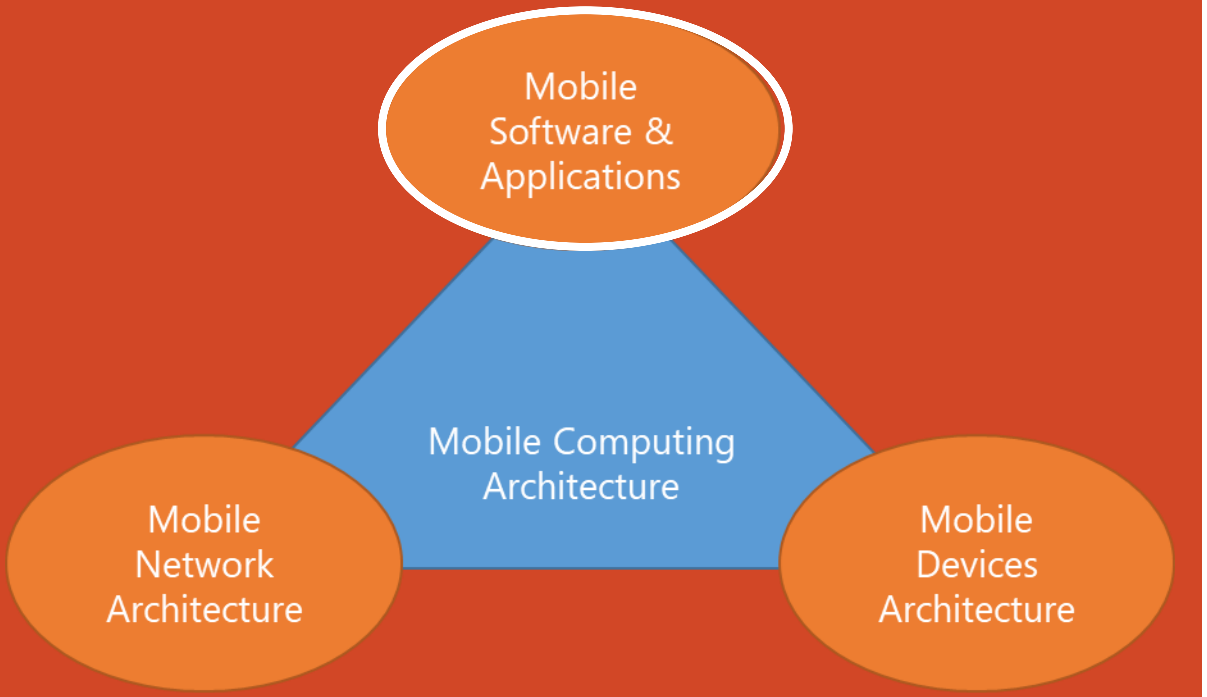
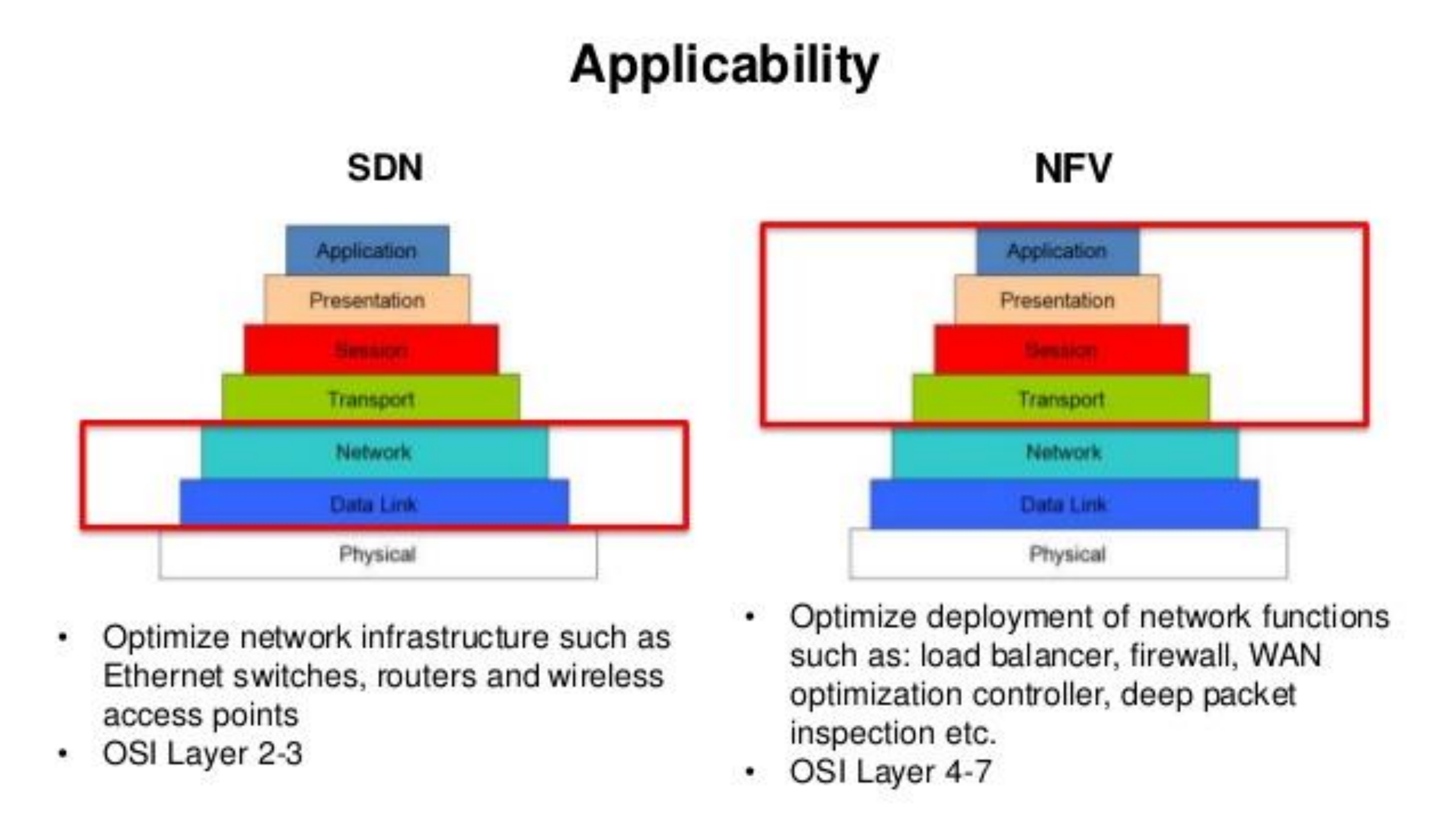
Mobile Computing Architecture

UW Bothell, WA

Lecture 8: Software Defined Networking (SDN)


Mobile Software & Applications
Mobile Computing Architecture
Mobile Network Architecture
Mobile Devices Architecture

Software Defined Networking and Network Functions Virtualization (NFV)



Applicability

SDN

- Optimize network infrastructure such as Ethernet switches, routers and wireless access points
- OSI Layer 2-3

NFV

- Optimize deployment of network functions such as: load balancer, firewall, WAN optimization controller, deep packet inspection etc.
- OSI Layer 4-7

SDN is many things to many people

- Split of Control and User plane – OpenFlow (initial definition)

- Software Defined Networks

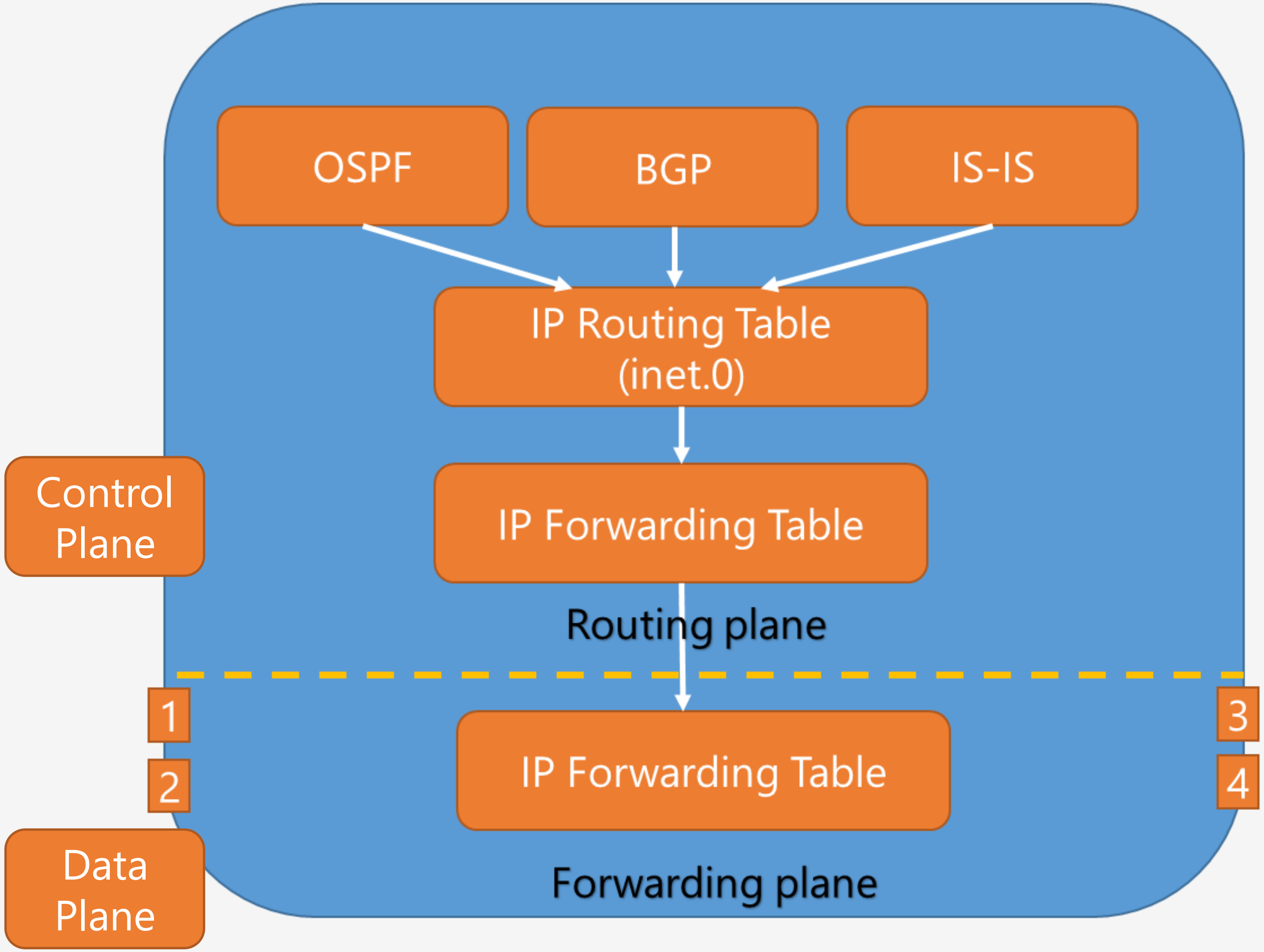
- Software Defined Infrastructure

- Software Defined Storage

- Software Defined xxxx

High Level Router Design

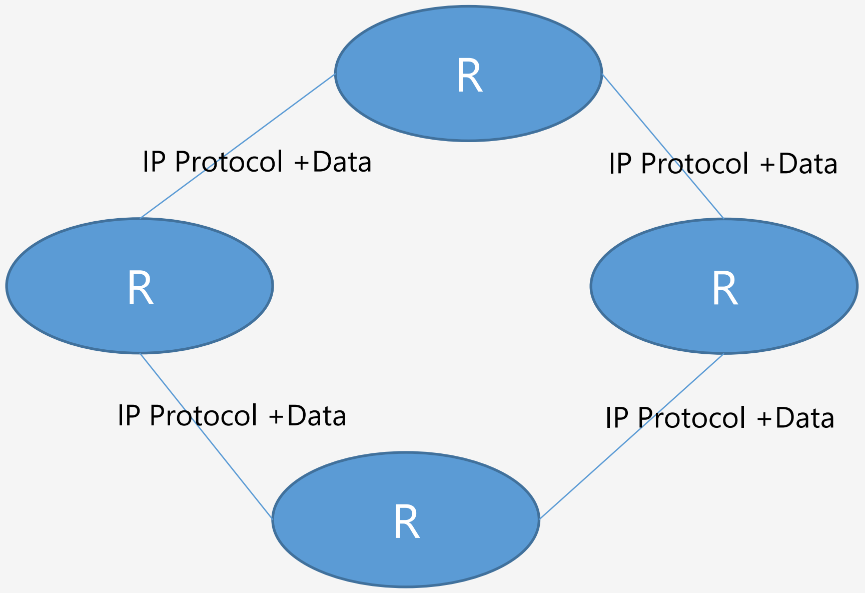
- A single box
- Routing (Control) and Forwarding (Data plane) tightly coupled
- Both planes incur development and manufacturing costs
- New routers are (typically) are purchased because of Data plane capacity ($$$)
- Traffic from different applications may need different treatment (priority) which may require network admins to reconfigure



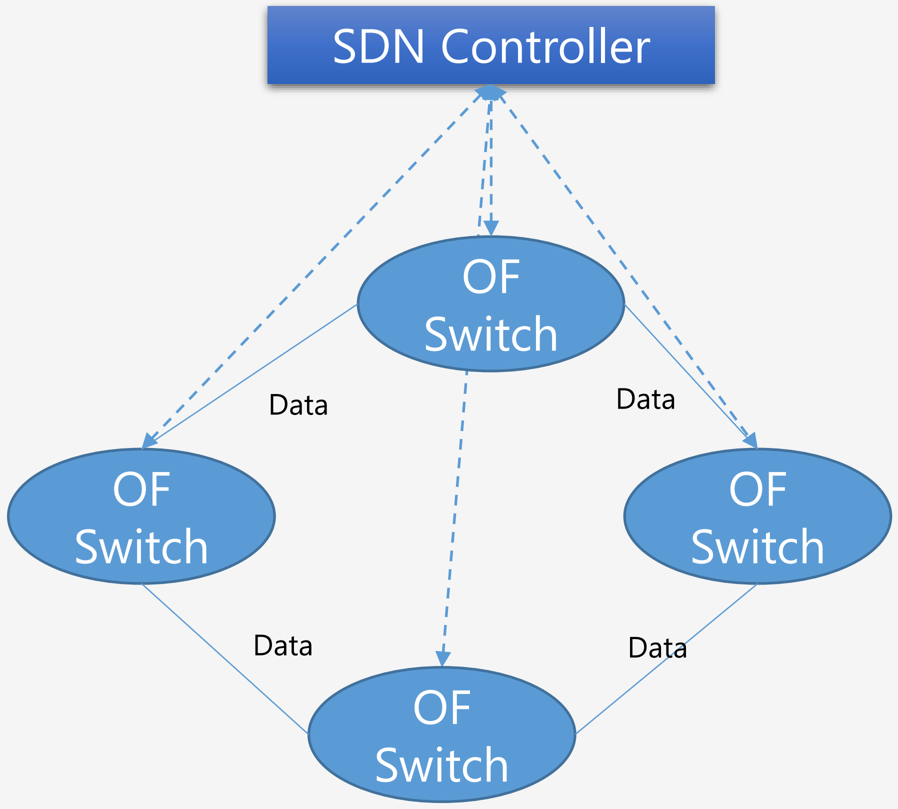
Sireen Malik. Mobile Computing Architecture, 2020

Software Defined Networking

Distributed IP Network



Centralized IP Network

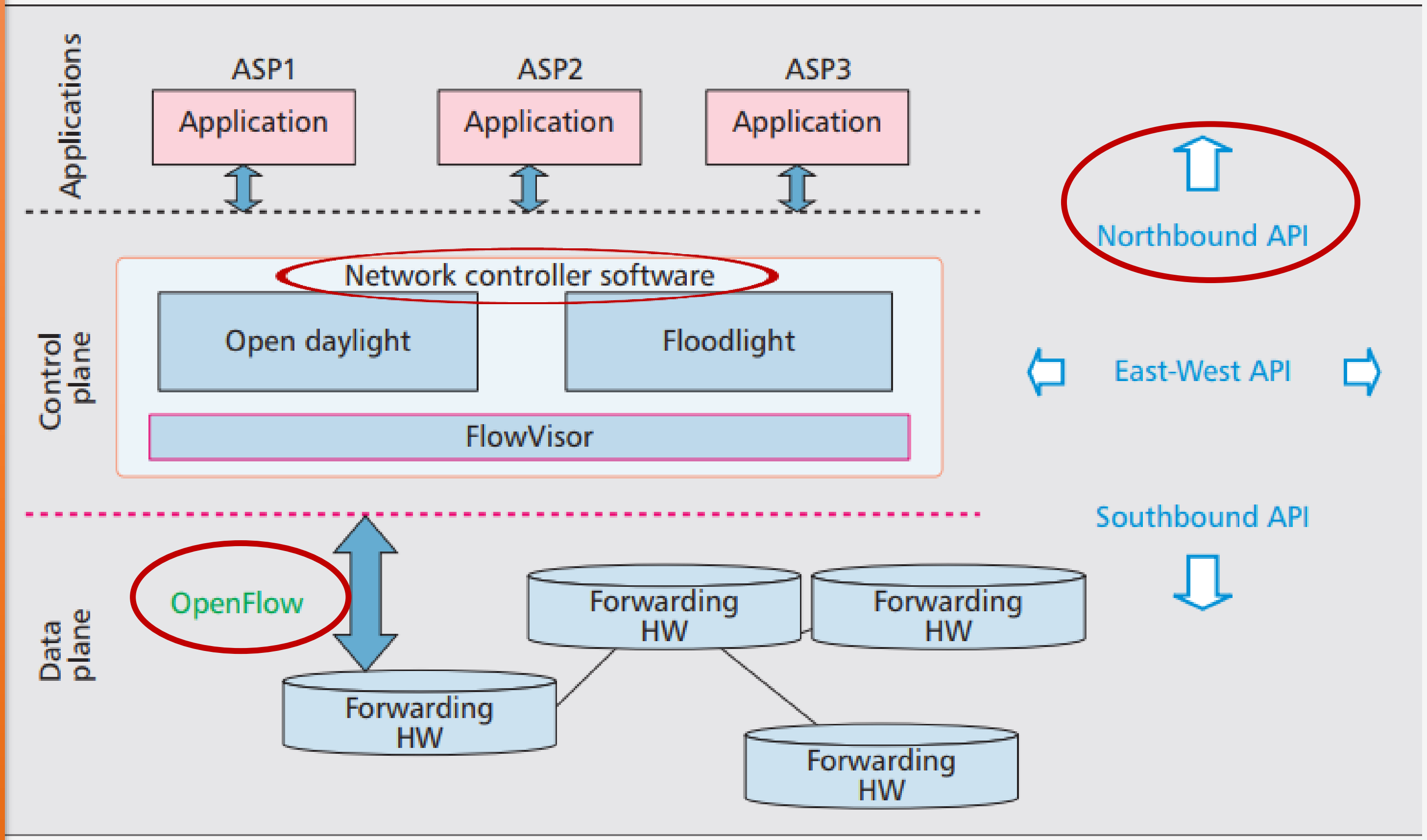


Software Defined Networking – OpenFlow | Open Daylight

SDN is defined by "the decoupling of control and packet forwarding planes in the network".

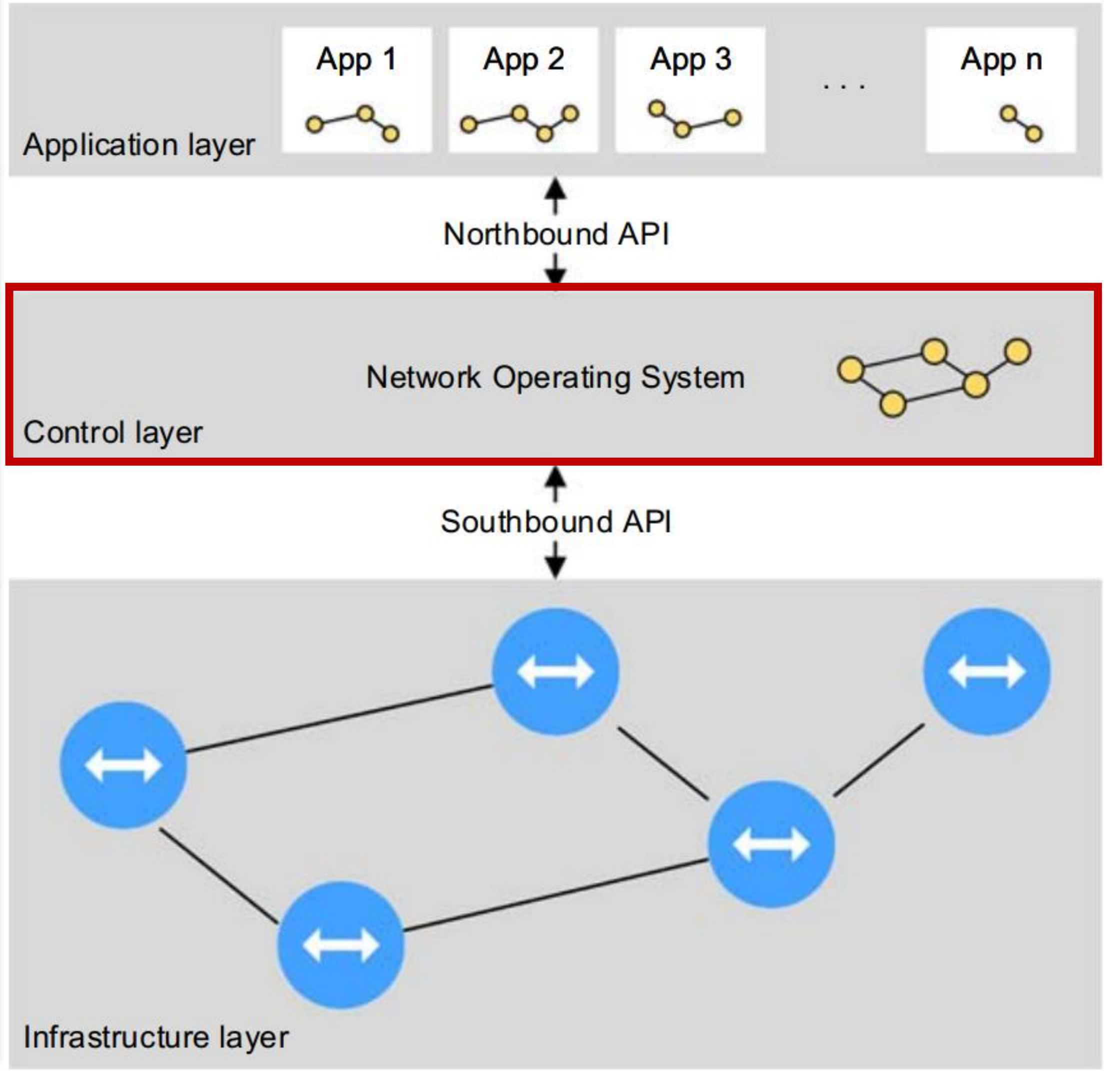
- It enables a network paradigm that enable Programmatic management, to directly connect to applications through application programming **interfaces (Northbound API**. SDN is used by enterprises to deploy their **applications faster while also cutting the overall deployment and operating costs.** IT administrators using SDN can manage and provision their network services from a **centralized point.**

- SDN decouples the network configuration and traffic engineering, separating them from their fundamental hardware infrastructure. This parting allows the use of **OpenFlow** and other open protocols. These open protocols can access network switches and routers that often use proprietary and otherwise closed firmware by applying globally aware software control at the network's edge.



Software Defined Networking – SDN Controller

- The controller takes the responsibility of establishing every flow in the network by installing flow entries on switch devices.
- Flow entries can be added to a data plane device in either a (1) proactive mode, where the flow rules are sent to the data plane devices as soon as the controller learns of it; or (2) reactive mode, where the controller sends flow entries to the data plane devices only as needed
- Controllers monitor the environment; thereby giving the controllers ability to have forwarding decisions integrated with real time traffic management.
- Southbound interface allows the controller to communicate, interact and manage the forwarding elements. **OpenFlow** is the most common implementation
- **OpenDaylight (ODL)** is an open-source SDN controller that has been available since 2014
- **OpenContrail** is a flavor of SDN controller originally from Juniper Networks. Etc.

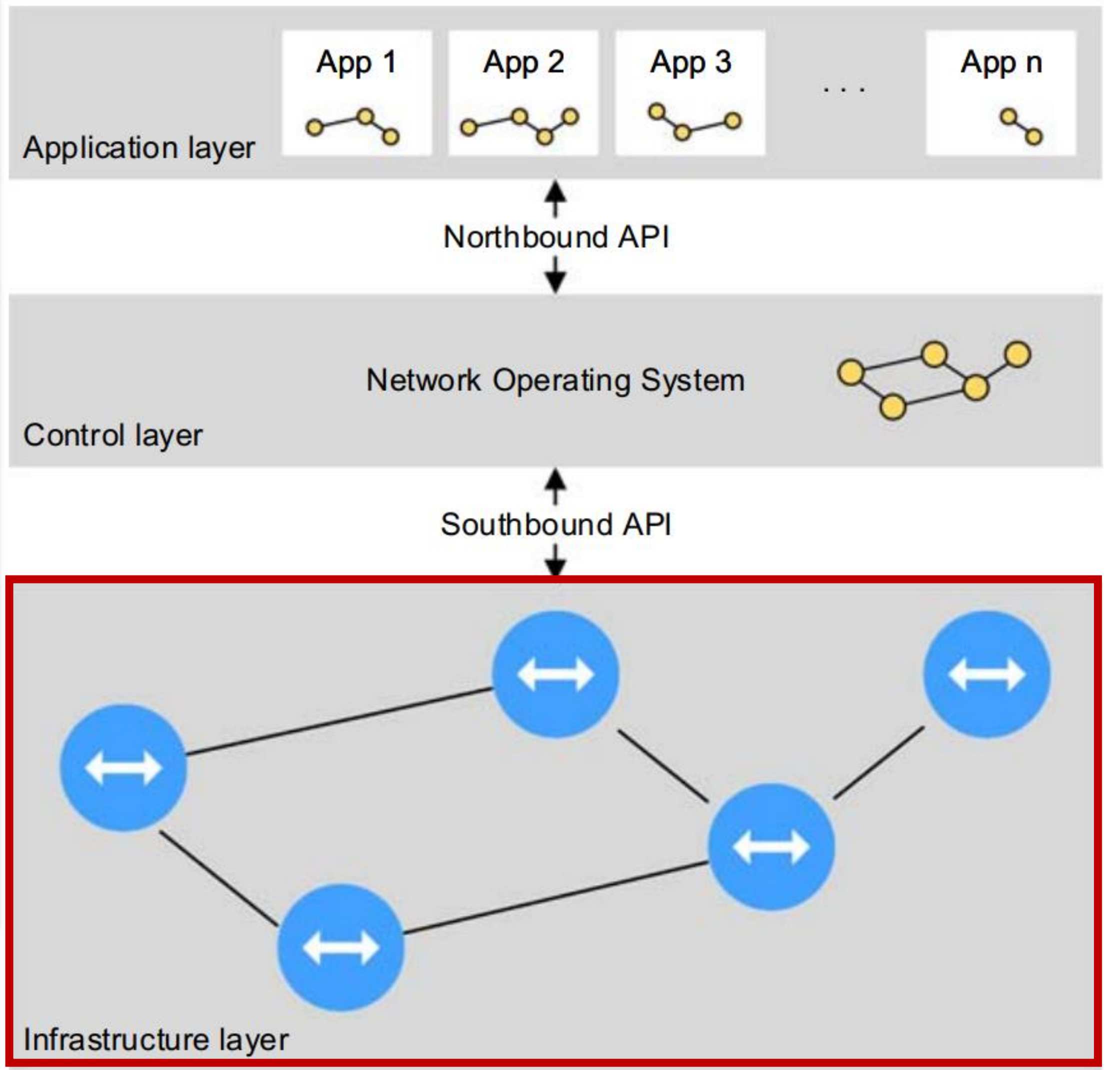


Book: Software-Defined Networking and Security

Sireen Malik. Mobile Computing Architecture, 2020

Software Defined Networking – SDN Data Plane

- The data plane in the SDN architecture is tasked with enabling the transfer of data from the sender to the receiver.
- Data plane itself does not generate or receive any data, but instead act as conduits for data.
- Data plane devices need to support a southbound API, to communicate with the controllers.
- Devices in the data plane come in two flavors:
  - Software-based, such as Open vSwitch; and
  - Hardware-based such as a OpenFlow enables HP switch. As can be envisaged, software-based devices have a more complete feature set, but are generally slower.



Book: Software-Defined Networking and Security

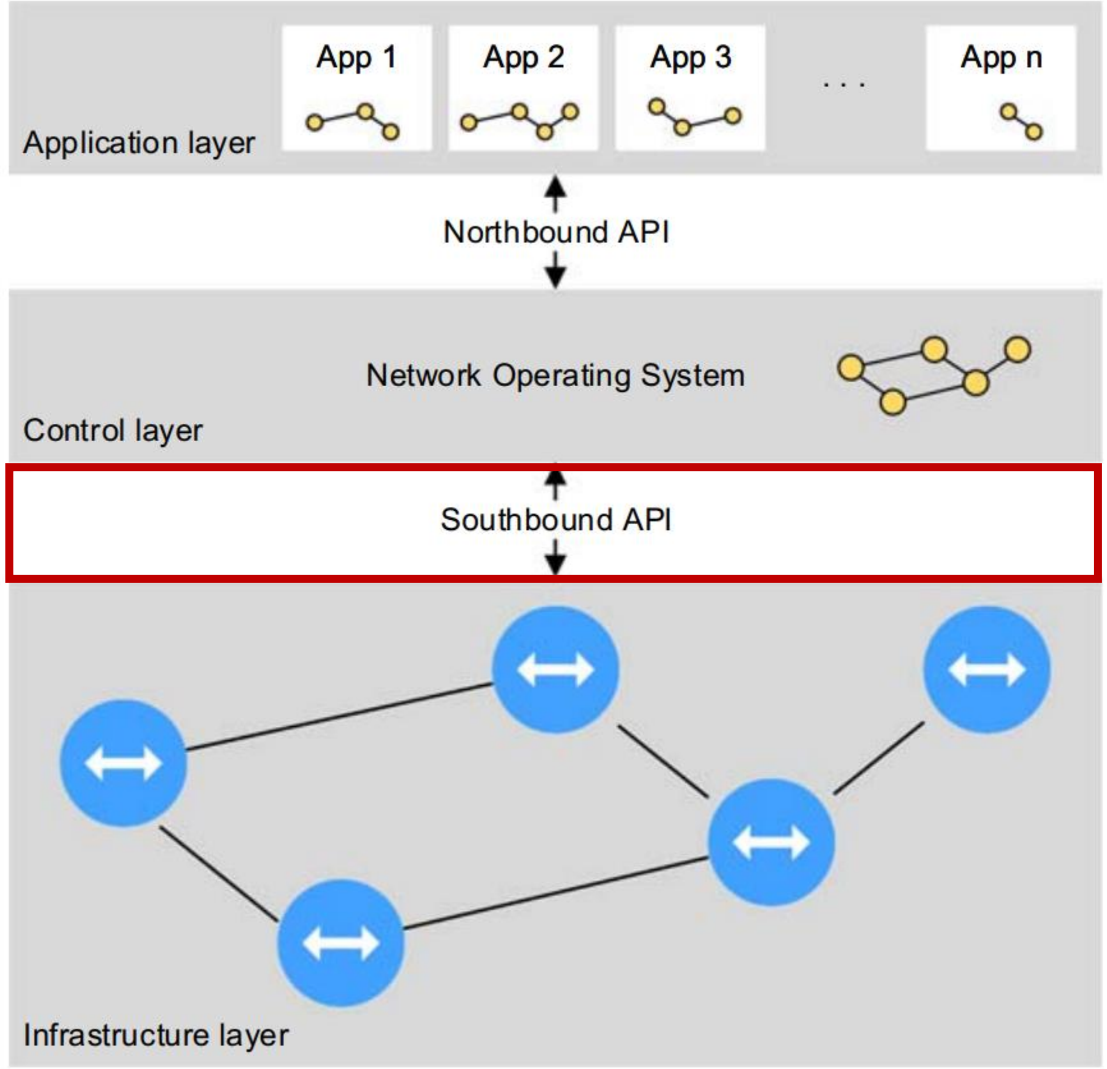
Software Defined Networking – OpenFlow

- OpenFlow, defined by the ONF [REF], is a protocol between the control and forwarding layers of an SDN architecture

- OpenFlow architecture consists of end hosts, a controller and OpenFlow enabled switches. TheSDN controller communicates with the switches using an OpenFlow API.

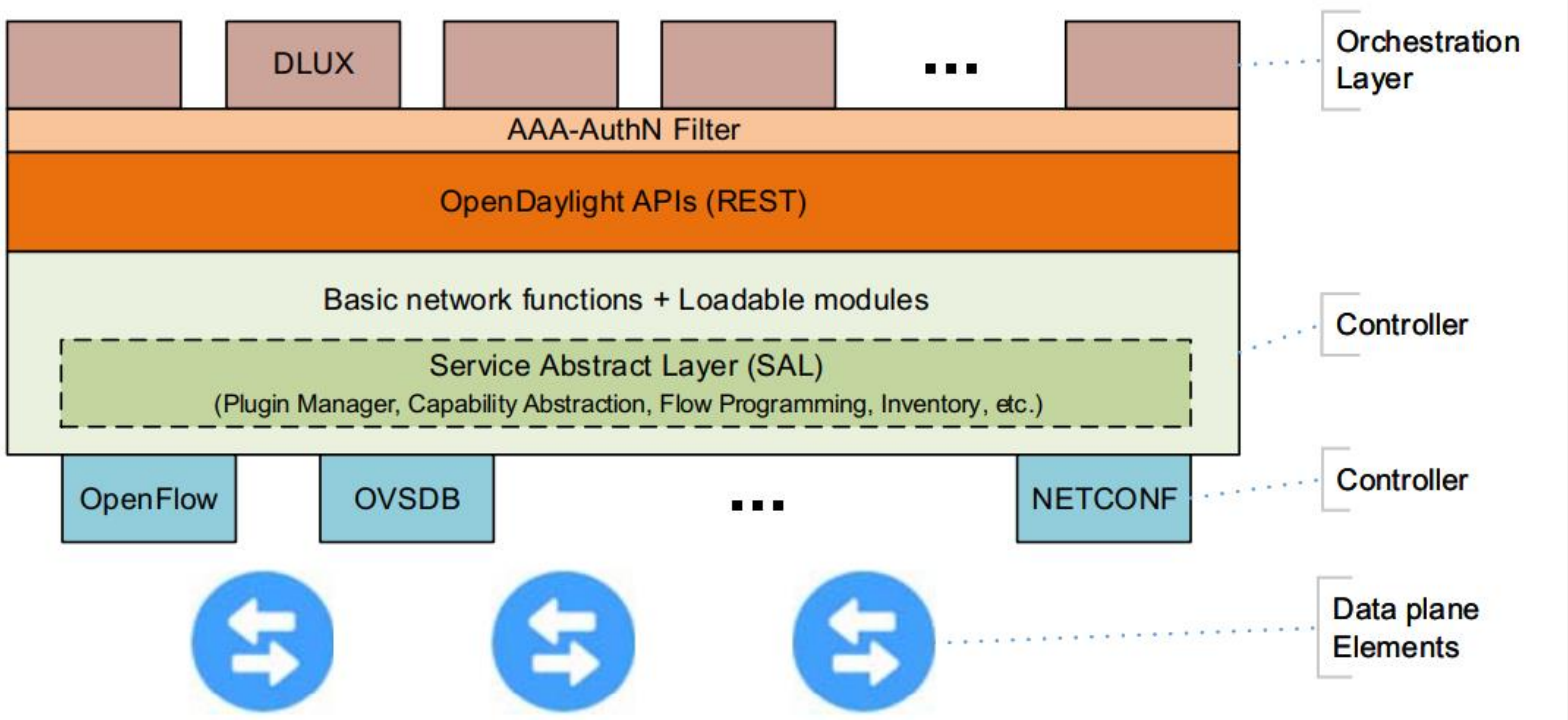
- When a packet arrives at an OpenFlow switch, packets are processed as follows:
    - Flow table lookup attempting to match the header fields of the packet in question to the local flow table is done and switched.

    - When multiple entries that match the incoming packet are present in the flow table, the packet with the highest priority is picked and switched



Book: Software-Defined Networking and Security

Software Defined Networking – Open DayLight (ODL) Architecture


DLUX
...
Orchestration Layer
AAA-AuthN Filter
OpenDaylight APIs (REST)
Basic network functions + Loadable modules
Controller
Service Abstract Layer (SAL)
(Plugin Manager, Capability Abstraction, Flow Programming, Inventory, etc.)
OpenFlow
OVSDB
...
NETCONF
Controller
Data plane Elements

Book: Software-Defined Networking and Security

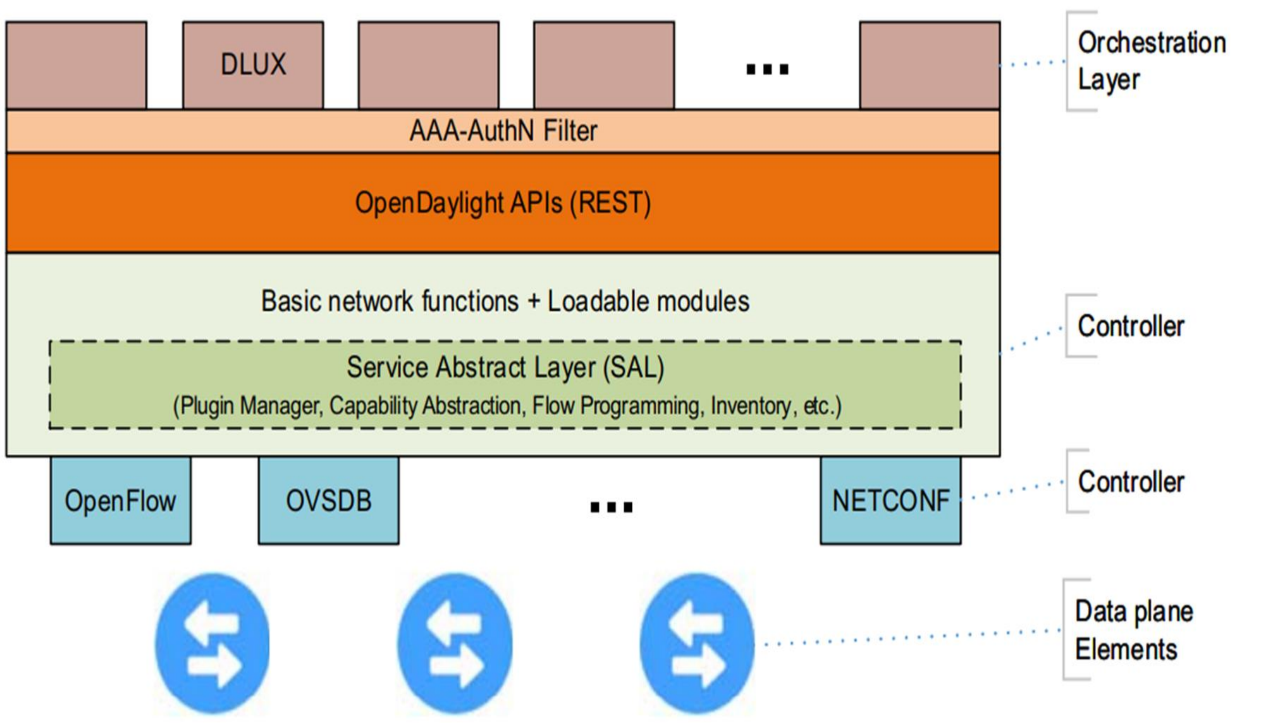
Software Defined Networking – Open DayLight (ODL)

- OpenDaylight (ODL) is an open-source project under the Linux Foundation

- Applications running on the ODL controller use a Service Abstraction Layer (SAL) to communicate with different types of devices using a variety of communication protocols, and provide RESTful APIs for use by external applications

- It uses **YANG data structures** along with shared data stores and messaging infrastructure to implement a Model Driven SAL approach

- ODL uses Apache Karaf as its container. Applications in Karaf are independent of each other, and can be started, stopped or restarted without affecting other applications. **NETCONF**, a RESTful API, is used to perform Create, Retrieve, Update and Delete (CRUD) operations, which itself is a means to configure network elements in a vendor-agnostic manner using the YANG modeling


DLUX
...
AAA-AuthN Filter
OpenDaylight APIs (REST)
Basic network functions + Loadable modules
Service Abstract Layer (SAL)
(Plugin Manager, Capability Abstraction, Flow Programming, Inventory, etc.)
OpenFlow
OVSDB
...
NETCONF
Orchestration Layer
Controller
Controller
Data plane Elements

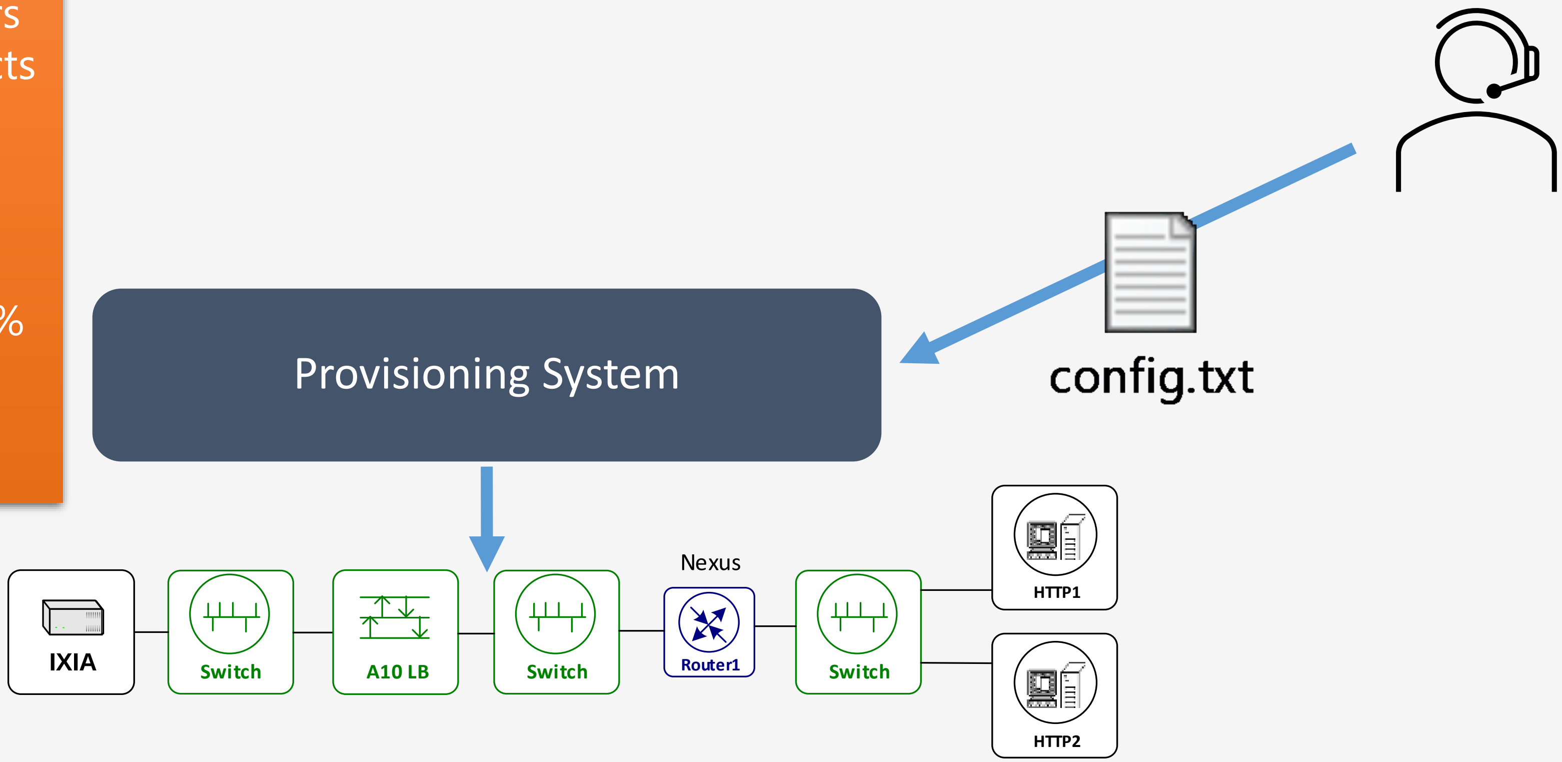
Book: Software-Defined Networking and Security

Software Defined Networking

(Without OpenFlow switches and SDN Controller. Deployed networks)

Manual Network Configuration

- Requires Vendor and Product specific expertise
  - Multiple vendors
  - Multiple products
  - Multiple Operating Systems
- Time consuming
- Error Prone (40%-50% issues)



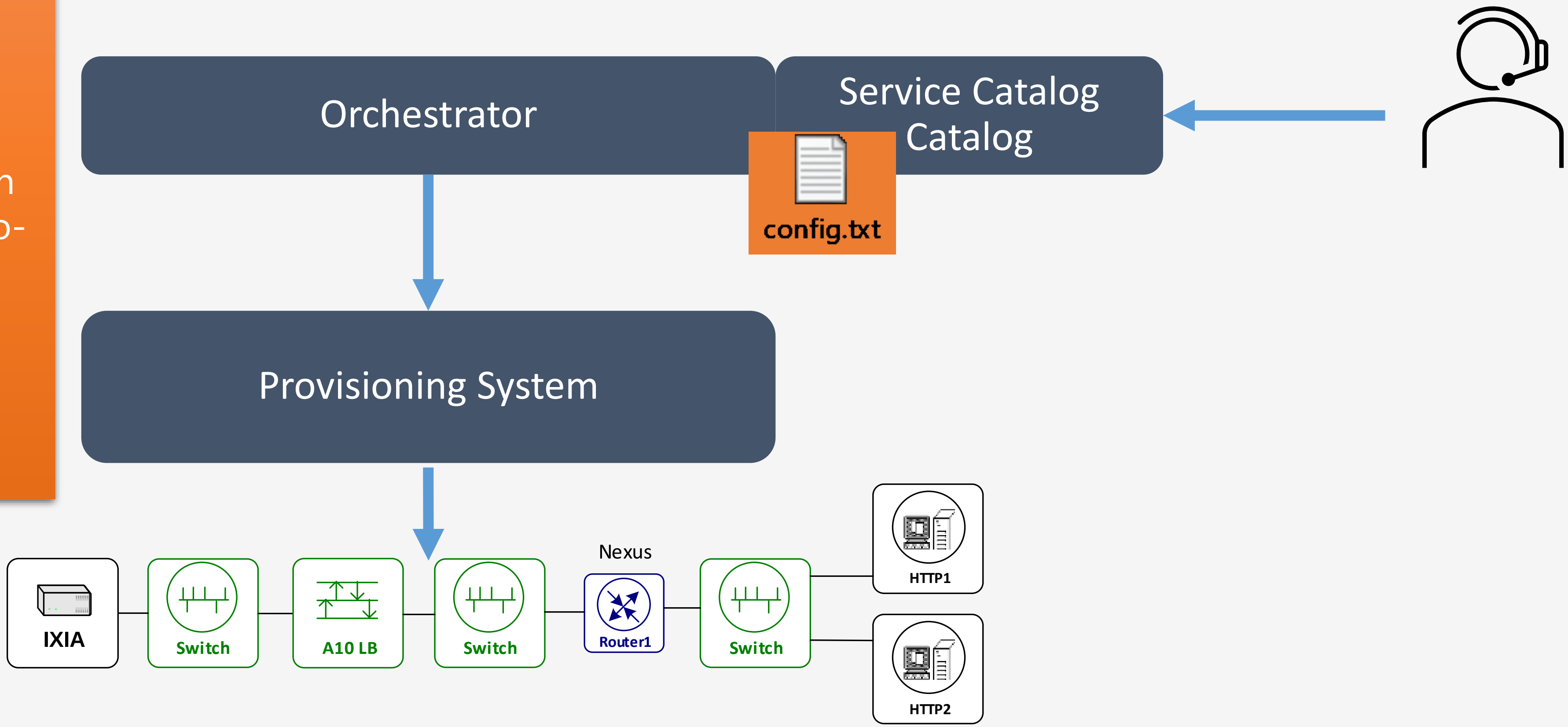
Provisioning System

config.txt

Network Configuration with Orchestrator

Orchestrator:

- Workflow automation
- Executes tasks and co-ordinate activities

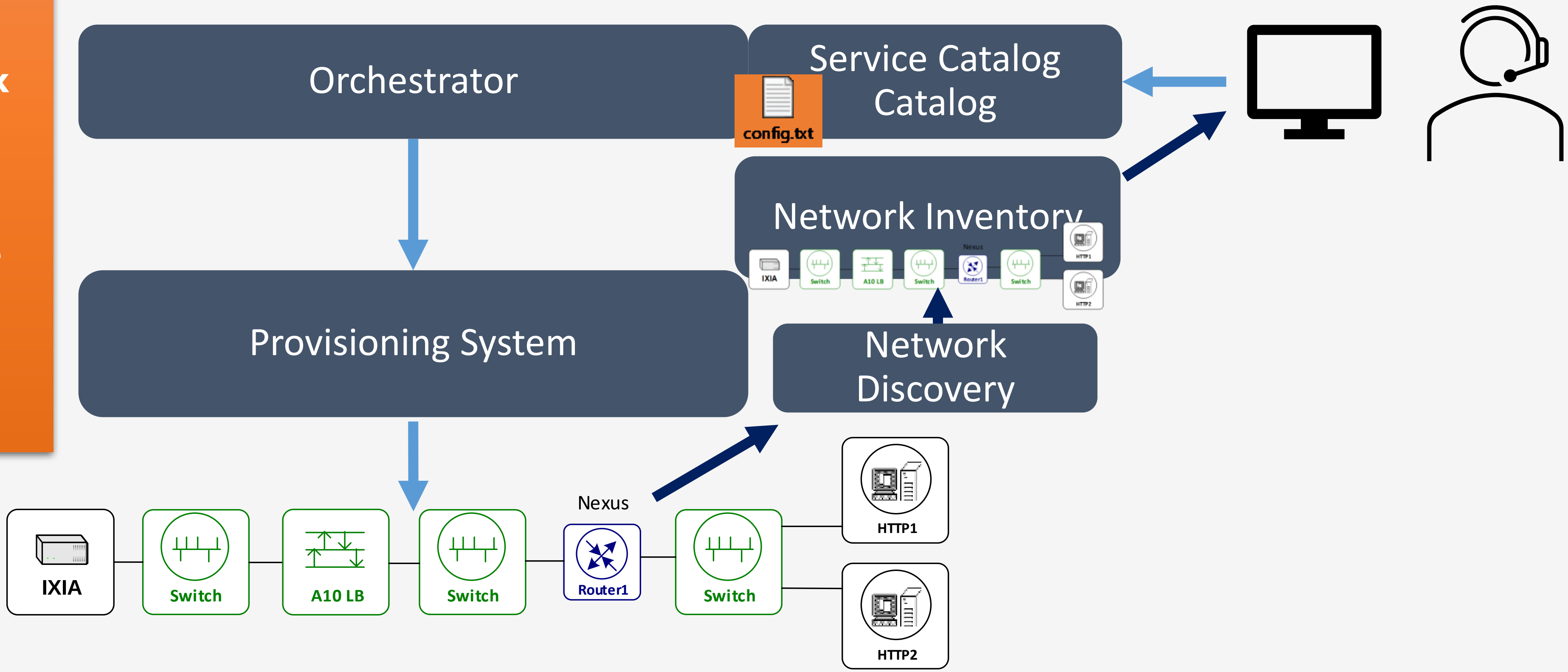


Network Configuration with Orchestrator + Inventory

How do we know what is present in the network?

=> We used a Network Inventory

- Maintains a digital representation of the network (using a Discovery function)



Orchestrator

Service Catalog Catalog

config.txt

Network Inventory

Provisioning System

Network Discovery

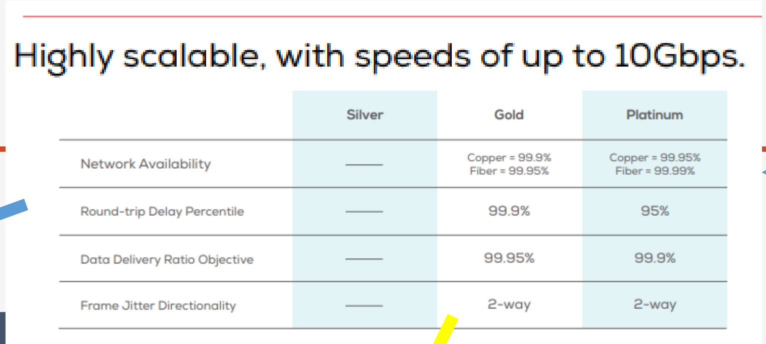
Sireen Malik, Mobile Computing Architecture, 2020

Network Configuration with Orchestrator + Inventory + Business Process Automation

Suppose we have to deploy an IP circuit between two locations with 100Mbps with resilience @ $100 per month

The business process is also automated

The product template is stored in Product Catalog

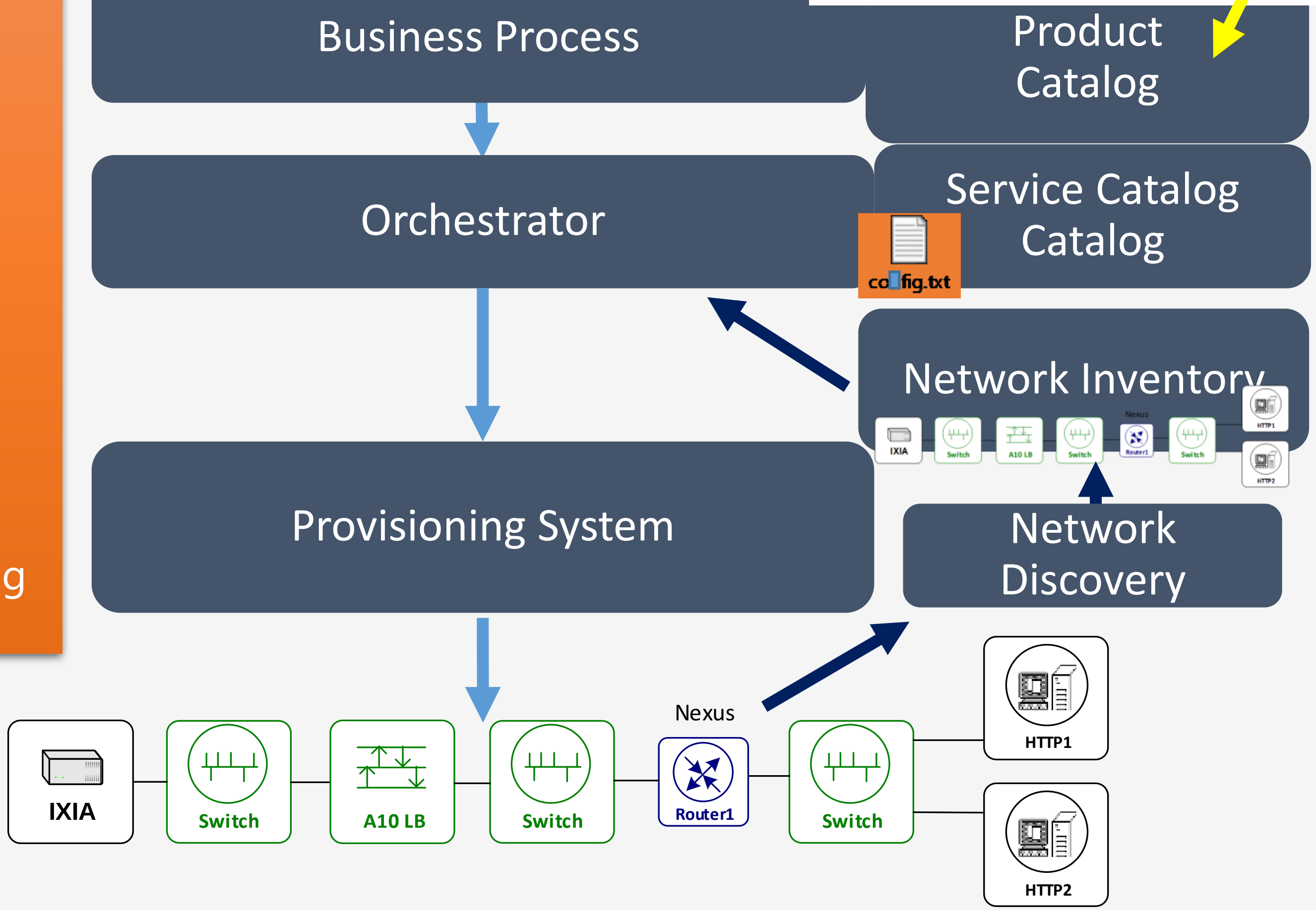


Highly scalable, with speeds of up to 10Gbps.

	Silver	Gold	Platinum

Network Availability	——	Copper = 99.9%
		Fiber = 99.95%
			Copper = 99.95%
			Fiber = 99.99%

Round-trip Delay Percentile	——	99.9%	95%
Data Delivery Ratio Objective	——	99.95%	99.9%
Frame Jitter Directionality	——	2-way	2-way



SDN Stack

Increased Time to Market

Higher Quality

Repeatable tasks

Reduced complexity for Customer Service

Business Process

Product Catalog

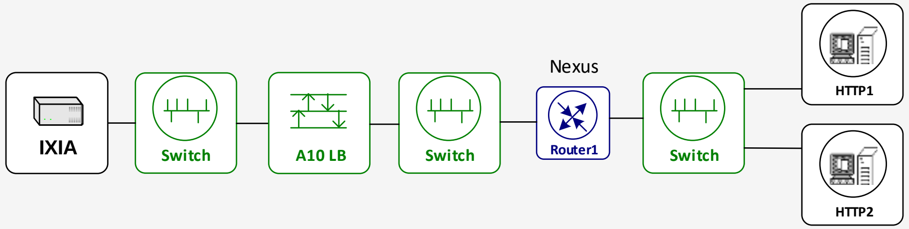
Orchestrator

Service Catalog Catalog

Network Inventory

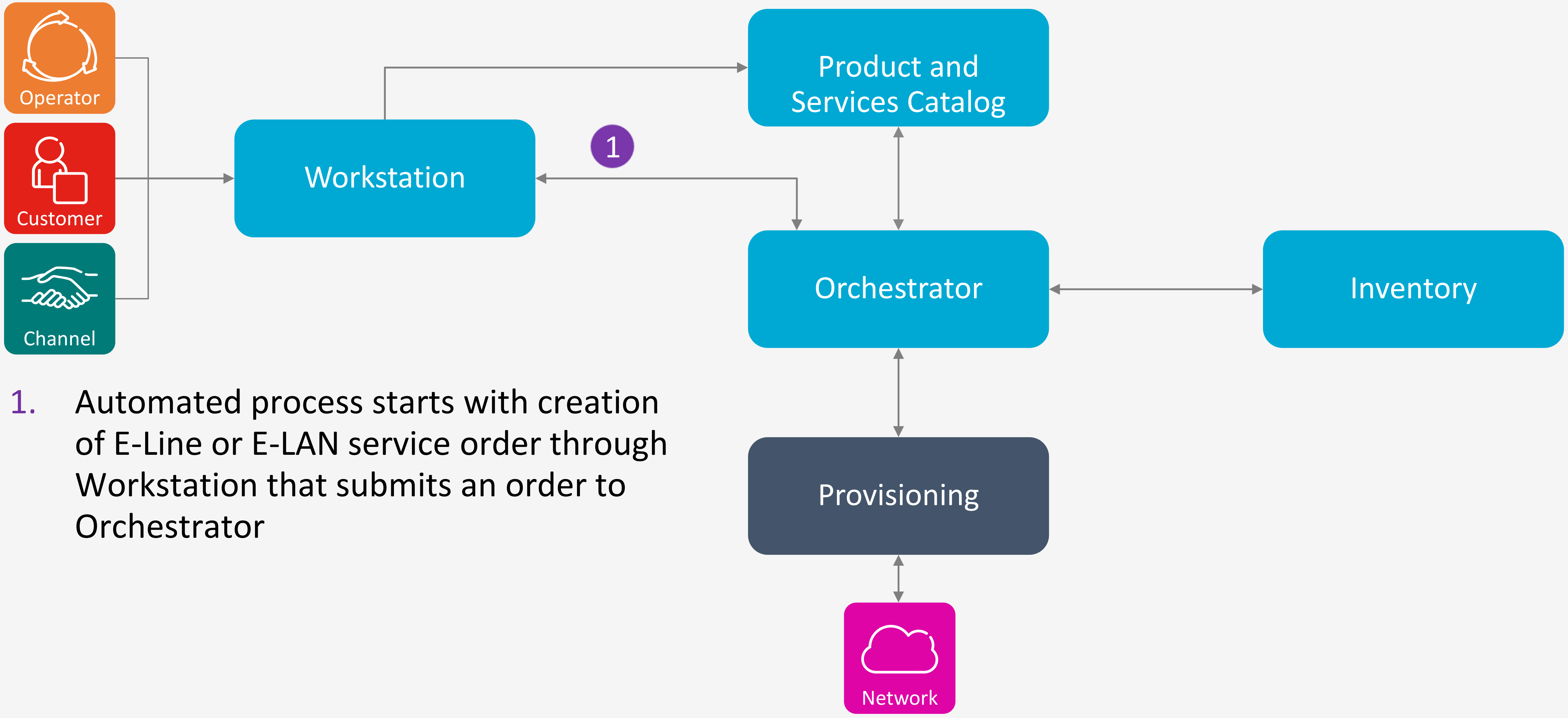
Provisioning System

Network Discovery



Automated Lead-to-Service

Add Flow Example for Carrier Ethernet

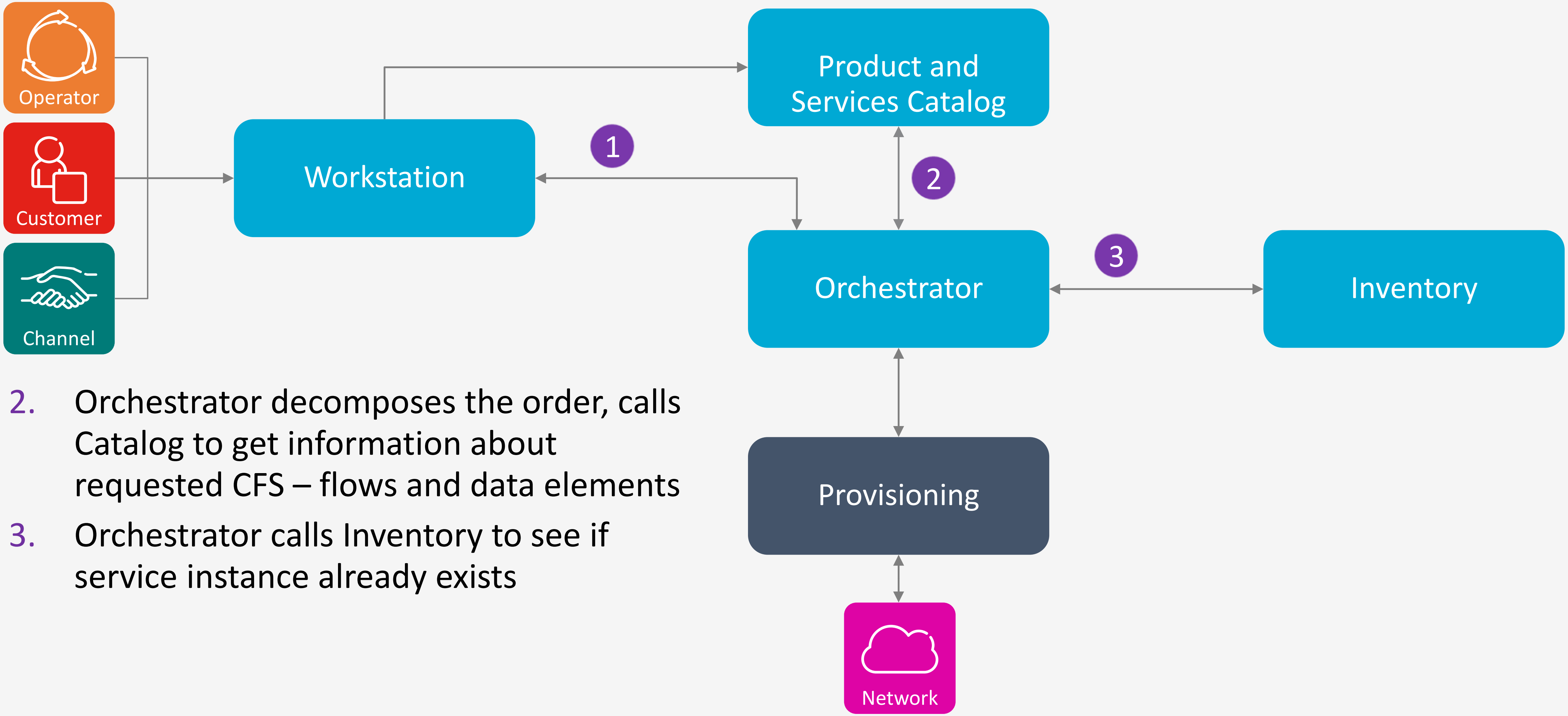


1. Automated process starts with creation of E-Line or E-LAN service order through Workstation that submits an order to Orchestrator

Sireen Malik, Mobile Computing Architecture, 2020

Automated Lead-to-Service

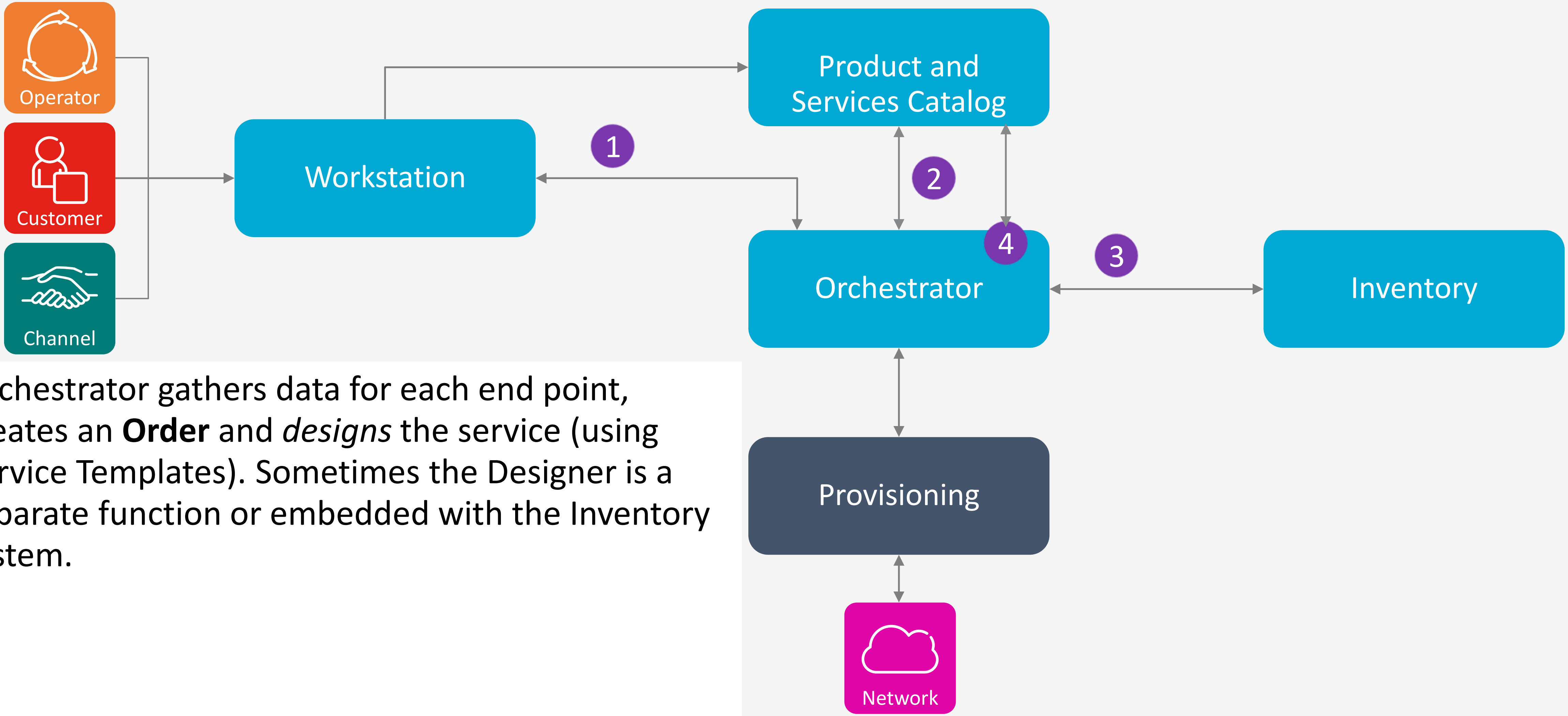
Add Flow Example for Carrier Ethernet



2. Orchestrator decomposes the order, calls Catalog to get information about requested CFS – flows and data elements
3. Orchestrator calls Inventory to see if service instance already exists

Automated Lead-to-Service

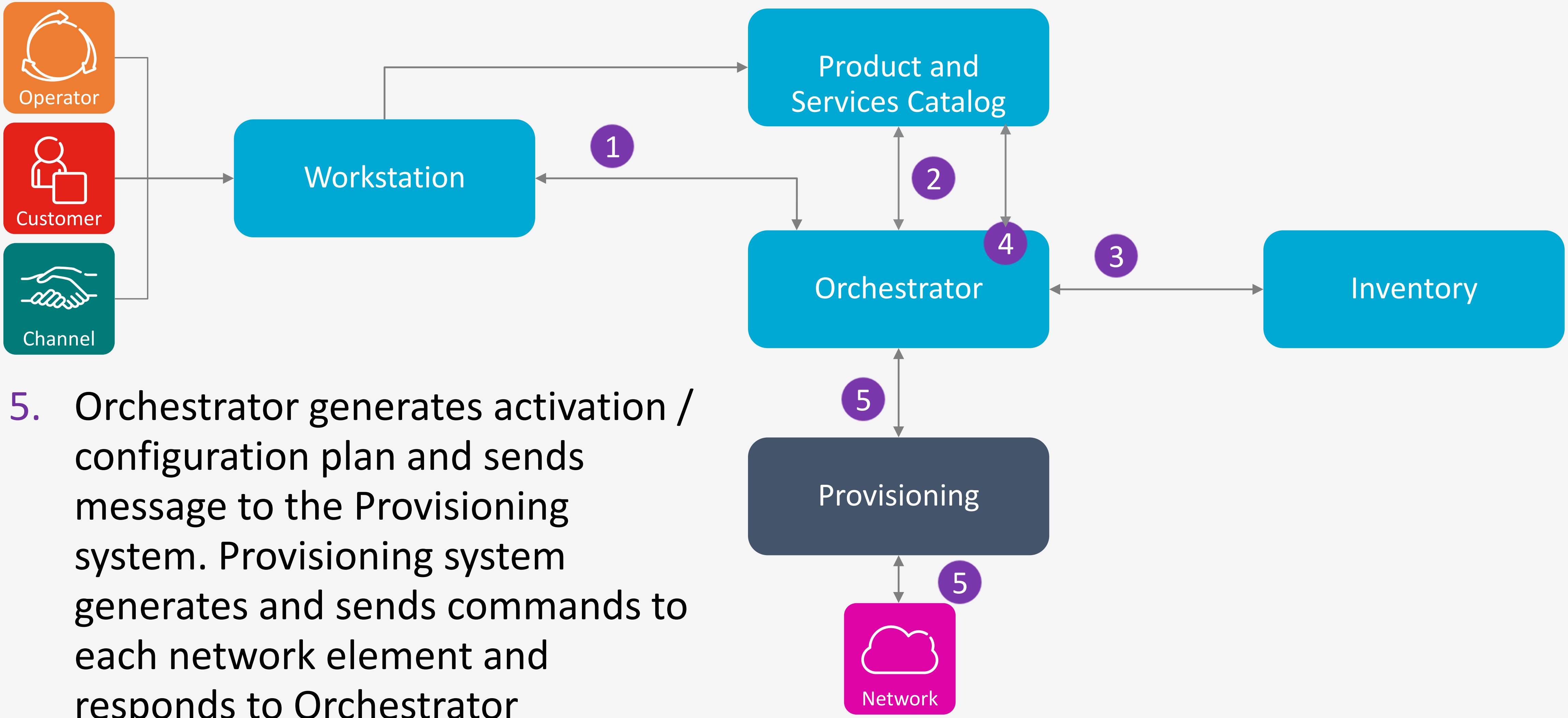
Add Flow Example for Carrier Ethernet



4. Orchestrator gathers data for each end point, creates an **Order** and *designs* the service (using Service Templates). Sometimes the Designer is a separate function or embedded with the Inventory system.

Automated Lead-to-Service

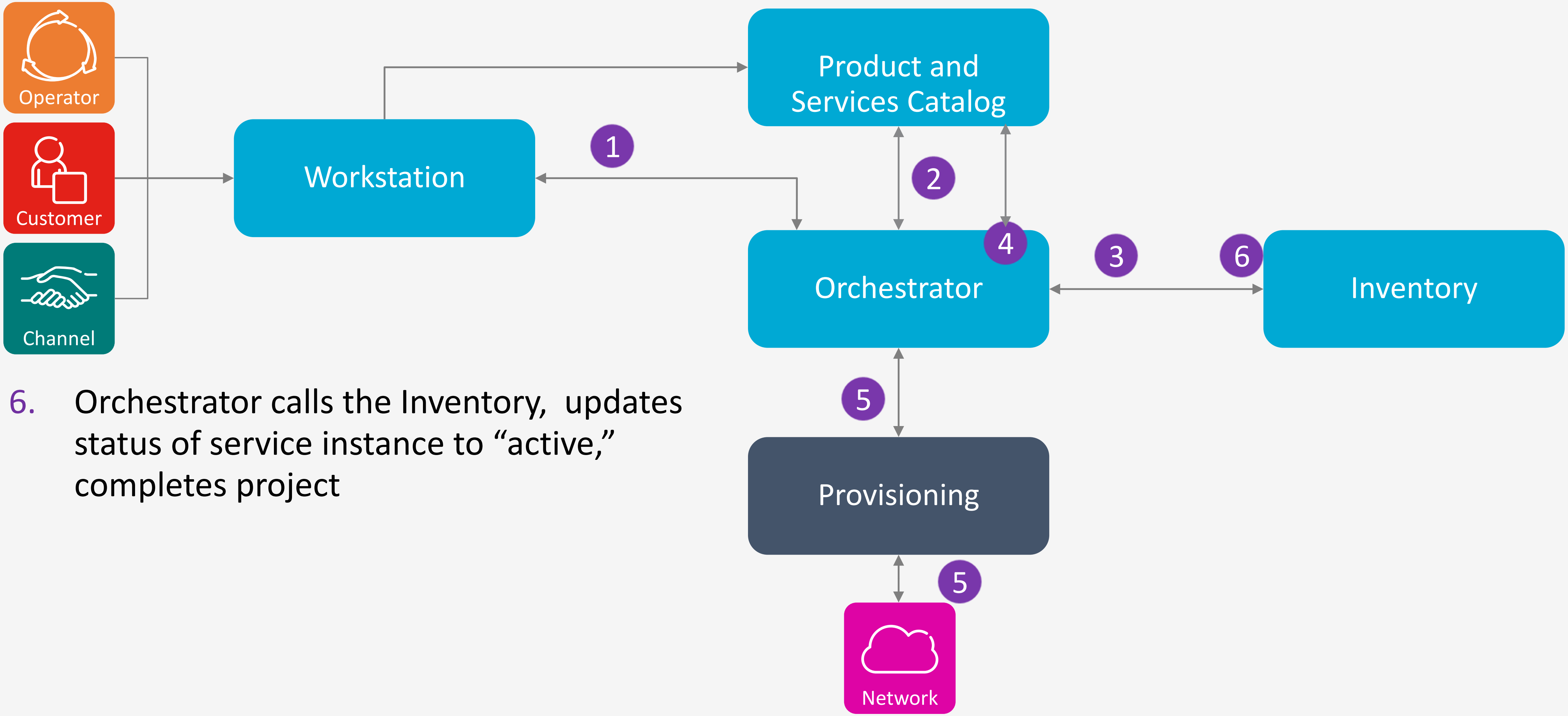
Add Flow Example for Carrier Ethernet



5. Orchestrator generates activation / configuration plan and sends message to the Provisioning system. Provisioning system generates and sends commands to each network element and responds to Orchestrator

Automated Lead-to-Service

Add Flow Example for Carrier Ethernet



6. Orchestrator calls the Inventory, updates status of service instance to "active," completes project

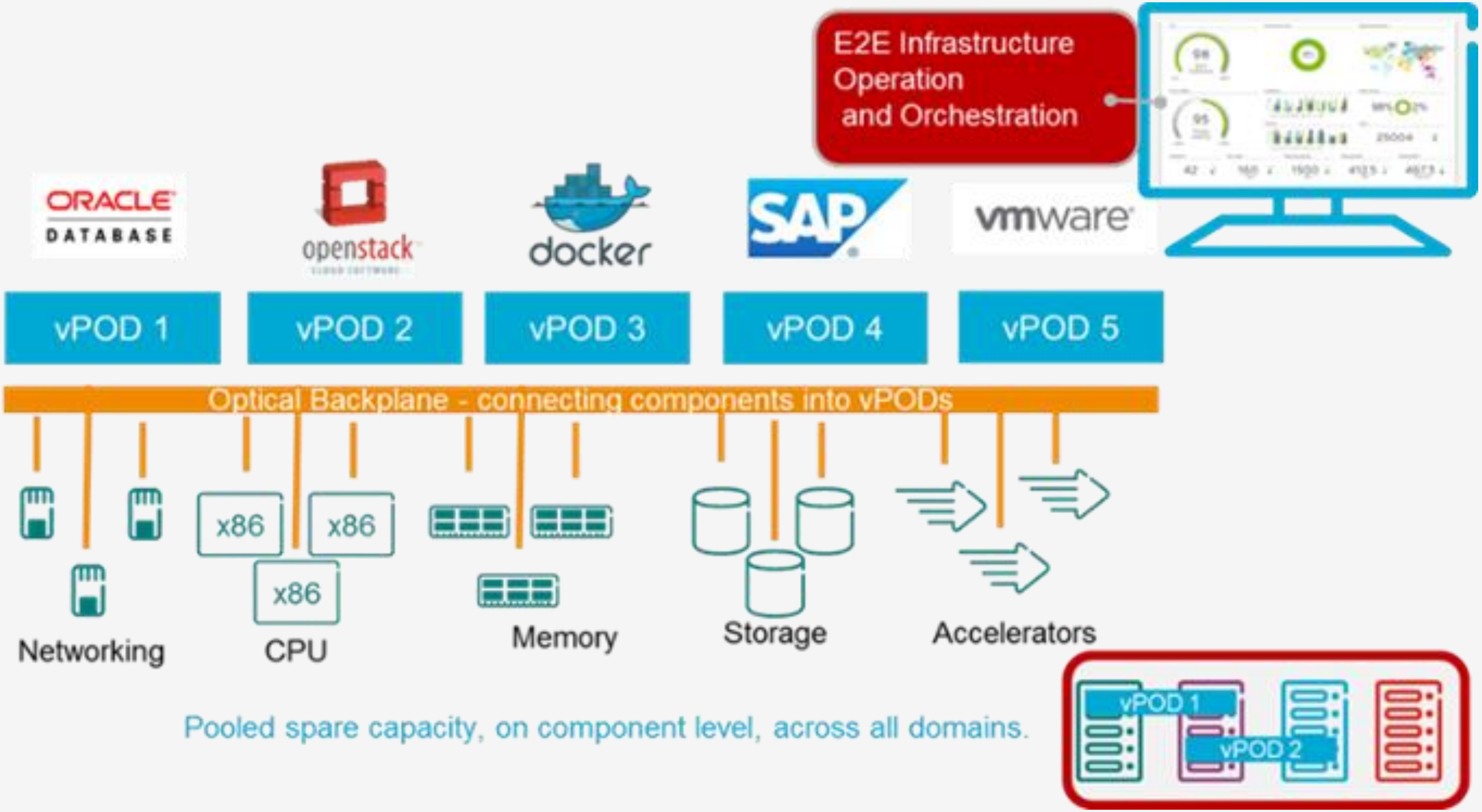
Sireen Malik, Mobile Computing Architecture, 2020

Software Defined Infrastructure

(for compute and storage)

Sireen Malik. Mobile Computing Architecture, 2020

Software Defined Infrastructure (SDI)


E2E Infrastructure Operation and Orchestration
ORACLE DATABASE
openstack
docker
SAP
vmware
vPOD 1
vPOD 2
vPOD 3
vPOD 4
vPOD 5
Optical Backplane - connecting components into vPODs
x86
x86
x86
Networking
CPU
Memory
Storage
Accelerators
Pooled spare capacity, on component level, across all domains.
vPOD 1
vPOD 2

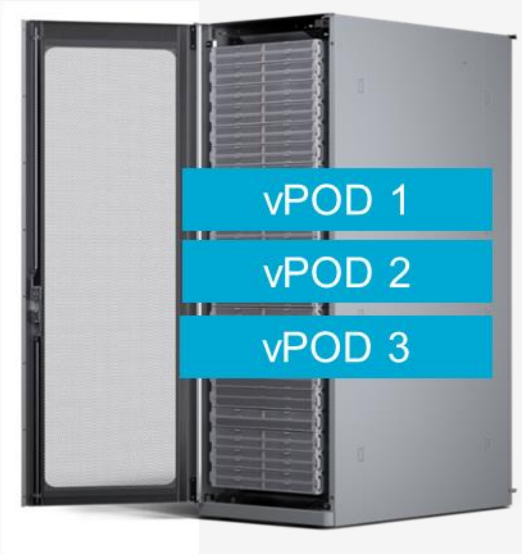
Sireen Malik, Mobile Computing Architecture, 2020

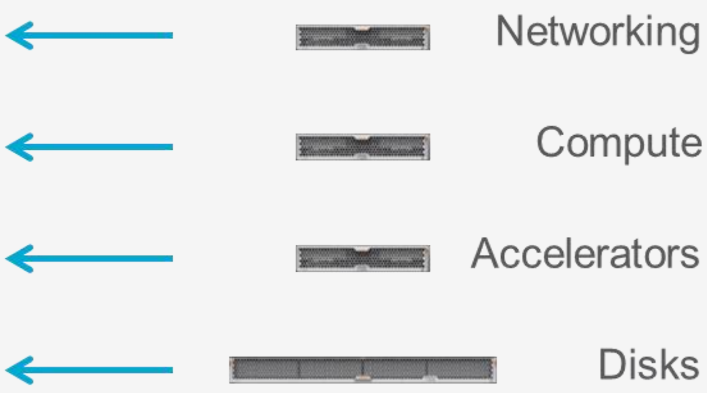
Hyperscale Architecture - Example

Visibility

Software Defined Infrastructure

Get lean: Desegregation and Pools





Open management and integration with Command Center

of the complete datacenter, Ericsson and 3PP

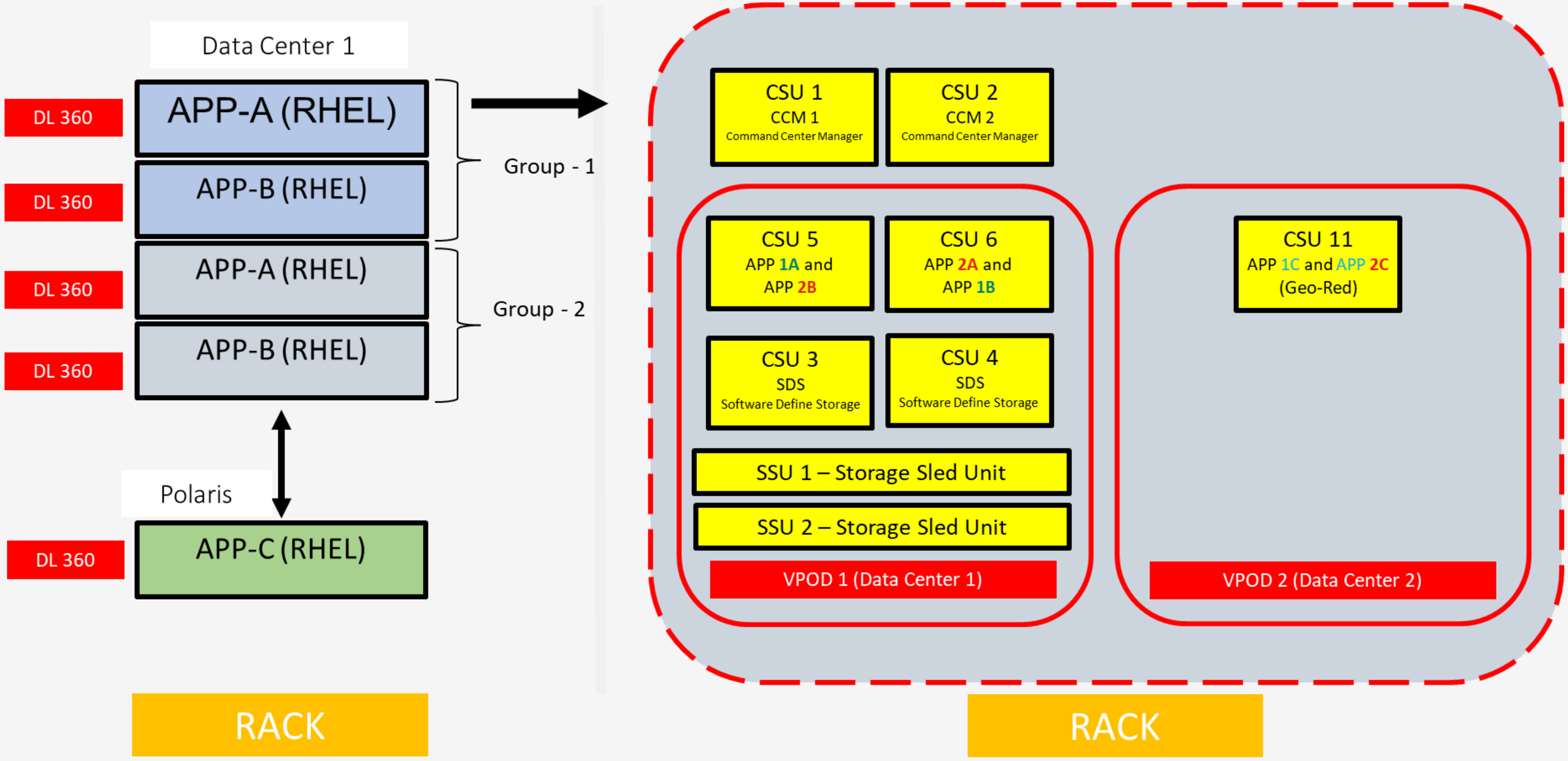
vPOD, virtual POD

SW defined infrastructure combining components into virtual performance optimized datacenters

Disaggregated hardware

Seamless scalability with efficient life cycle management

SDI Deployment Example



Data Center 1

DL 360 | APP-A (RHEL)

DL 360 | APP-B (RHEL)

Group - 1

DL 360 | APP-A (RHEL)

DL 360 | APP-B (RHEL)

Group - 2

Polaris

DL 360 | APP-C (RHEL)

RACK

CSU 1
CCM 1
Command Center Manager

CSU 2
CCM 2
Command Center Manager

CSU 5
APP 1A and
APP 2B

CSU 6
APP 2A and
APP 1B

CSU 3
SDS
Software Define Storage

CSU 4
SDS
Software Define Storage

SSU 1 – Storage Sled Unit

SSU 2 – Storage Sled Unit

VPOD 1 (Data Center 1)

CSU 11
APP 1C and APP 2C
(Geo-Red)

VPOD 2 (Data Center 2)

RACK

Reading Material

SDN and Security.pdf