The bridge to possible

Introducing XRd

Lightweight, Programmable and Containerized

Raja Kolagatla, Product Manager, Cisco Networking

Cisco Webex App

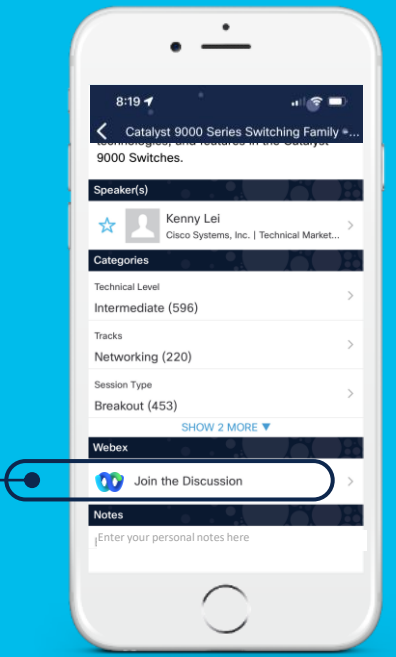
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated until February 24, 2023.



A Few Questions to Start With …

- Are Service Providers interested in network virtualization, which functions and for what?
- What has changed to make Virtual Routing more relevant and going mainstream?
  - SDN and NFV have been around for 10+ years.
- To what degree does the Virtual Routing complement the Physical Routing?
  - The value of software routing functions being moved around at will.
- Can Virtual Routing run on modern servers crank up enough throughput for SP use cases?

Agenda

- Introduction and Market Trends
- Product Overview
- XRd Use Cases and Deployment
- Automation
- Conclusion

Introduction and Market Trends

How does NetFlix Open-Connect work?

Open Connect is Netflix’s in-house content distribution network specifically built to deliver its TV shows and movies.

Market and Architectural Evolution

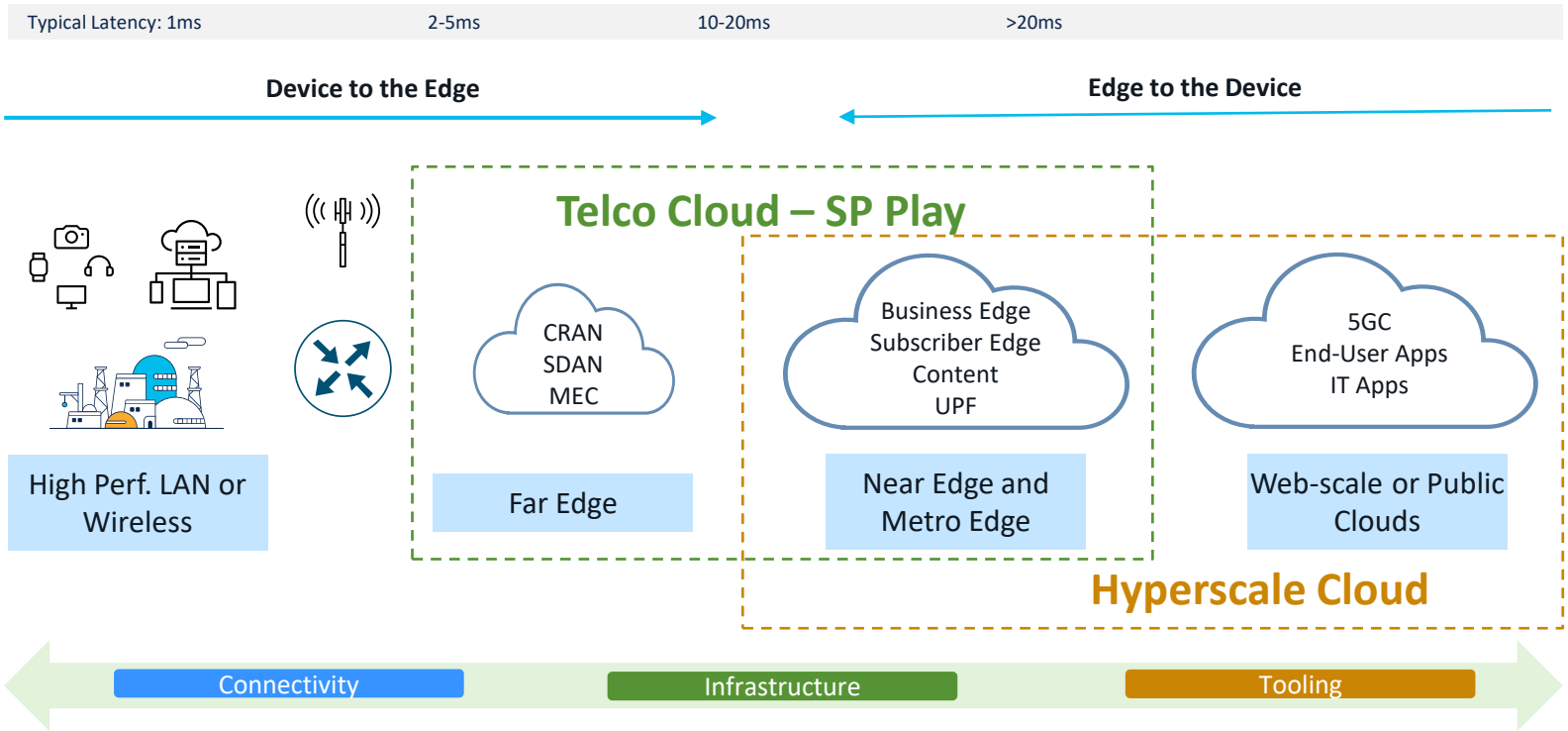
Key Inflection Points and Telco Edge

- Key Inflection Points in Networking
  - Emergence of edge computing
  - New edge applications and services
  - Localized content
  - New revenue creation
- Telco Edge is Evolving
  - On-Premise
  - Far Edge
  - Near and Metro Edge
  - Public Cloud

Calls for Dynamic Workload Placement
Floating Edge

Market and Architectural Evolution

Dynamic work-load placement: low latency, high bandwidth, massive scale



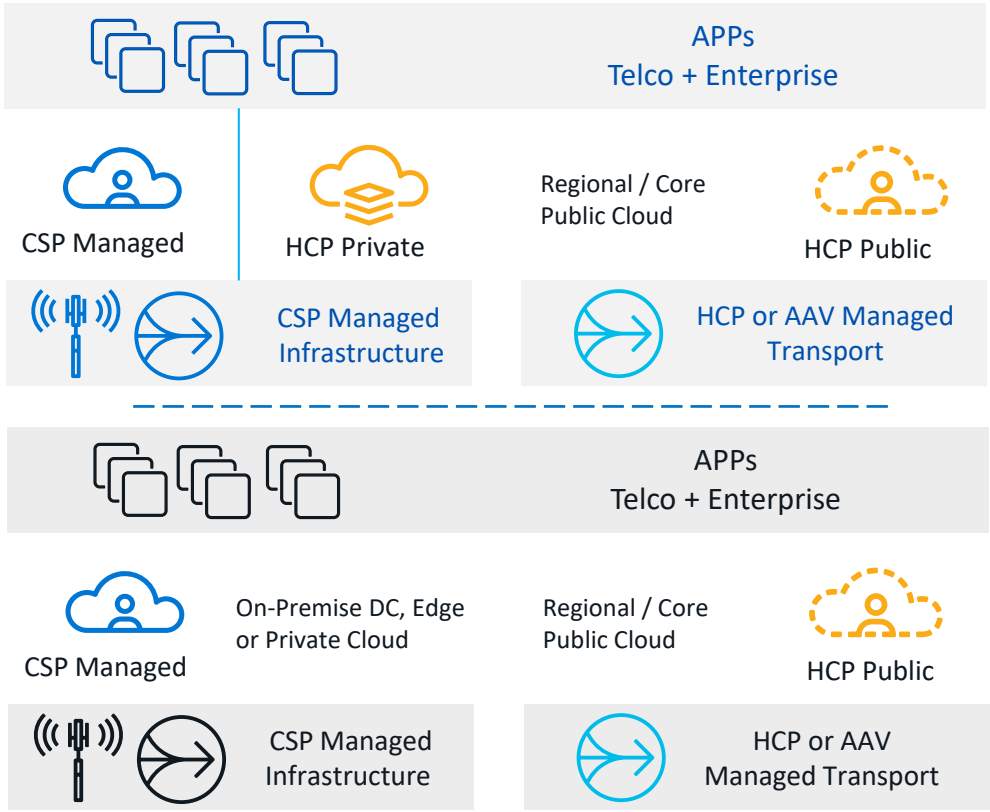
Hybrid-Cloud Adoption

Combination of on-premise and off-premise infrastructure and applications

CSP Telco hybrid cloud environment: Any combination of CSP managed and HCP (Hyperscale cloud provider) managed infrastructure.

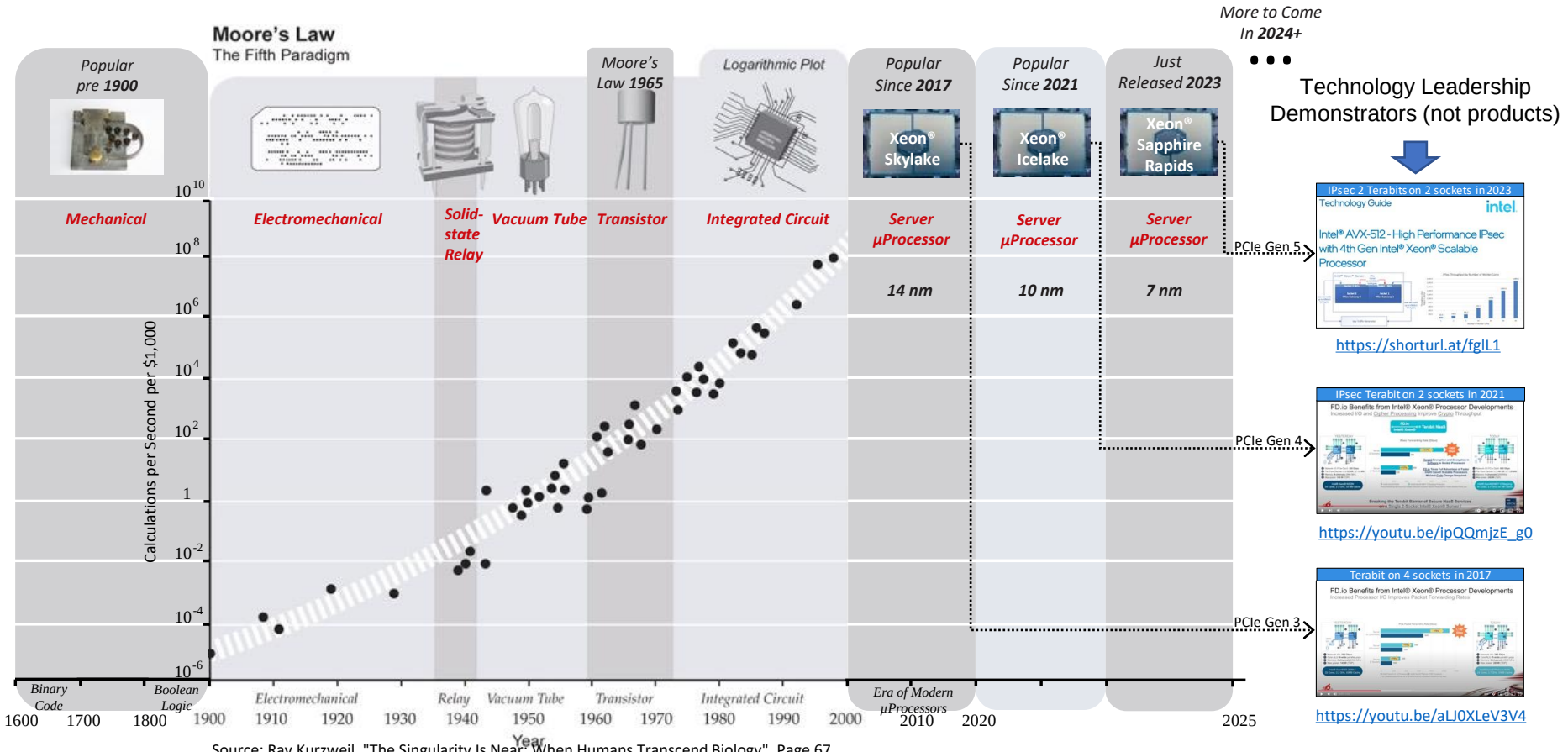
Trend Drivers:

- TCO Optimization
- Open and Multi-vendor environments
- Enterprise Focused Use cases
  *decides edge, private or public cloud deployment



Azure's partnership with operators: "We meet operators where they are—on-premises, at the edge, and in the cloud"
https://azure.microsoft.com/en-us/resources/future-of-mobile-networks-and-cloud-computing/

Remember **1965** ”Moore’s Law” – Well, It Surely Does Ramble on …



https://shorturl.at/fglL1

https://youtu.be/ipQQmjzE_g0

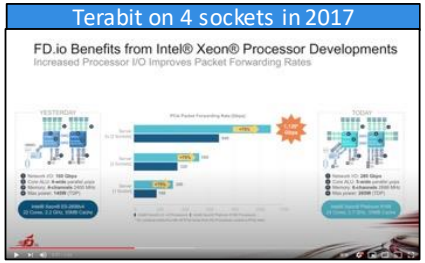
https://youtu.be/aLJ0XLeV3V4

Source: Ray Kurzweil, "The Singularity Is Near: When Humans Transcend Biology", Page 67, The Duckworth Publishers 2009. Data points between 1600 and 1900, and after 2000 represent presenter’s perspective and benchmarked data.

New Moore's law for NFV: Performance doubles every 3 years!

Intel® Xeon® Skylake
PCIe Gen 3.0 x16 lanes
for 100 GbE*

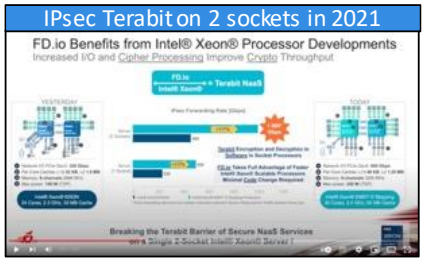
Terabit on 4 sockets in 2017



https://youtu.be/aLJ0XLeV3V4

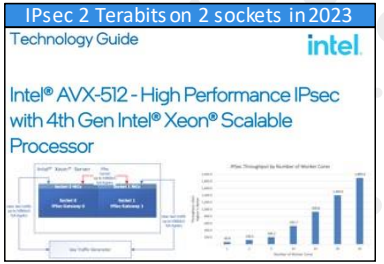
Intel® Xeon® Icelake
PCIe Gen 4.0 x16 lanes
for 200 GbE*

IPsec Terabit on 2 sockets in 2021



https://youtu.be/ipQQmjzE_g0

IPsec 2 Terabits on 2 sockets in 2023



https://shorturl.at/fglL1

Intel® Xeon® Sapphire Rapids
PCIe Gen 5.0 x16 lanes
for 400 GbE*

Factors contributing to the increased throughput on servers:

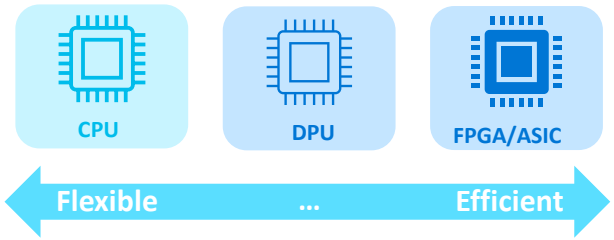
- Increasing packet processing power (and less Watts) with better ISA* and lower nm***
- More transistors, more network optimized PCIe buffering
- PCIe speeds doubling every 3-4 years

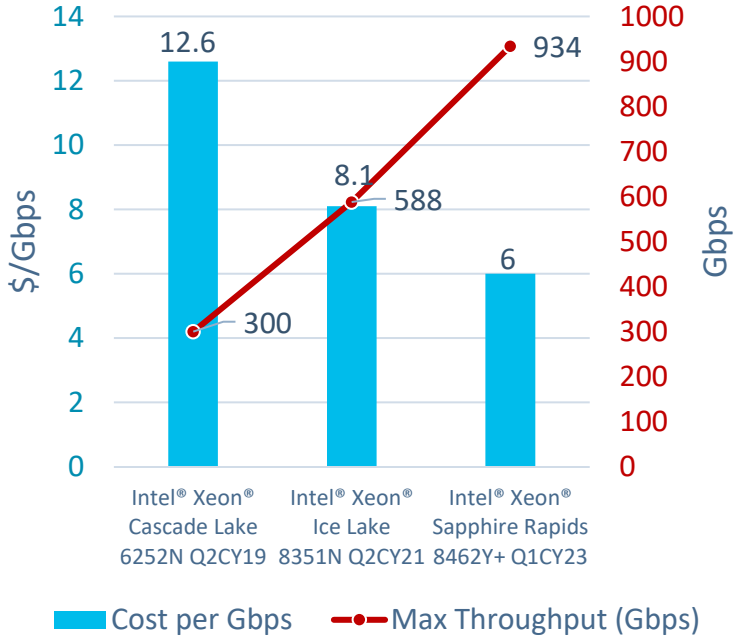
*Terabit demo: 48x lanes per socket
** Instruction Set Architecture
*** nano meter fab technologies

Advancements in Server Throughput …

…are accompanied by cost-performance improvements

- x86 CPU evolution does not follow the Moore's Law anymore, but network performance does ☺
- And there is a growing number of hardware accelerators to further boost performance



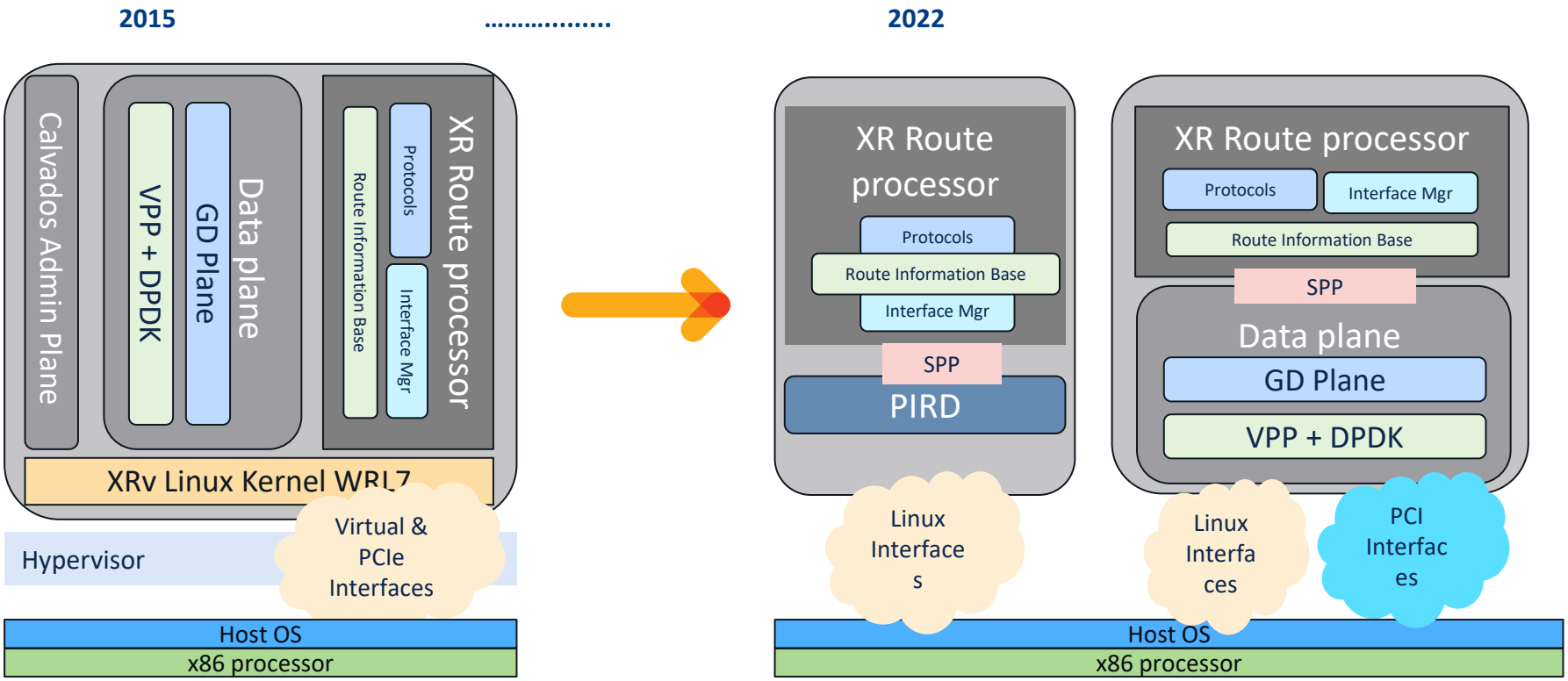


Server Data Plane Cost-Performance Guesstimates

Based on VPP performance on Intel® Xeon® Processor (single socket)

Product Overview

XRd architecture evolution from XRv9K- working on VPP stack since a decade!



XRv9K: LXCs within VM

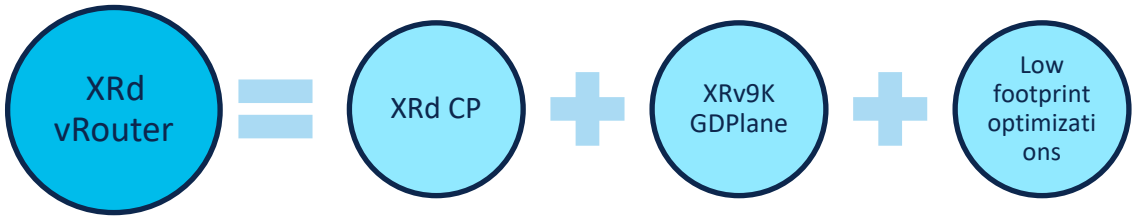
XRd: Control Plane & vRouter Containers

Glossary

- Control plane: Cisco developed packages for core network functions (BGP, MPLS, etc.); Yocto packages for standard Linux tools and libraries(bash, python, tcpdump, etc.).
- PIRD: Platform Independent Reference Data plane - provide a functional platform layer to enable the XR software packet path, designed to be used on virtual platforms.
- GD Plane: General Data Plane
- VPP: Vector Packet Processing
- DPDK: Data Plane Development Kit that consists of libraries to accelerate packet processing workloads running on a wide variety of CPU architectures.

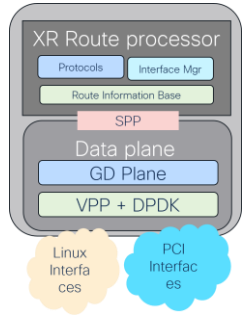
XRd Architecture

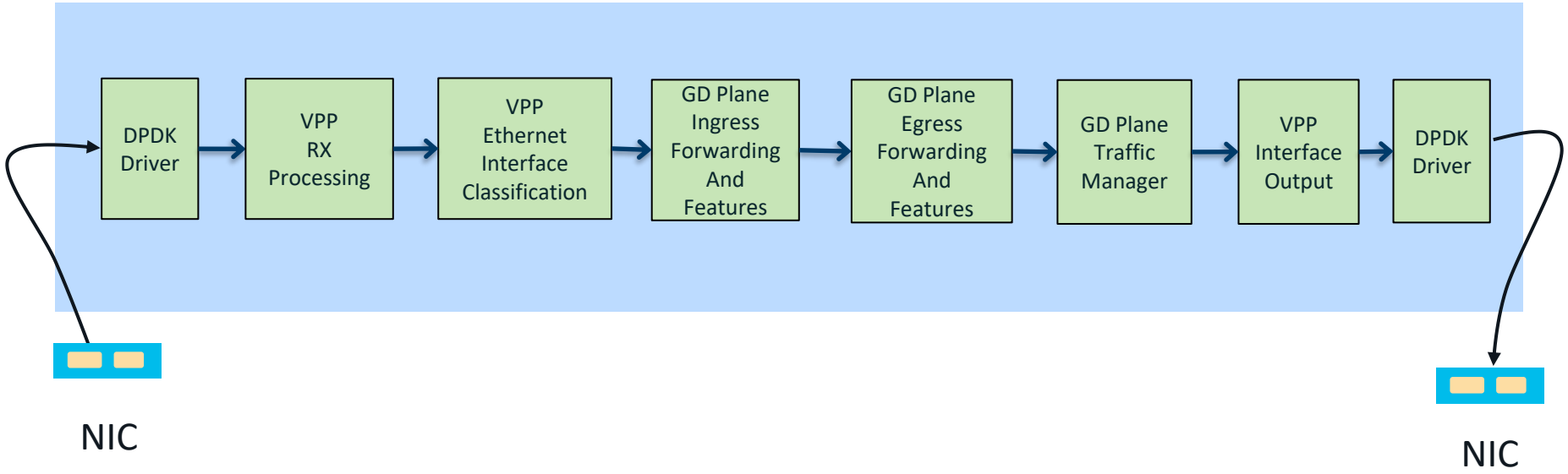
- Two variants: Control Plane only and complete vRouter (Control Plane + Data Plane)
- Lyndt a.k.a. XR7 based architecture (No Admin Plane)
- Optimizations for low footprint core (vCPU) use cases
- Throughput scales up with additional resources (vCPUs)
- Boot time: About 90 seconds
- Resource requirements: Typical 8G RAM, 2 CPUs (Min) and 7GB disk



Life of a Packet for XRd

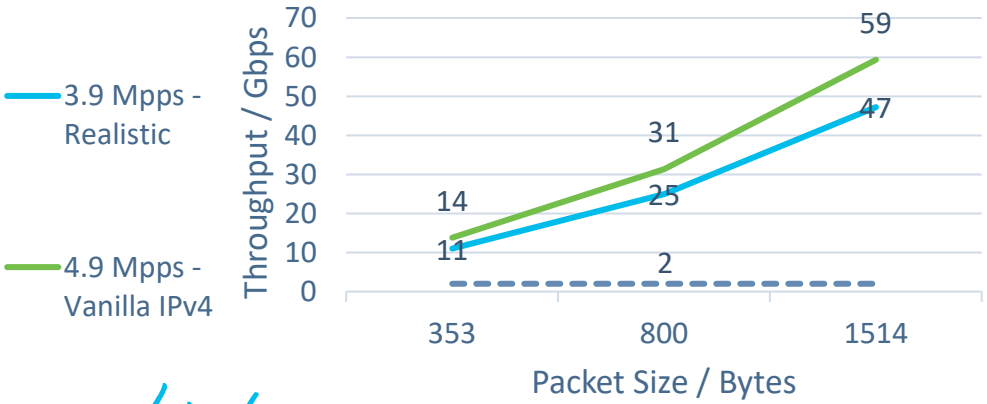
vRouter container





Performance: 10’s Gbps/core, Terabit-level/2-socket system

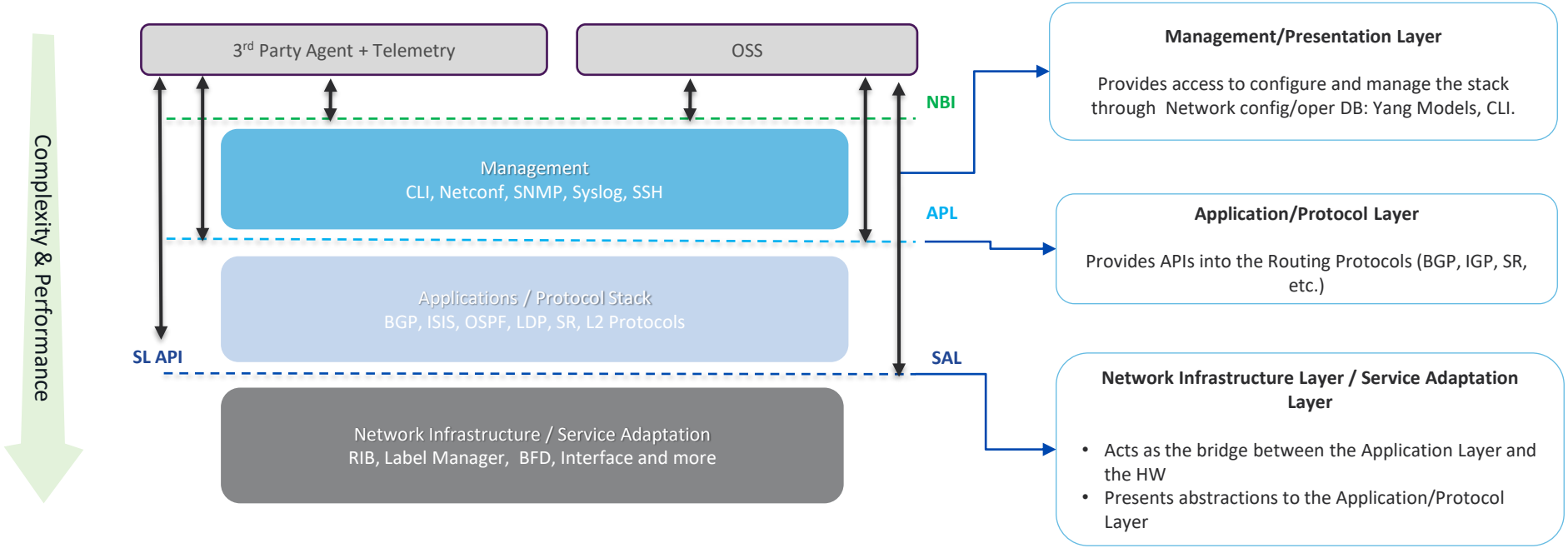
- ‘Realistic’ configuration uses L3VPN over SR-MPLS over ECMP VLANs with Egress HQoS
- Latency: ~50 usec average through XRd
- Ice Lake CPU @ 3.5 GHz turbo



With multi-core close to linear speedup, terabit level performance is becoming possible.

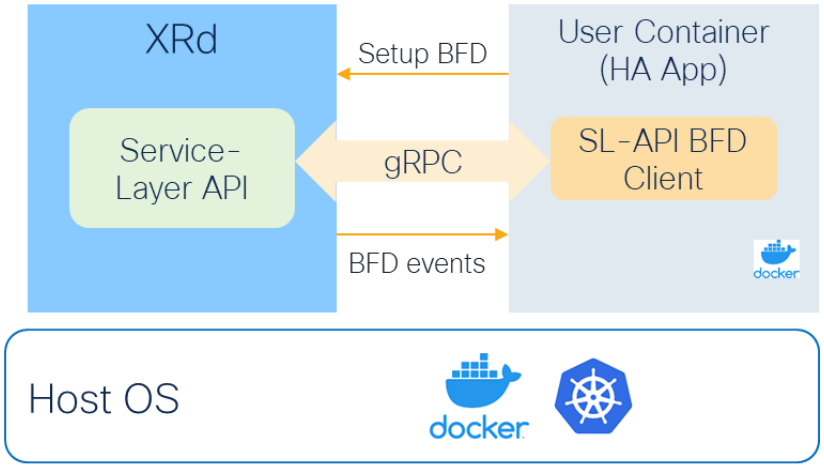
Model Driven Manageability

APIs at every layer for the stack



Solution for HA on Public Cloud with Service Layer API

Use-case for AWS

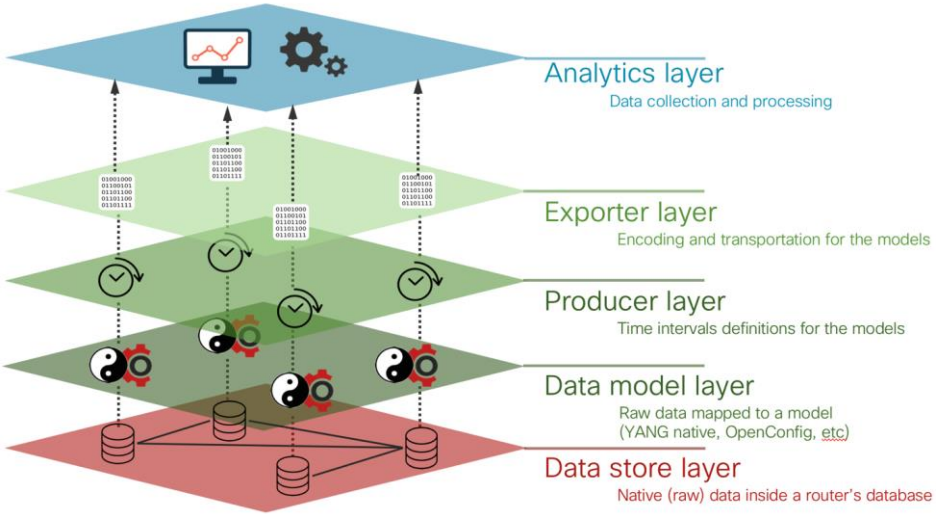


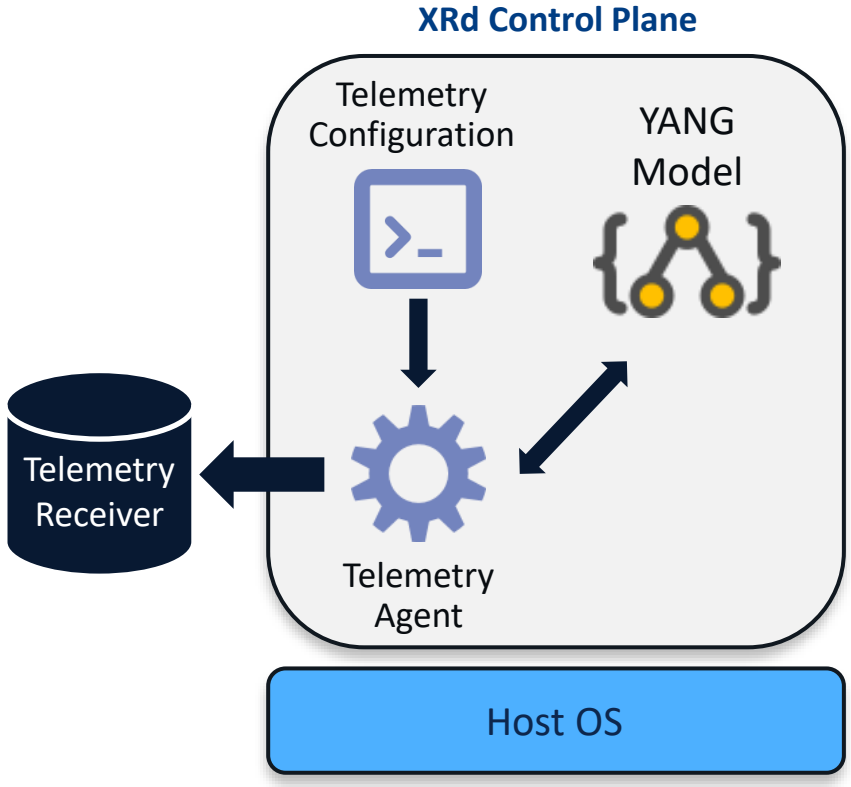
HA App interaction with IOS-XR

- AWS underlay doesn't support broadcast packets, to run HSRP/VRRP to be run between active & standby XRd's.
- Service-Layer API– "Service-Layer" acts as client for BFD allowing us to add/modify BFD sessions directly WITHOUT any IOS-XR config. BFD session state is transient/ephemeral – tied to the client.
- Real-time BFD events are then received over gRPC, allowing the SLAPI client to react to Peer-router unreachable/down events

Streaming telemetry

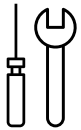
Supported by virtue of XR7





XRd Usecases & Deployment

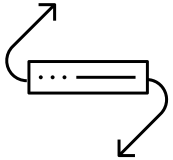
Cisco XRd Use Cases



Light weight lab simulation

CI/CD deployment

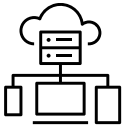
Thorough XR Coverage
Fast boot time: ~2 Min



VIRTUAL ROUTE REFLECTOR / PCE

Industry Leading Scale

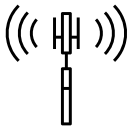
Up to 70M Paths
20M Routes
100 RR Groups



VIRTUAL PROVIDER EDGE

Consistent architecture w/ ASR 9000

T'put: ~100Gbps Business VPNs, M'cast, Peering



VIRTUAL CELL SITE ROUTER

Lowest XR footprint

T'put: ~30 Gbps[#]
Routes: 20K VPNv4/V6
SRv6/SR-MPLS/SR-TE
HQoS

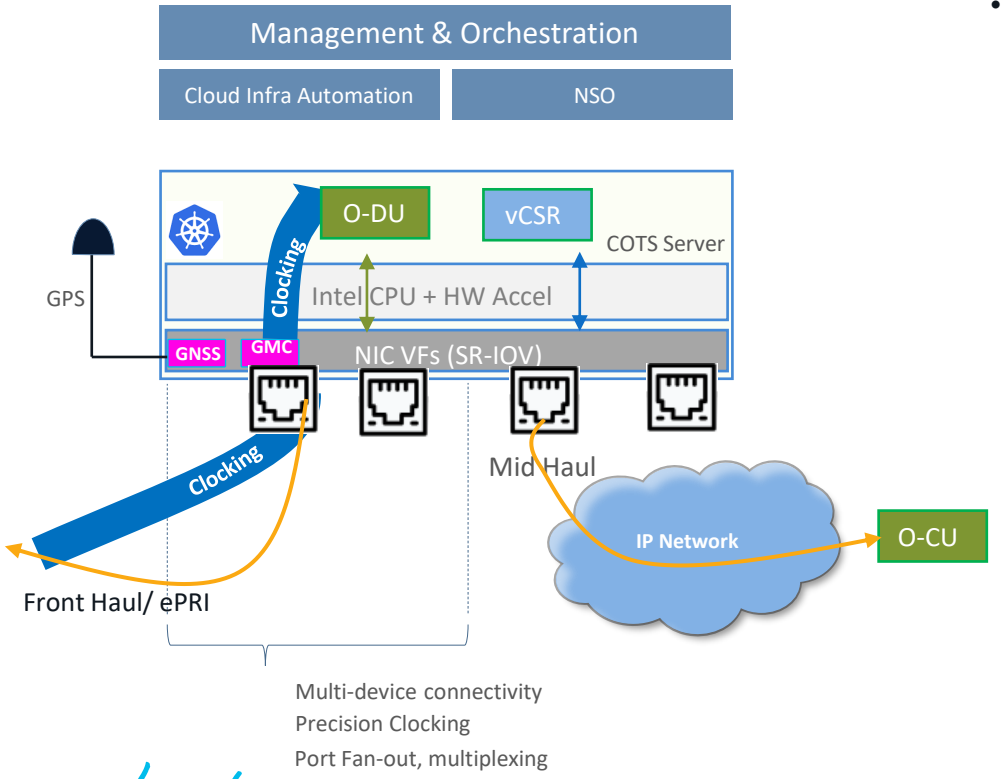
CLOUD ROUTER

Public Cloud Gateway

Routes: Up to 100K[#]
GRE Tunnels: ~400[#]
BGP Sessions: 1000[#]
SRv6[$], SR-MPLS, GRE Overlay

[#]Linearly scalable with the vCPU allocation
[$]Dependent on underlay

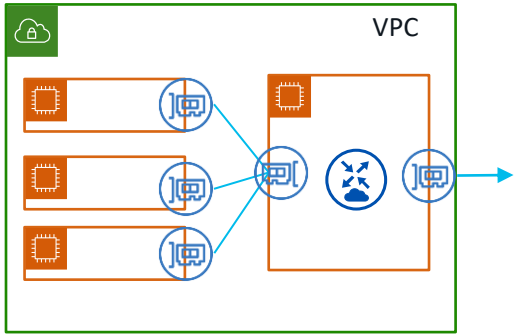
virtual Cell Site Router (vCSR) Deployment with XRd



- Key features for vCSR role
  - Transport: SR-MPLS, SRv6
  - SRv6-TE, SR-TE
  - ISIS, BGP L3VPN (v4/v6)
  - TI-LFA
  - ACL
  - VRRP
  - HQoS
  - GRE, ECMP – Load-balancing

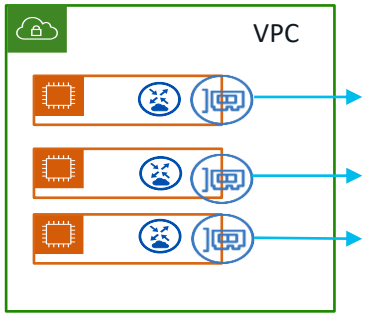
Cloud Networking for Public Cloud

Traditional vRouter



Inter-subnet communication within VPC or inter-VPC communication happens via vRouter

XRd vRouter



Distributed Routing simplify the routing architecture

- Key features for Cloud Router role
  - BFD, GRE Tunnelling
  - BGP VPNs (v4/v6) & ISIS
  - QoS for GRE Interfaces
  - ACLs
  - Load balancing – GTP, ECMP
  - PBR, ABF
  - SR-MPLS, SRv6 (if supported by underlay)


EC2 Instance

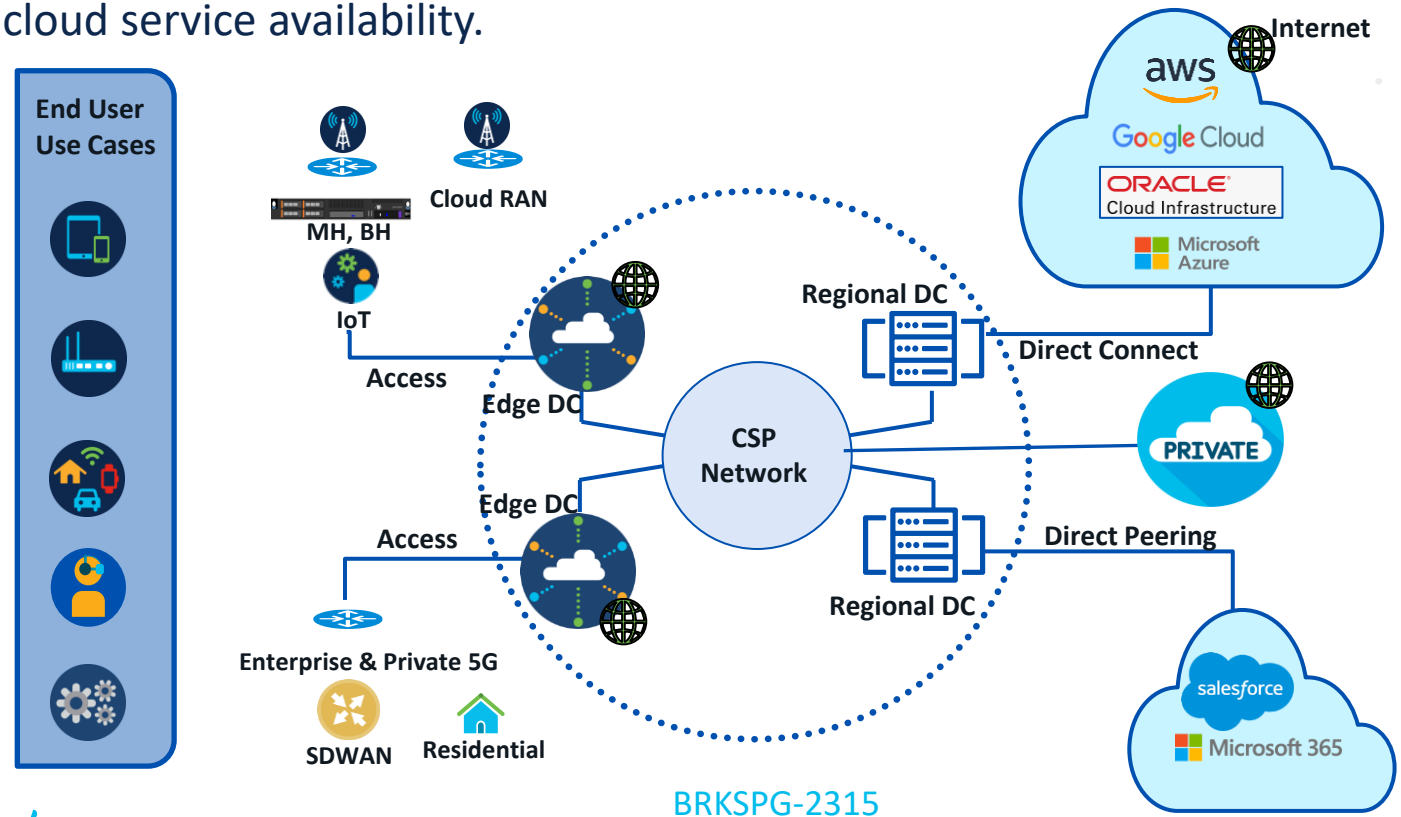

XRd vRouter


ENI interface


VPC

Cloud-ready Converged SDN Transport

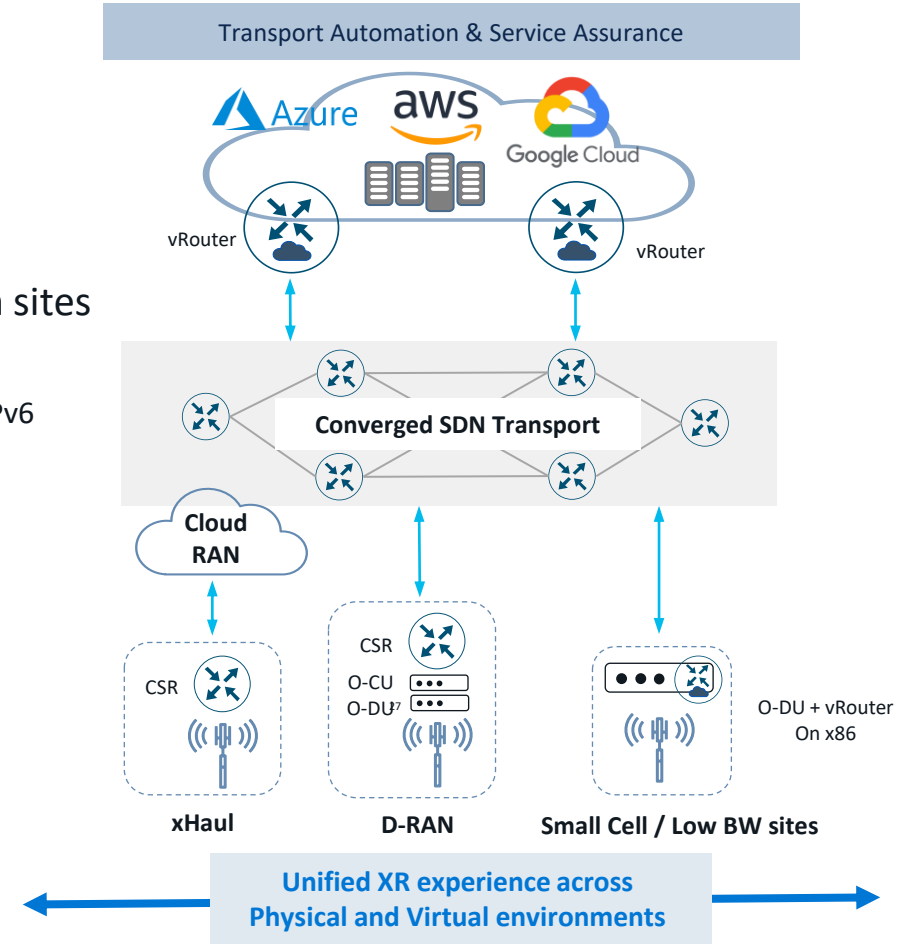
Cisco's Cloud-ready Converged SDN Transport is extended to support public cloud infra and cloud service availability.



BRKSPG-2315

XRd vRouter Role in Cloud-ready Converged SDN Transport

- L3VPN, Anycast & BGP extensions to VPCs
- vCSR integrated with vDU server for low bandwidth sites
- End to End SR with Slicing
  - GRE tunnels are option in case the public cloud doesn't support IPv6
- Automation and Service Assurance
- vRouter functions:
  - XRd as CNF*
  - XRv9k as VNF** (if needed)



* Containerized Network Function
** Virtualized Network Function

Telco Workload & Network Requirements

- Telco workloads have strict requirements
  - Latency budgets, jitter
  - Seamless failover to ensure service continuity, and regulatory requirements are mandated by federal and/or state agencies.
  - Data persistency & Storage management
- Multiple limits in cloud provider networks are imposed on customers based on traffic flows and virtual machine templates that aren’t fully aligned with Telco specific requirements.
- Public cloud providers does not provide the level of visibility and control in their underlay network that Telco requires

XRd Deployment workflows

XRd Host Requirements

- Any linux distribution, with a recent (4.x) kernel and built from ‘official packages’
  - cgroups version v1
  - Drivers: vfio-pci or igb_uio
  - Host privileges
- Container tooling: Docker (v18+) and Kubernetes (v1.22)
  - Supports any OCI-compliant tools withing K8S ecosystem
  - Some tweaks to host OS capabilities
- Beyond vanilla Kubernetes:
  - AWS EKS (supported from 7.8.1onwards)

Host Check Script

Walkthrough for setting up Host Environment:

https://xrdocs.io/virtual-routing/tutorials/2022-08-22-setting-up-host-environment-to-run-xrd/

Publicly Available Host Check Script

https://github.com/ios-xr/xrd-tools/blob/main/scripts/host-check

```
cisco@xrdcisco:~/xrd-tools/scripts$ ./host-check -p xrd-vrouter -e docker -e xr-compose
============================
Platform checks - xrd-control-plane
============================
PASS -- CPU architecture (x86_64)
PASS -- CPU cores (8)
PASS -- Kernel version (5.4)
PASS -- Base kernel modules
        Installed module(s): dummy, nf_tables
PASS -- Cgroups version (v1)
PASS -- systemd mounts
        /sys/fs/cgroup and /sys/fs/cgroup/systemd mounted correctly.
FAIL -- Inotify max user instances
        The kernel parameter fs.inotify.max_user_instances is set to 128 but
        should be at least 4000 (sufficient for a single instance) - the
        recommended value is 64000.
        This can be addressed by adding 'fs.inotify.max_user_instances=64000'
        to /etc/sysctl.conf or in a dedicated conf file under /etc/sysctl.d/.
        For a temporary fix, run:
          sysctl -w fs.inotify.max_user_instances=64000
WARN -- Inotify max user watches
        The kernel parameter fs.inotify.max_user_watches is set to 8192 -
        this is expected to be sufficient for 2 XRd instance(s).
        The recommended value is 64000.
        This can be addressed by adding 'fs.inotify.max_user_watches=64000'
        to /etc/sysctl.conf or in a dedicated conf file under /etc/sysctl.d/.
        For a temporary fix, run:
          sysctl -w fs.inotify.max_user_watches=64000
INFO -- Core pattern (core files managed by the host)
PASS -- ASLR (full randomization)
INFO -- Linux Security Modules
        AppArmor is enabled. XRd is currently unable to run with the
        default docker profile, but can be run with
        '--security-opt apparmor=unconfined' or equivalent.
PASS -- RAM
        Available RAM is 28.7 GiB.
        This is estimated to be sufficient for 14 XRd instance(s), although memory
        usage depends on the running configuration.
        Note that any swap that may be available is not included.

============================================================
!! Host NOT set up correctly for xrd-control-plane !!
============================================================
```

Helm chart snapshot for XRd

myvalues.yaml

```
xrd1:
  # Image configuration
  image:
    # Repository for the container image (required).
    repository: "my-repo/xrd-vrouter"
    # Image tag (required).
    tag: "7.8.1"

  # Pod resource configuration.
  resources:
    limits:
      memory: 10Gi
      hugepages-1Gi: 6Gi

  # Security context for the XRd container.
  # Privileged mode is currently required for XRd in K8s due to device
  # access requirements.
  securityContext:
    privileged: true

  # Node labels for pod assignment.
  nodeSelector:
      xrd.node: nodeA

  # Persistent storage controls.
  persistence:
    enabled: true
    size: "6Gi"
    accessModes:
    - ReadWriteOnce
    storageClass: "gp2"
```

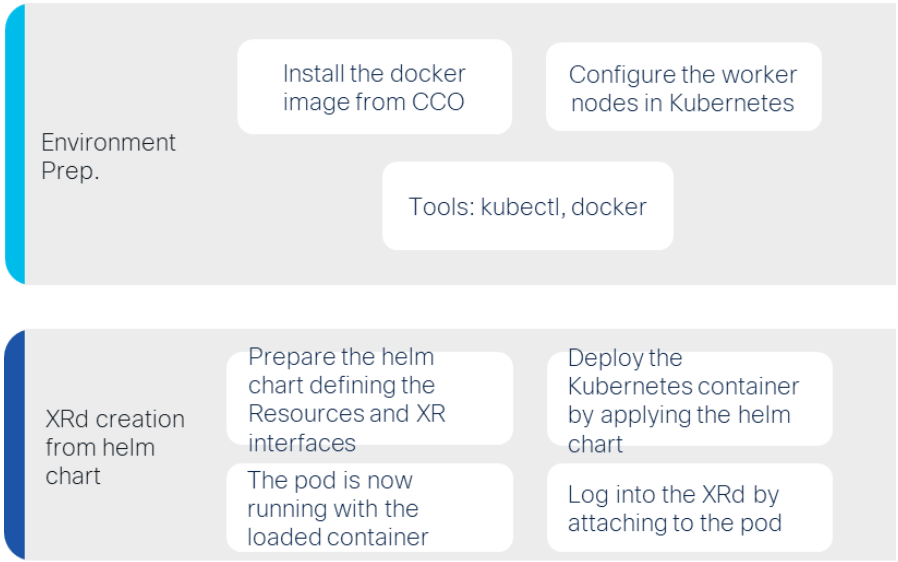
```
# XR configuration and boots scripts.
config:
  ascii: |
    hostname xrd1
    logging console debugging
    logging monitor debugging
    vrf nfs
     address-family ipv4 unicast
      import route-target
       100:100
      !
      export route-target
       100:100
      !
     !
    !
    interface HundredGigE0/0/0/0
     vrf nfs
     ipv4 address 10.0.4.12 255.255.255.0
    !
    interface HundredGigE0/0/0/1
     ipv4 address 10.0.2.12 255.255.255.0
    !
    interface HundredGigE0/0/0/2
     ipv4 address 10.0.3.12 255.255.255.0
     !

# XRd line interfaces
interfaces:
  - type: pci
    config:
      last: 3
pciDriver: "igb_uio"
cpu:
  cpuset: 2-3
```

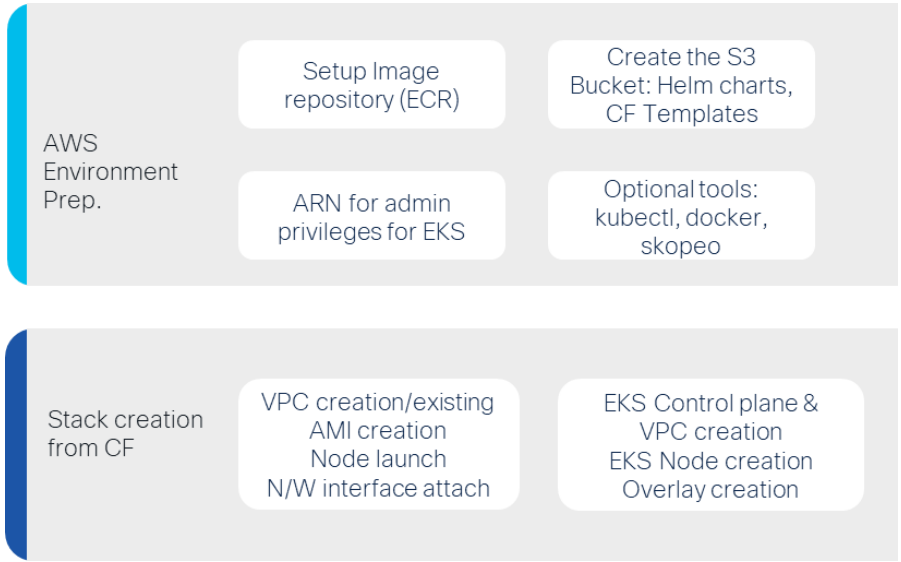
Specify PCI Interfaces

Use a subset of cpus

- Helm: a package manager for Kubernetes
- Specify details of your deployment in a YAML file
- Pass the YAML file to helm for deployment:
  - helm install my-xrd xrd/xrd-vrouter -f myvalues.yaml
- https://github.com/ios-xr/xrd-helm

XRd deployment workflows





On Vanilla K8S

On AWS EKS

Detailed steps can be found at : https://xrdocs.io/virtual-routing/tutorials

Automation

XRd Automation & Operationalization

Day “-1”

Integration to IaaC and IaaS layers

- Instance sizing
- Helm charts
- Templating via Infra Build tools (Terraform, Cloud Formation etc)

Day “0”

Deployment and Activation

- Infra Deployment by the Infra automation layer
- CNF Deployment via Infra build tools
- ZTP and Golden Config

Day “1”

Fulfillment & Service provisioning

- Provisioning
- Inventory Management
- Config Change Management
- Run-rate Operations

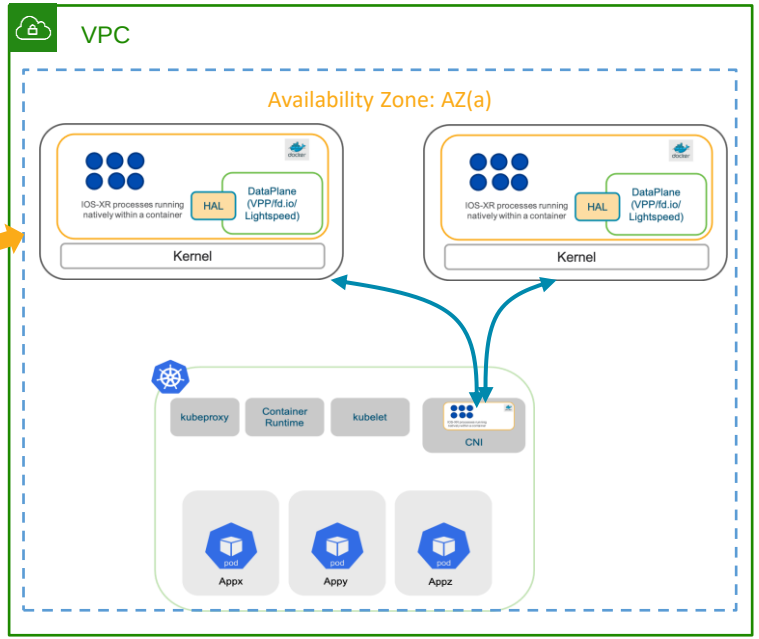
Day “N”

Service Assurance and CLA

- Fault Management
- Performance Management
- Event Management & Closed Loop Actioning

XRd Automation & Operationalization with Terraform – AWS example

Instance Provisioning driven by Terraform/CloudFormation



CNC

- Manage XRd/XRv9k using CLI/YANG over SSH/Netconf/gRPC
- Monitor using XR Telemetry
- Provide a Unified dashboard for CloudOps (Terraform Plugin events, cloud-specific instance details and stats)

Wrap up!

XRd Value Proposition

Common XR, Multiple form-factors - Runs Across
VM/Containers; Multiple CP/DP Deployment Use Cases

Versatile

Cloud Focus

Docker, Kubernetes, Multus CNI
Orchestration

Lightweight and Fast

Optional RPMs,
Low foot-print, speedy core and multi-core DP

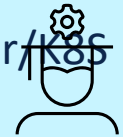
Industry Leading Control Plane

15+ years of advanced networking CP
250K+ Devices Deployed Across XR

Continue your education!!

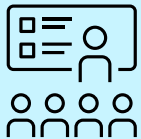
Meet the engineer in 1:1 meetings

Try out the XRd in the Docker/K8S environment
or AWS



https://xrdocs.io/virtual-routing/
https://github.com/ios-xr/xrd-tools

Related sessions



Cloud-Ready Converged SDN Transport - BRKSPG-2315

Introduction to SRv6 uSID Technology - BRKSPG-2203

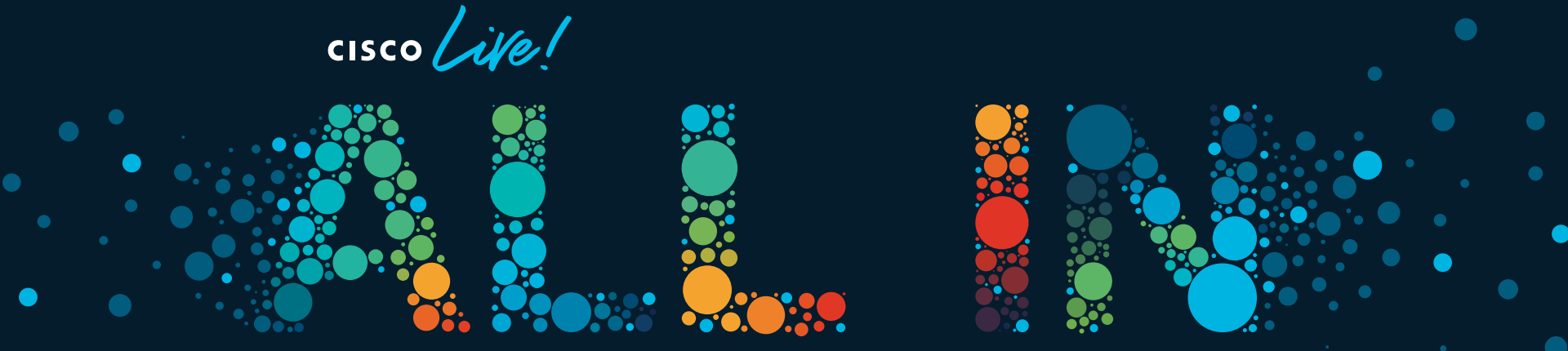
SRv6 and Cloud-Native: a Platform for Network Service Innovation - LTRSPG-2212

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (open from Thursday) to receive your Cisco Live t-shirt.

- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at https://www.ciscolive.com/emea/learn/sessions/session-catalog.html


cisco
The bridge to possible
Thank you
cisco Live!


CISCO Live!
ALL IN