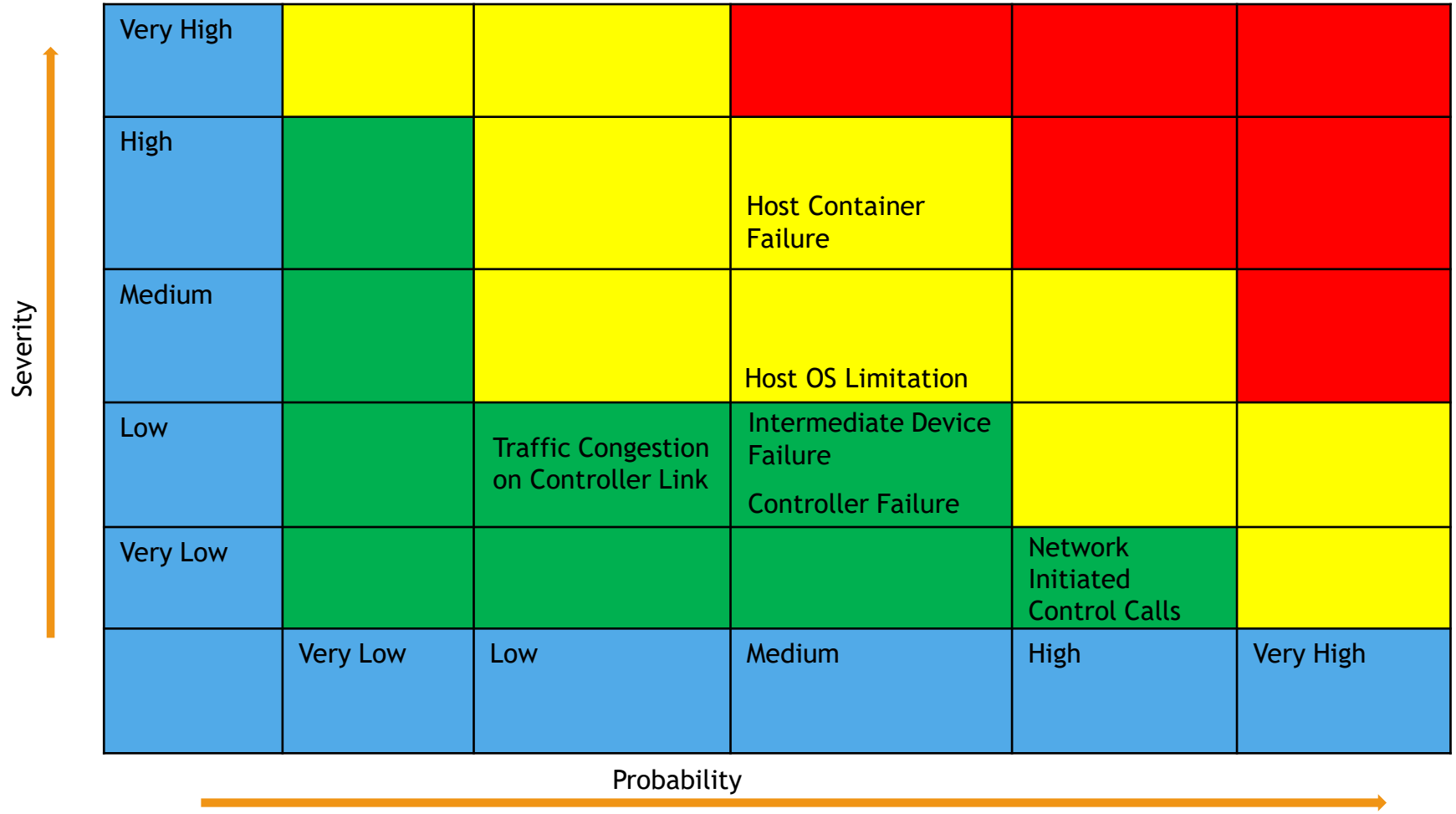
Risk Factors and their Impact - CDR



Risk Factors and their Impact - CDR



Risk Factors and their Impact - CDR



Project Risks: Hardware Failures

Controller Failure

Solution
Introduce a Backup Controller

Future Work
- Deploy a backup server
- Install ONOS on both main and backup servers in a distributed setup

Host Container Failure

Solution
Spin up a backup container on test VM

Future Work
- Deploy a backup server with high compute and storage to support backup container

BACKUP CONTROLLER AND TEST VM HARWARE

Product: DELL POWEREDGE R430 1U RACK SERVER
Features:
- Processor: Intel® Xeon® processor E5-2600 v4
- Memory:64GB
- Storage: 500GB SATA, 146GB SAS
- Networking: 4-1GB Ethernet Network Interface Card

Project Risks: Hardware Failures (continued)

Intermediate Device Failure

Solution

- Use a wireless router with L2 capabilities for redundant connection between devices

Future Work

- Test and deploy Cisco wireless router in the Telecommunications LAB to support high speed wired and wireless connection between all devices

BACKUP HARWARE FOR REDUNDANT CONNECTIONS

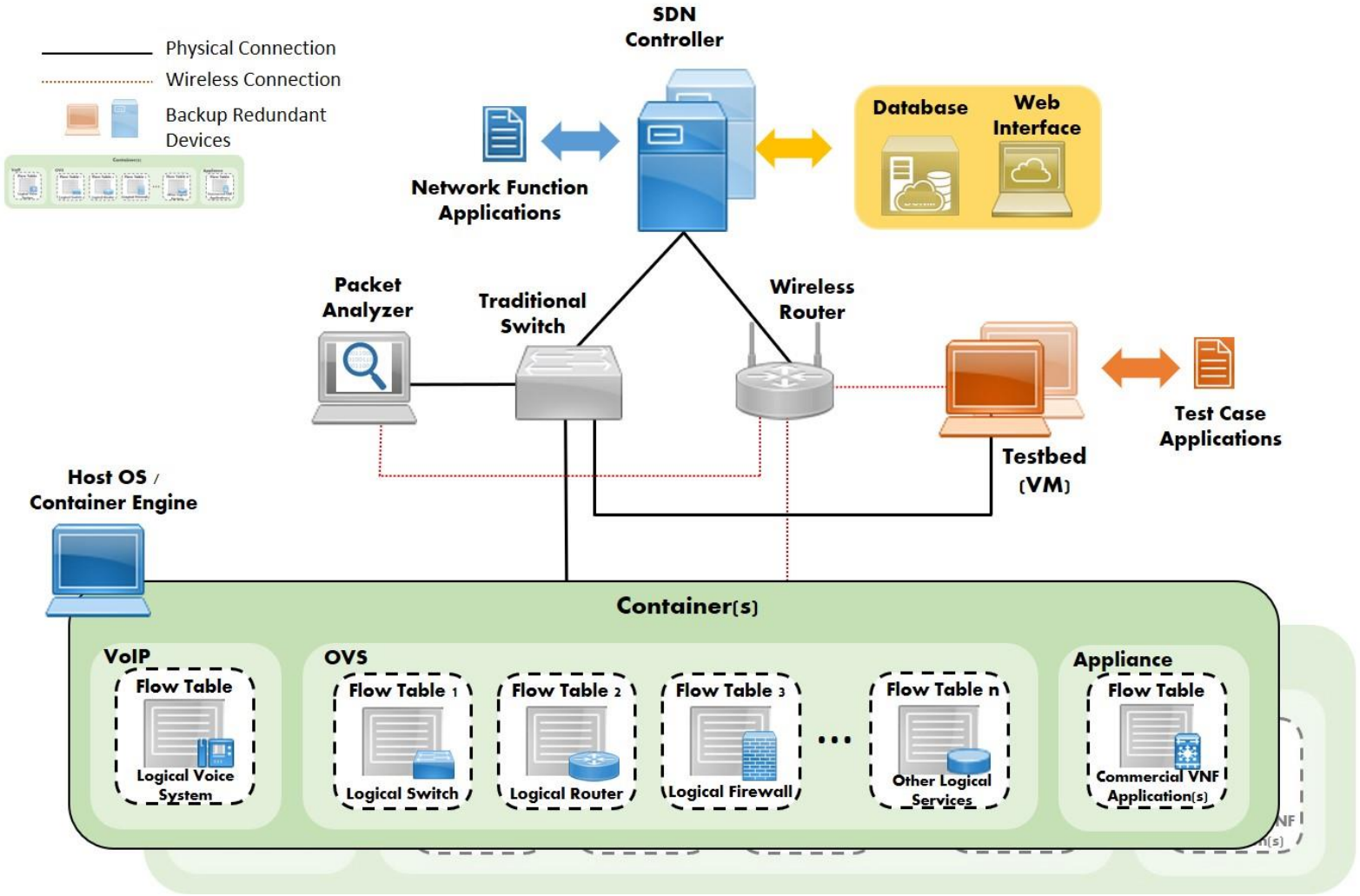
Product: CISCO 1811W Wireless Router
Features:

- Connectivity: Wired and Wireless
- Wireless Protocol: 802.11a/b/g
- Max Range Indoors: 320fts
- Integrated Switch: 8 port Switch

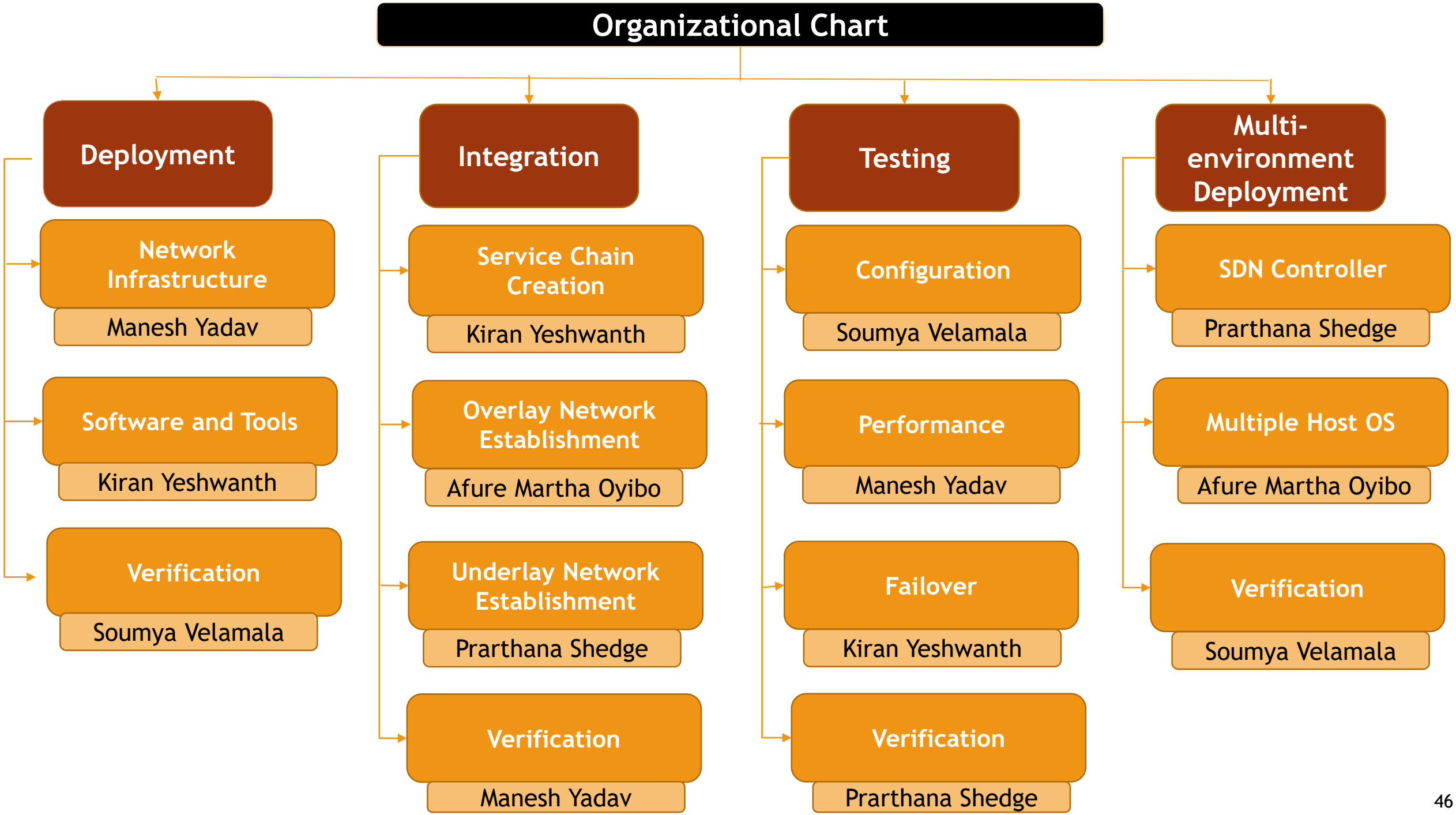
Project Risks: Non-hardware Failures

Host OS Software support(Windows only) for container management	**Solution** Obtain an Enterprise Windows OS edition with intel vPro support **Future Work** • Discuss with Prof Jose about obtaining right host laptops with software support • Fall back to non-windows hosts if we unable to obtain software support
Traffic Congestion on Controller Link	**Solution** Backup secondary link for test traffic **Future Work** • Test and deploy dual connection to controller via two NIC cards for OpenFlow and test bed traffic segregation
Network Control Calls	**Solution** Implement proactive flows **Future Work** • Test multiple proactive flows based on topology to reduce network calls

Revised CONOPS



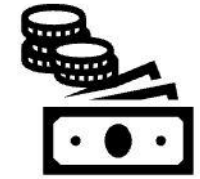
Organizational Chart



Work Breakdown Structure

Network Infrastructure Setup	Deploy VNFs using containers	**Phase 1**
	Deploy SDN Controllers using containers	
	Create test VM on hypervisor	
	Setup traditional switches to form the backbone network	
Software and Tools Installation	Web GUI creation	**Phase 2**
	Database management system installation	
	Setup traffic generator and analyzer	
Network Integration	Service chain creation between VNFs	
	Network orchestration	
	Establish underlay and overlay connectivity	
	Integrate web interface with database management system	
	Integrate test environment on test VM	
	Establish connectivity between test VM and database management system	
	Establish Northbound communication between Controller and Application layer	
	Establish Southbound communication between Controller and VNFs	
	Establish end-to-end service accessibility	
	Automate network configuration	
Testing and Verification	Test configuration and active status of network infrastructure	**Phase 3**
	Test connectivity between topology devices	
	Test VNFs behavior for different traffic scenarios	
	Compare performance of VNFs in comparison to traditional networks	
	Detect network failures and verify activation of backup solutions	
One-Touch Orchestration	Enable provisioning and orchestration of network services via web interface	

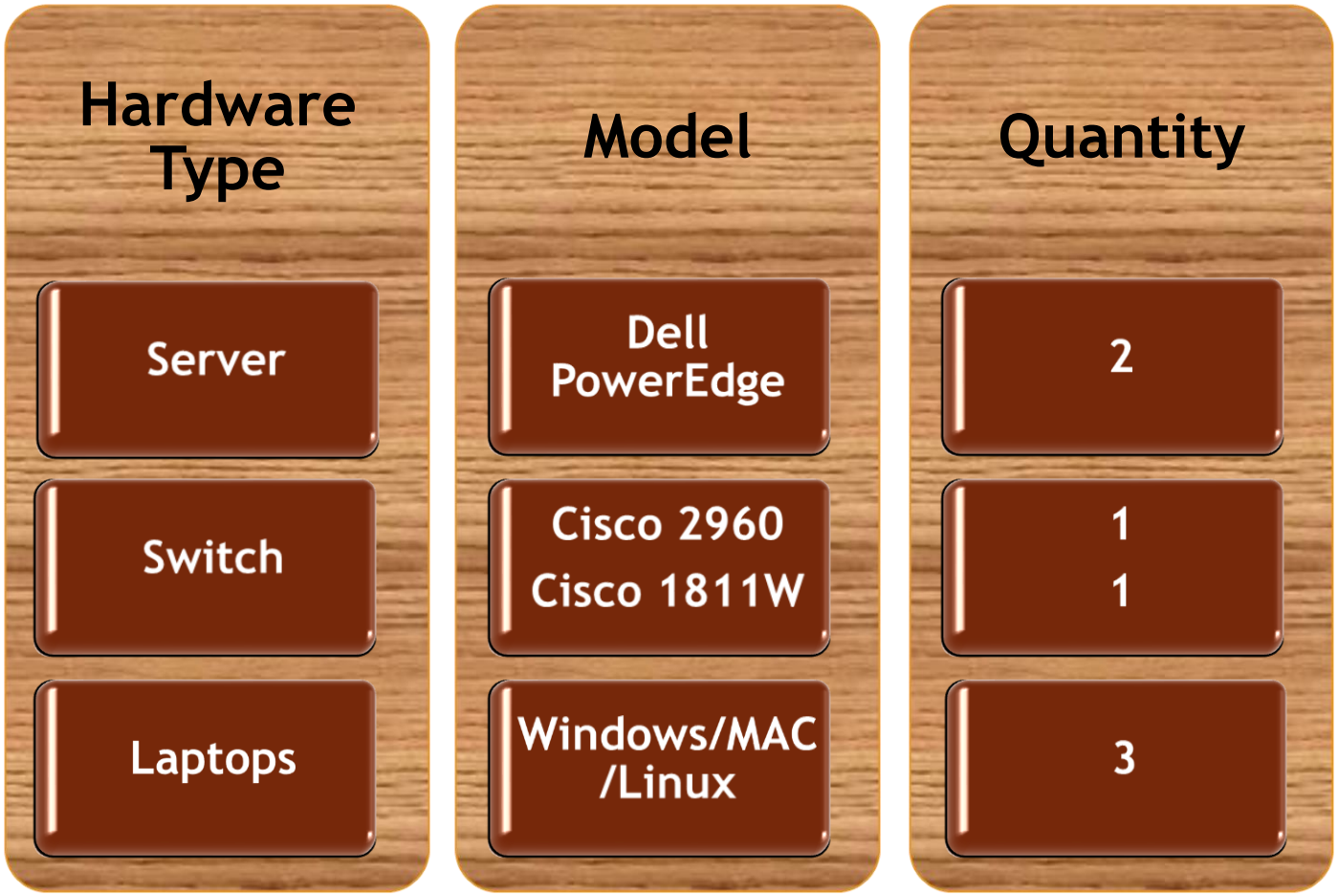
Cost Plan



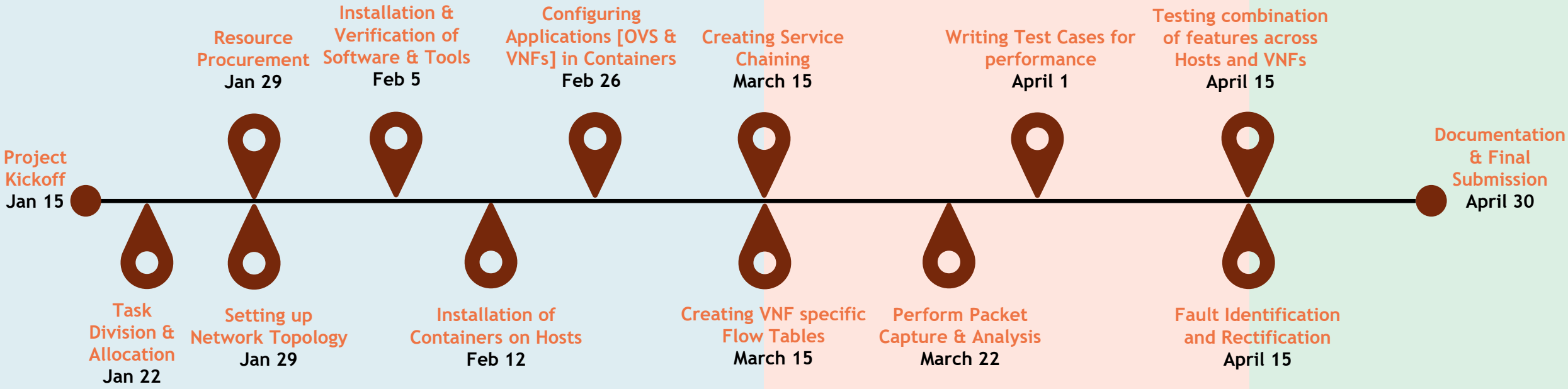
Cost Plan

- **Infrastructure (MacBook) $250** — VNFs will be hosted and tested on Windows, Linux, and macOS, hence procurement of laptop that supports macOS is required for project's success
- **License & Network Infrastructure $0** — Besides laptops, Telecommunication lab resources, and open-source software's, no additional resources are necessary
- **Other Expenses $100** — Additional expenses for accessories and minor damages/repairs

Hardware Requirements



Project Targets and Key Deadlines



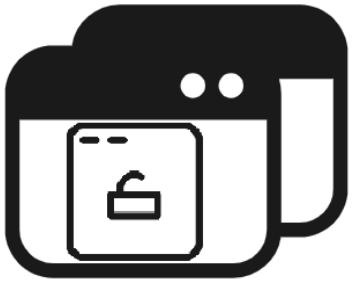
Q&A

Thank You

Backup Slides

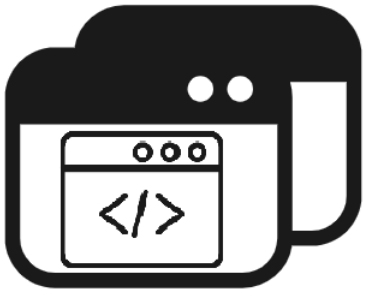
Evaluation Parameters

Cost

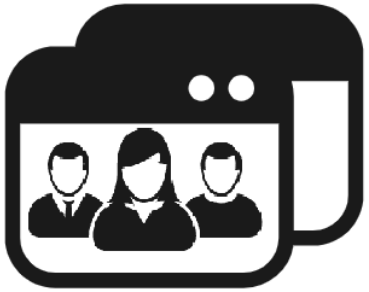


Open-source

Features

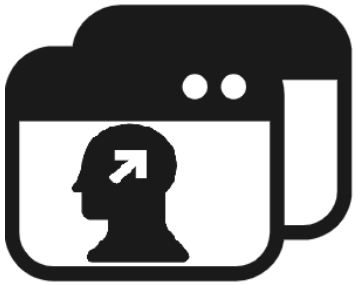


Programmability

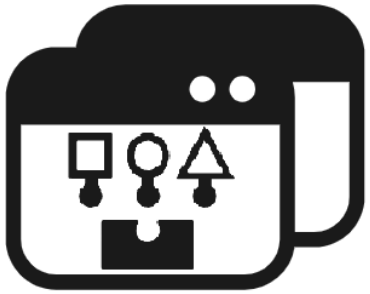


Community Support

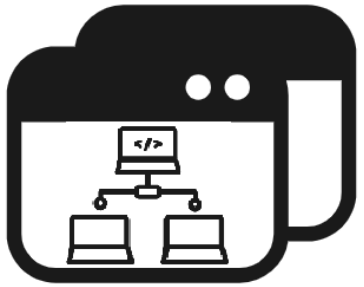
Documentation



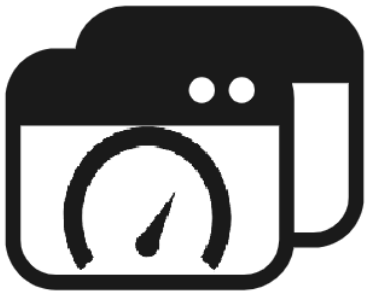
Learning Curve



Modularity



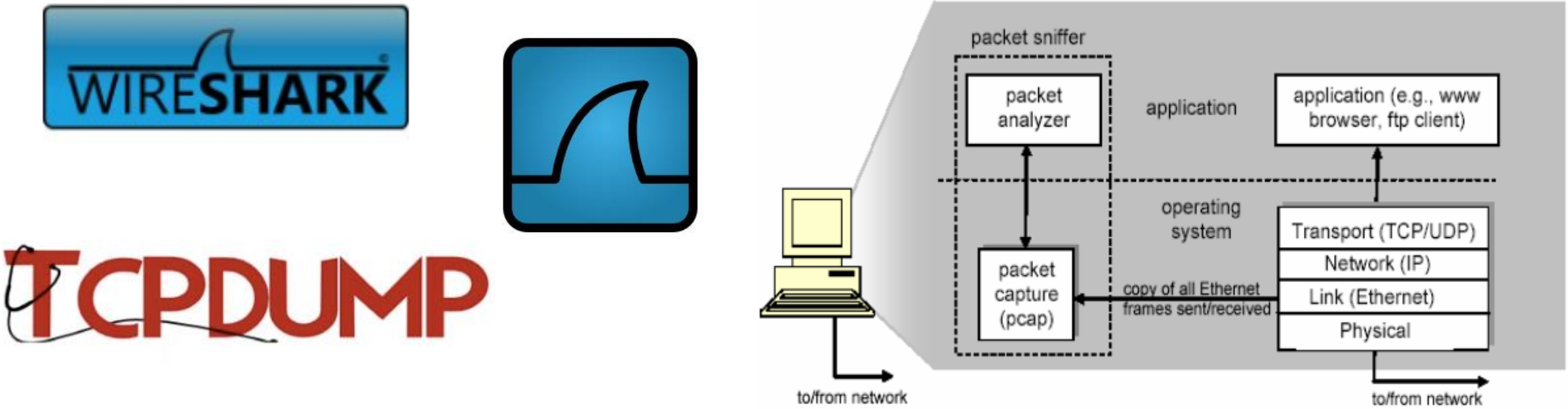
Ease of Deployment



Bare Metal Speed

Packet Analyzer

- Software that is used to capture and perform packet analysis of the network traffic
- Visualization of network bandwidth and resource utilization [5]



[5] https://en.wikipedia.org/wiki/Packet_analyzer

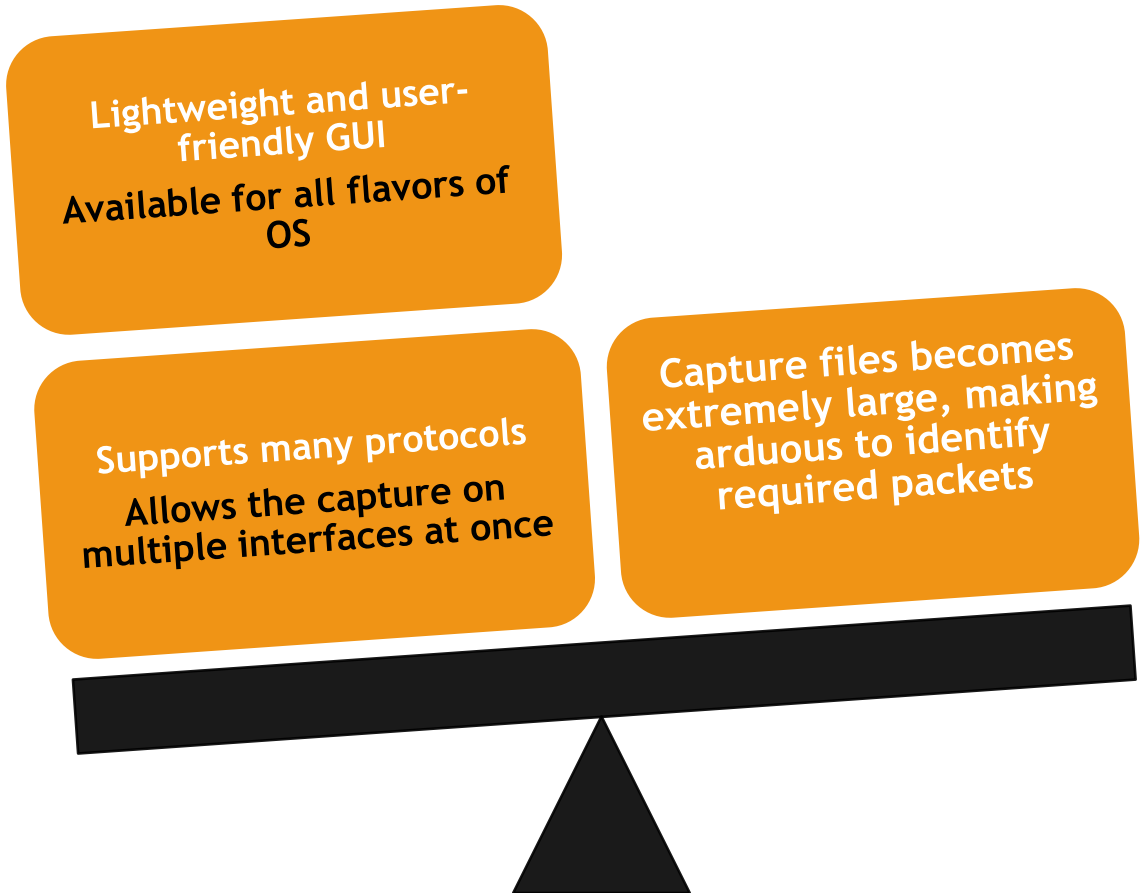
Baseline Design from CDD – Packet Analyzer

Comparison of Packet Analyzers

Parameters	Weights	Wireshark	Tshark	Tcpdump
Cost	0.20	**5**	5	5
Open-source	0.15	**5**	5	5
Feature Support	0.10	**4**	4	3
Programmability	0.15	**5**	5	4
Community Support & Documentation	0.15	**5**	4	4
Learning Curve	0.10	**4**	4	3
Modularity	0.15	**5**	5	4
Total	1.00	**4.80**	4.65	4.15

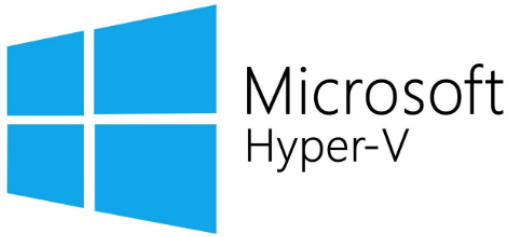
5 – Best choice for the Project
1 – Worst choice for the Project

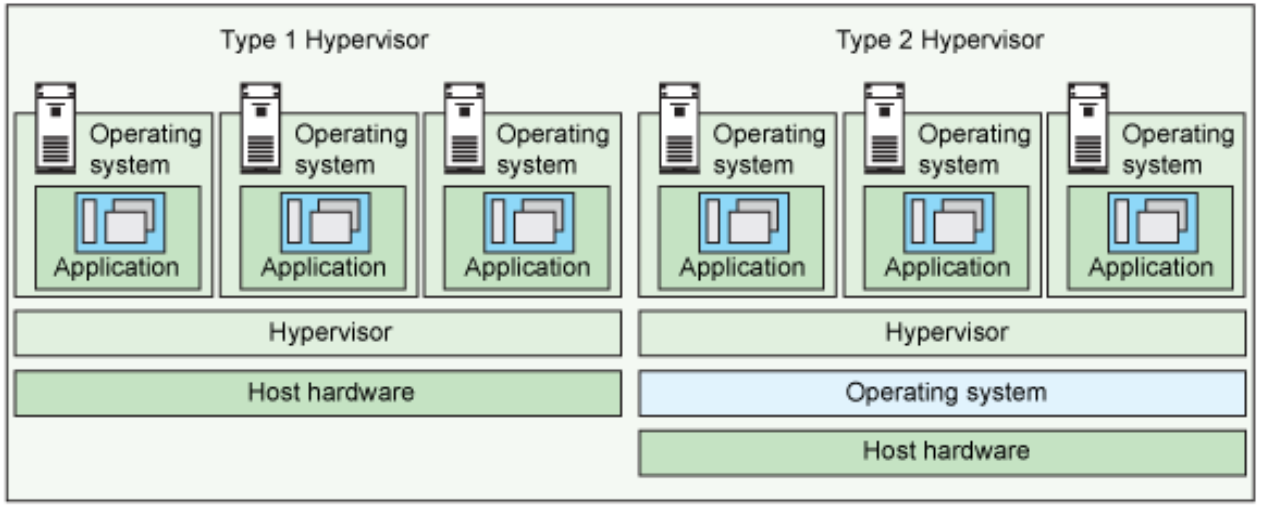
Pros & Cons of Wireshark Packet Analyzer



Hypervisors

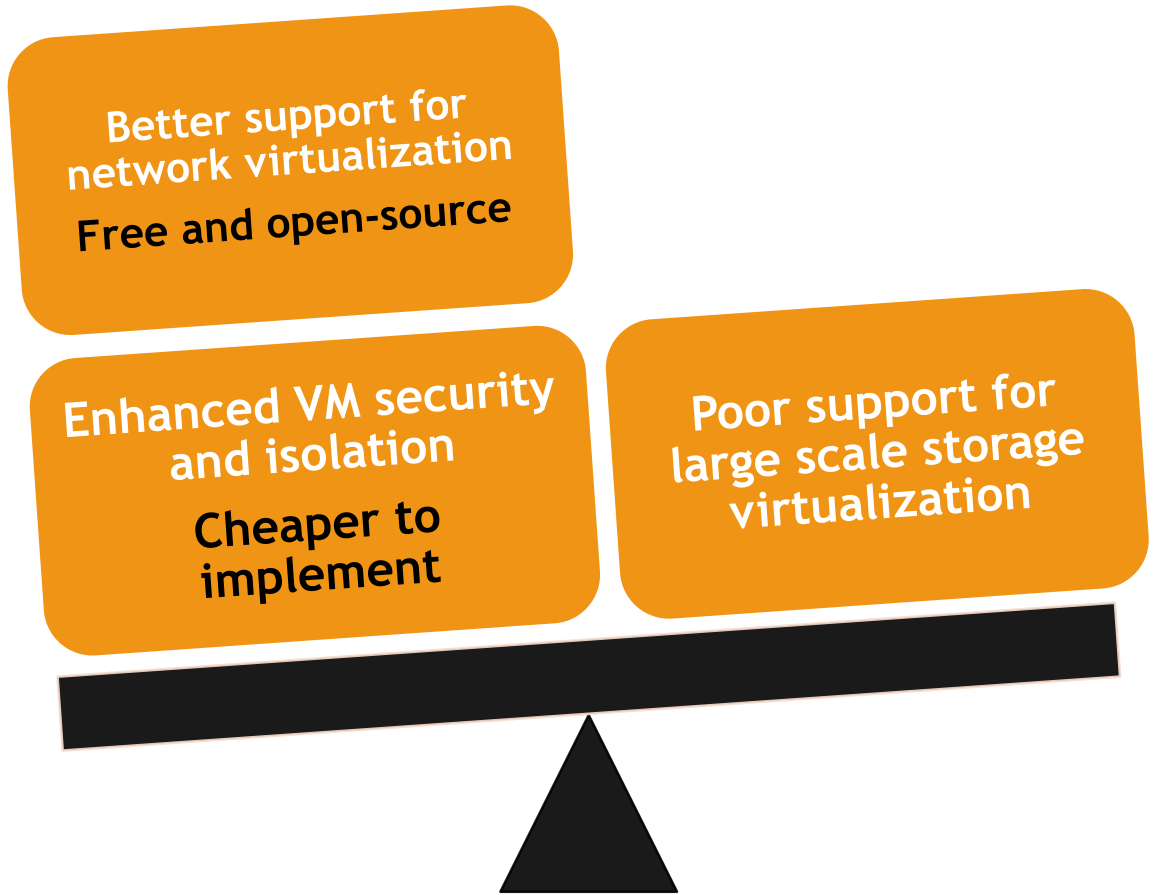
- Software that enables multiple Operating Systems to share the same hardware resources
- Controls the allocation of physical host hardware and kernel resources such as memory, disk space on each virtualized OS





Baseline Design from CDD – Hypervisors

Pros & Cons of KVM Hypervisor



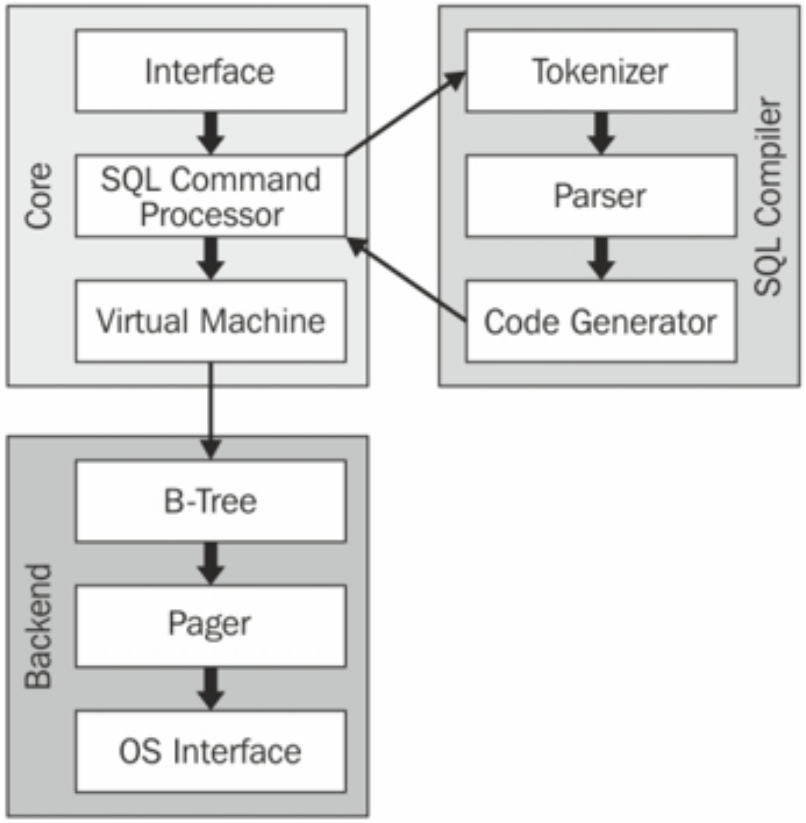
Comparison of Hypervisors

Parameters	Weights	ESXi	KVM	Hyper V	VMware Workstation	VirtualBox	QEMU
Cost	0.20	2	5	2	3	5	5
Open-source	0.15	4	4	2	4	4	4
Feature Support	0.10	3	4	3	4	4	3
Community Support & Documentation	0.15	3	4	3	4	3	3
Learning Curve	0.10	3	4	3	4	4	3
Modularity	0.15	2	4	2	4	4	3
Bare Metal Speed	0.15	4	4	4	2	2	2
Total	1.00	2.95	4.20	2.65	3.50	3.75	3.40

5 – Best choice for the Project
1 – Worst choice for the Project

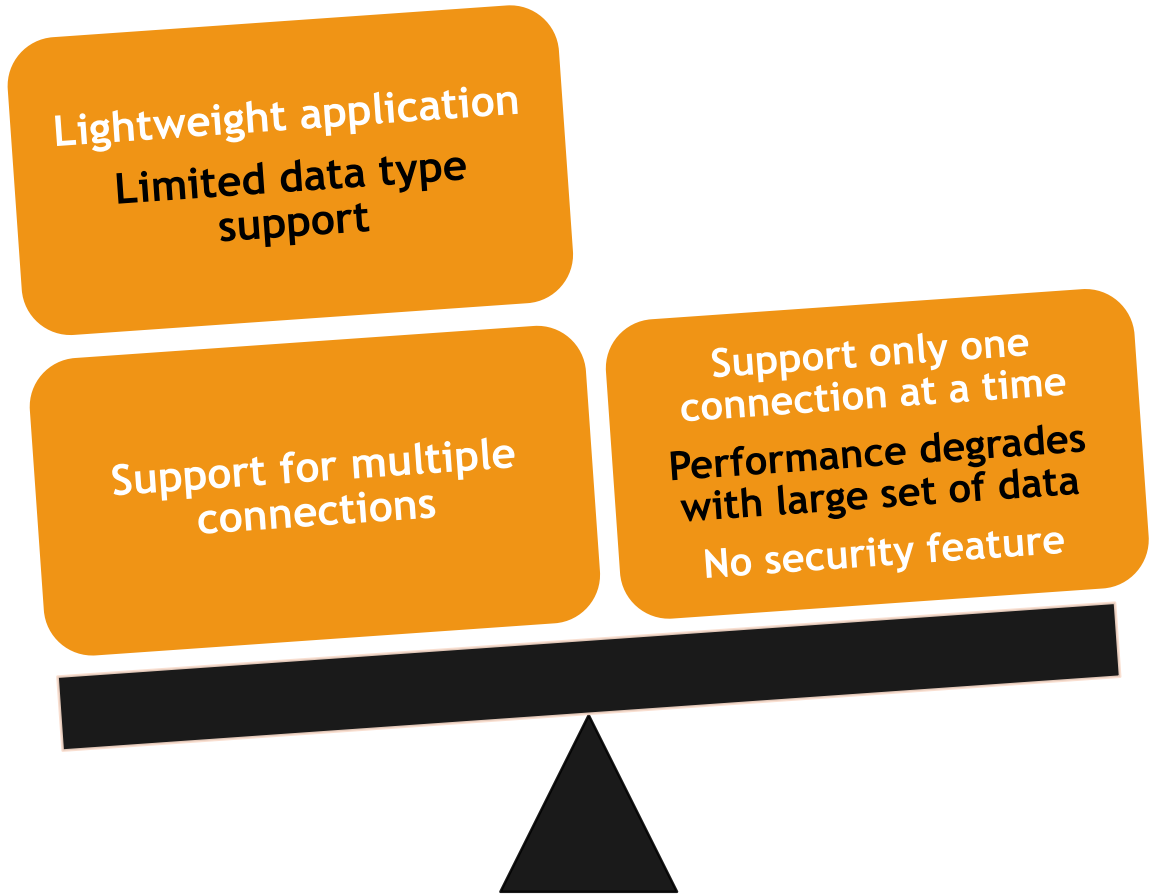
Database Storage

- Logical Structures: Used to store and index data
- Collection of data can be accessed, updated and analyzed using tools



Baseline Design from CDD – Database Storage

Pros & Cons of SQLite Database Storage



Comparison of Database Storages

Parameters	Weights	Oracle	MySQL	SQLite
Cost	0.20	1	5	**5**
Open-source	0.15	1	5	**5**
Feature Support	0.10	2	4	**3**
Programmability	0.15	3	4	**4**
Community Support & Documentation	0.15	3	4	**4**
Learning Curve	0.10	2	3	**4**
Modularity	0.15	2	4	**4**
Total	1.00	1.95	4.25	**4.25**

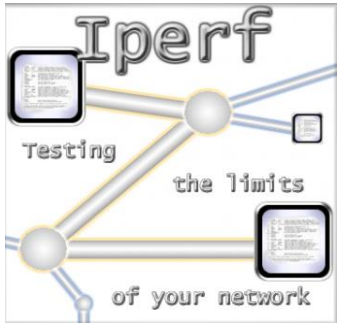
5 – Best choice for the Project
1 – Worst choice for the Project

Traffic Generator

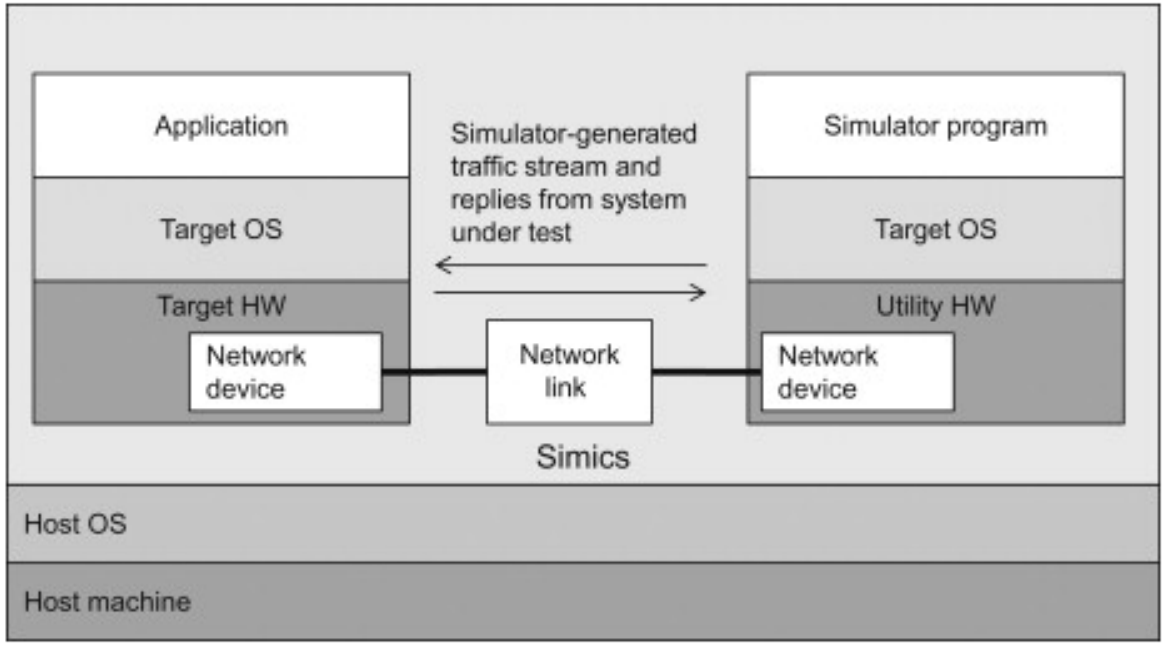
- Tool used to generate customized traffic and monitor network performance parameters
- Test the robustness of network under different network scenarios

OSTINATO

Network Traffic Generator and Analyzer



ixia



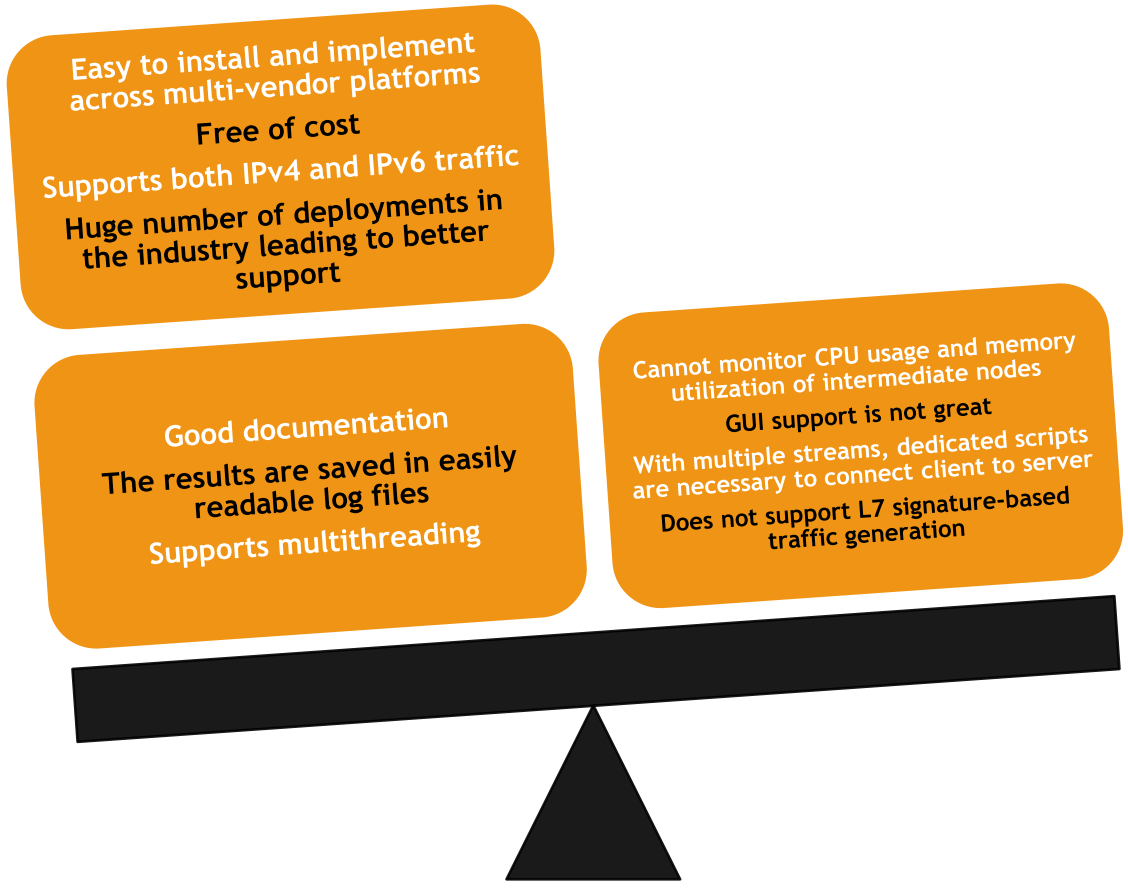
Baseline Design from CDD – Traffic Generator

Comparison of Traffic Generators

Parameters	Weights	Ostinato	Iperf	Ixia	WAN Killer
Cost	0.20	5	**5**	2	2
Open-source	0.15	4	**4**	2	2
Feature Support	0.10	2	**4**	3	2
Programmability	0.15	4	**4**	3	2
Community Support & Documentation	0.15	2	**4**	3	4
Learning Curve	0.10	2	**4**	2	2
Modularity	0.15	3	**4**	2	2
Total	1.00	3.35	**4.20**	2.40	2.30

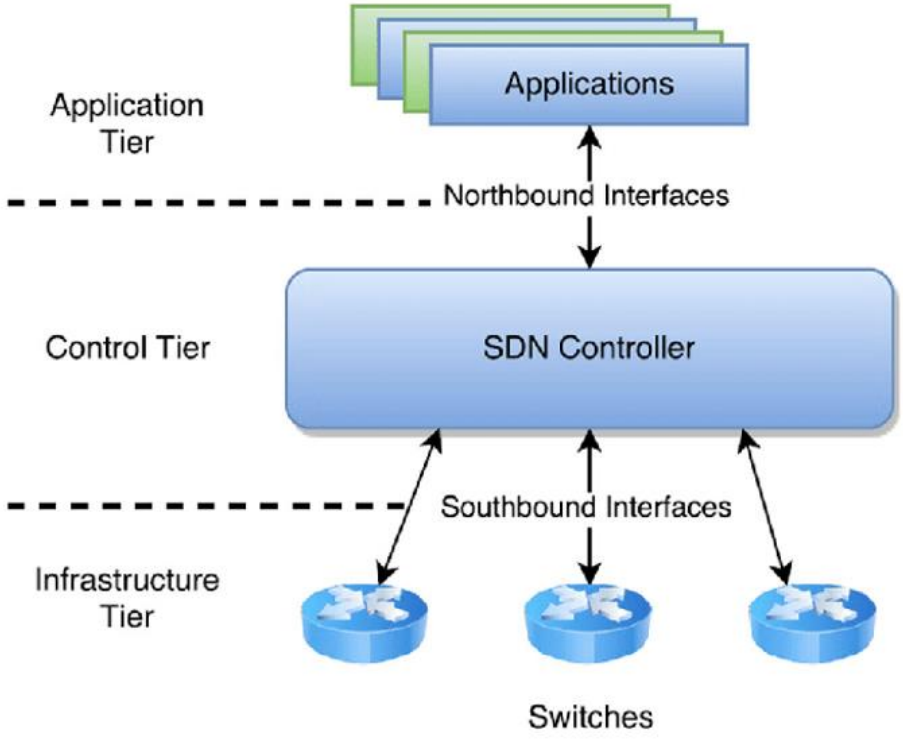
5 – Best choice for the Project
1 – Worst choice for the Project

Pros & Cons of Iperf Traffic Generator



SDN Controller

- Application layer in SDN: Responsible for provisioning, orchestration and abstraction of flows in the network devices
- Brain of the network
- Global view
- Open-source vs Commercial



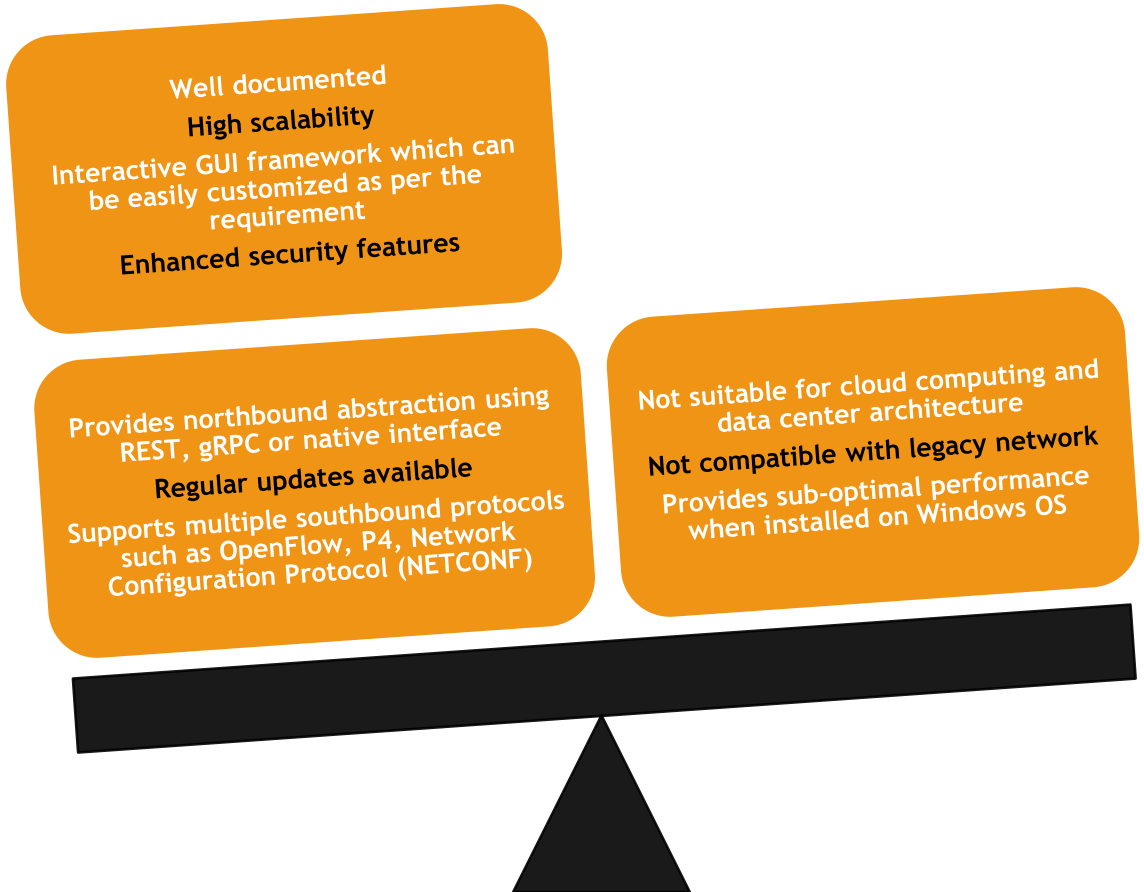
Baseline Design from CDD – SDN Controller

Comparison of SDN Controllers

Parameters	Weights	ONOS	ODL	Ryu	Floodlight	Cisco APIC
Cost	0.20	**5**	5	5	5	1
Open-source	0.15	**4**	3	4	4	1
Feature Support	0.10	**4**	2	3	4	2
Programmability	0.15	**5**	3	4	2	3
Community Support & Documentation	0.15	**4**	2	3	3	3
Learning Curve	0.10	**4**	2	3	3	3
Modularity	0.15	**4**	4	3	3	2
Total	1.00	**4.35**	3.20	3.70	3.50	2.05

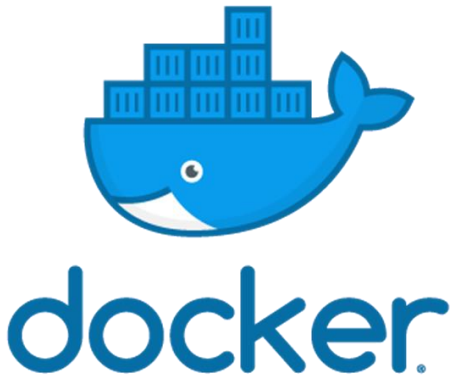
5 – Best choice for the Project
1 – Worst choice for the Project

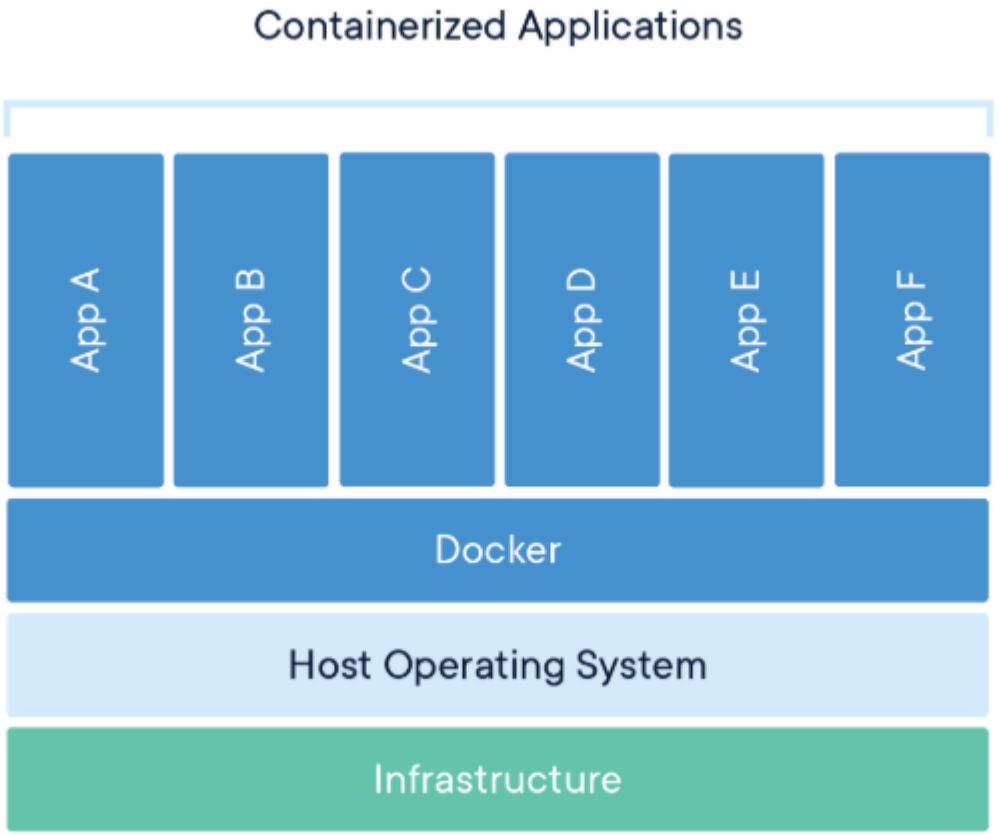
Pros & Cons of ONOS SDN Controller



Containers

- Standalone, executable software packages: Encases code, user libraries and other dependencies needed to run an application
- Easy to deploy
- Operates on Hosts Operating System





Baseline Design from CDD – Containers

Pros & Cons of Docker Containers

Comparison of Containers

Parameters	Weights	Docker	Rocket	Solaris Containers
Cost	0.20	**5**	5	1
Open-source	0.15	**4**	3	1
Compatibility with different OS	0.20	**4**	2	1
Community Support & Documentation	0.15	**4**	4	3
Learning Curve	0.10	**3**	4	4
Ease of Deployment	0.20	**4**	3	2
Total	1.00	**4.10**	3.45	1.80

5 – Best choice for the Project
1 – Worst choice for the Project

Validation and Verification for Hypervisor

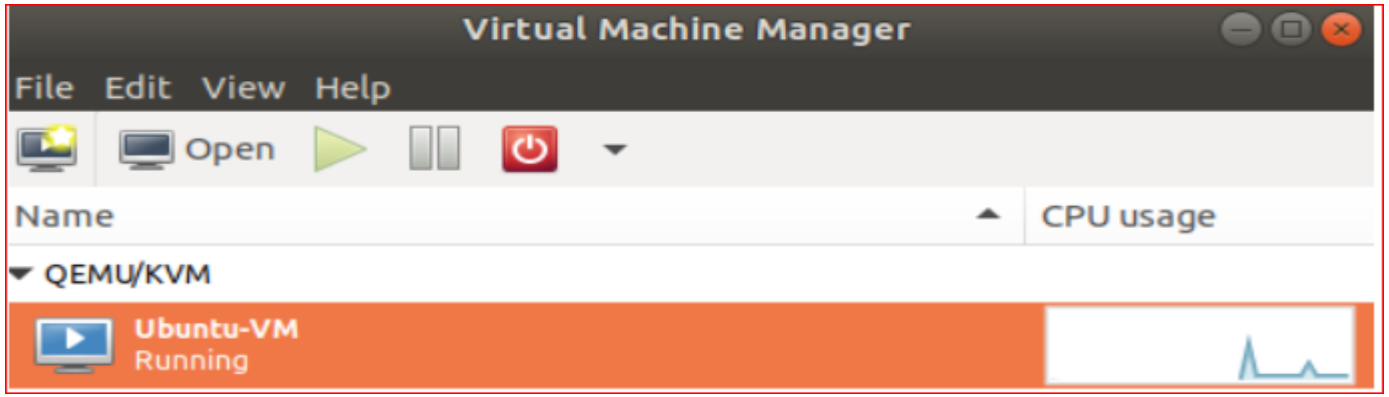
1. Hypervisor virtualization support

```
netman@netman-virtual-machine:~$ kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
```

2. Verify hypervisor installation

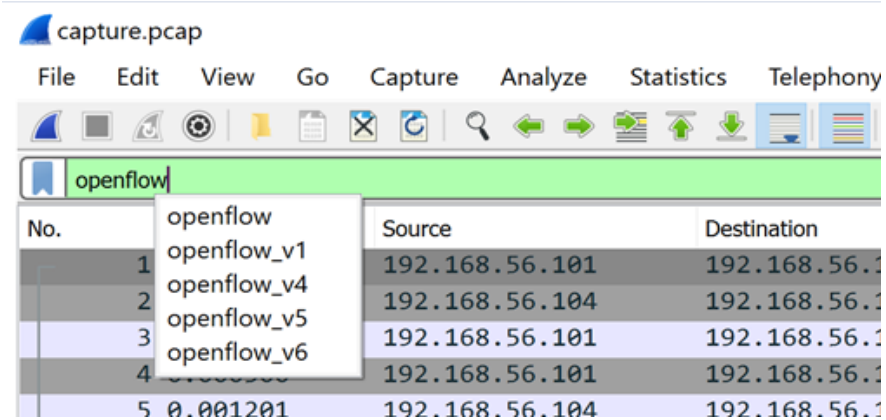
```
netman@netman-virtual-machine:~$ virsh list --all
 Id    Name                           State
----------------------------------------------------
```

3. Successful VM creation

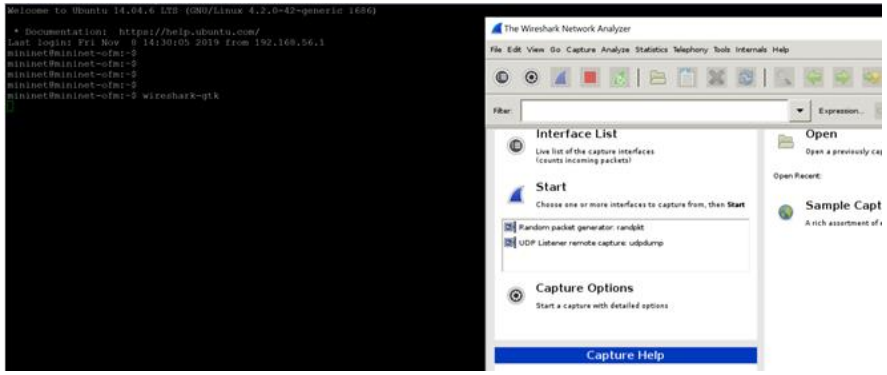


Validation and Verification for Packet Analyzer

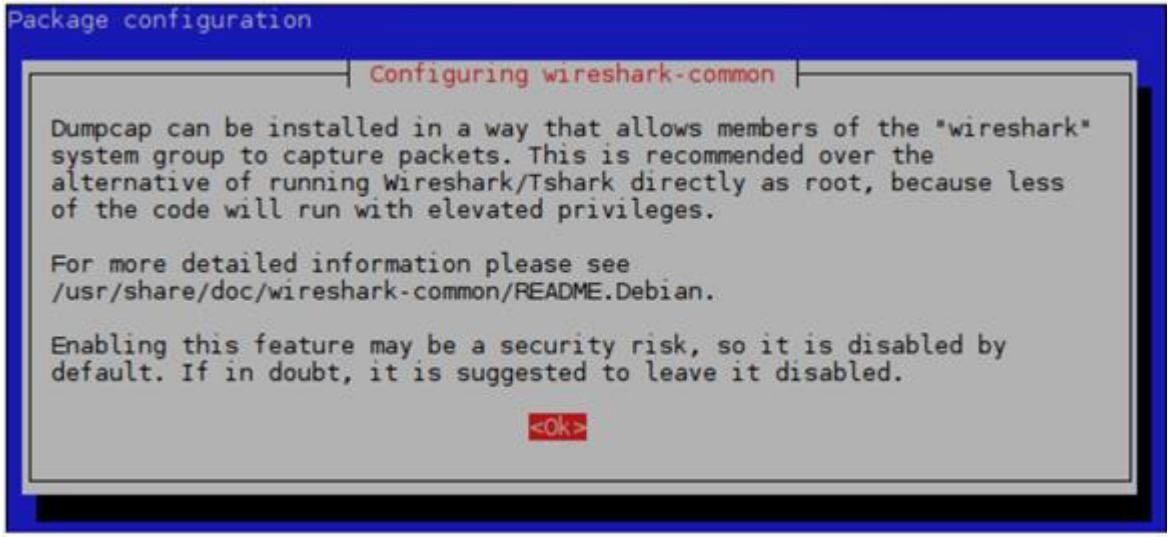
Supports various OpenFlow versions

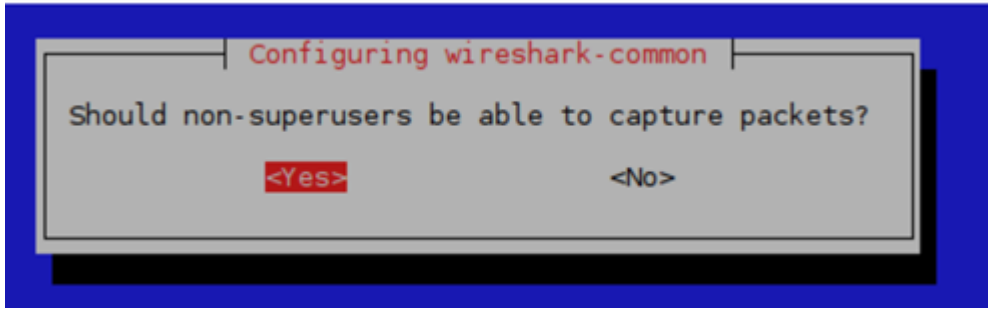


Ease to run application from the terminal, great for automation



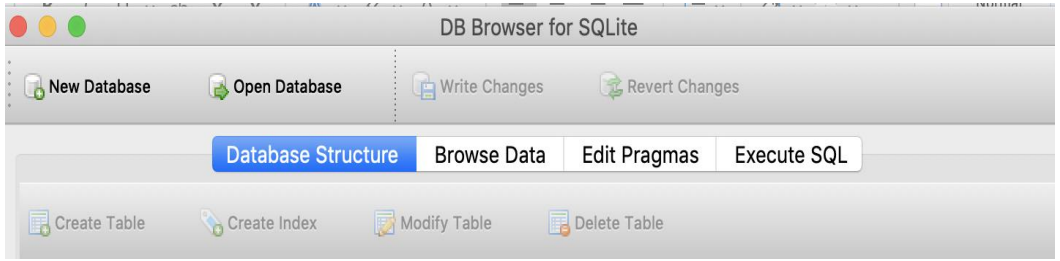
Superuser privilege to capture raw packets, hence more secure



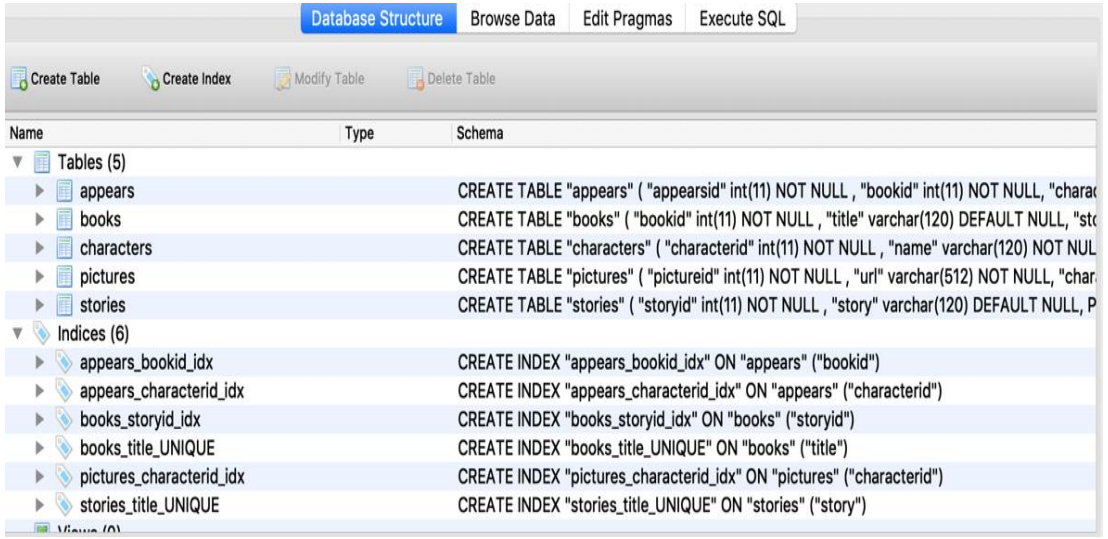


Validation and Verification for Database

User friendly GUI for easy management



Easy application setup and schema definitions



Risks and Mitigation - PDR

	Risk	Mitigation Action
1.	Controller Failure	Introduce a secondary controller
2.	Host Container Failure	Spin up a backup container on test VM
3.	Intermediate device failure	Use a wireless router for redundant connection between devices
4.	Traffic Congestion on Controller Link	Backup secondary link for test traffic
5.	Hypervisor Failure	Spin up a backup test bed on SDN Controller hardware
6.	Network Control Calls	Implement proactive flows.