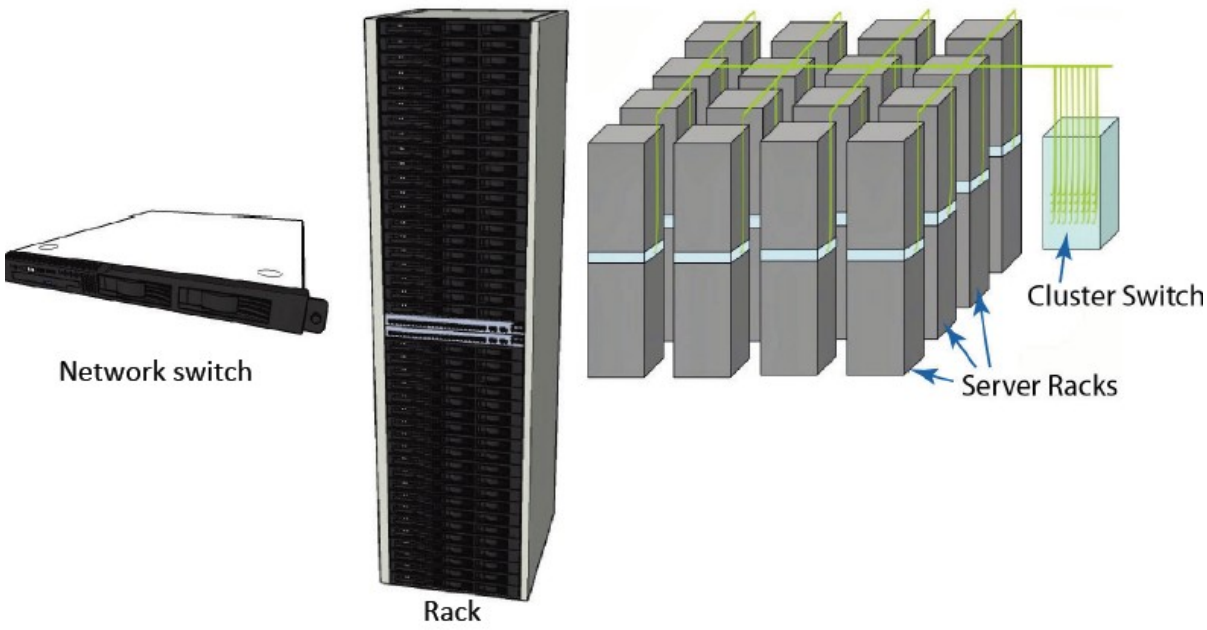
Sl.No	Program Name	
	Data center services (Engg + IT)	
1	Data center Engineering services	**Common framework**
2	Data Center IT Services	
3	Data center Security services	
4	Data Center QA services	
5	Datacenter Cloud services	
6	Datacenter compliances services	
7	Data center based business solutions	
8	Data center Networking services	

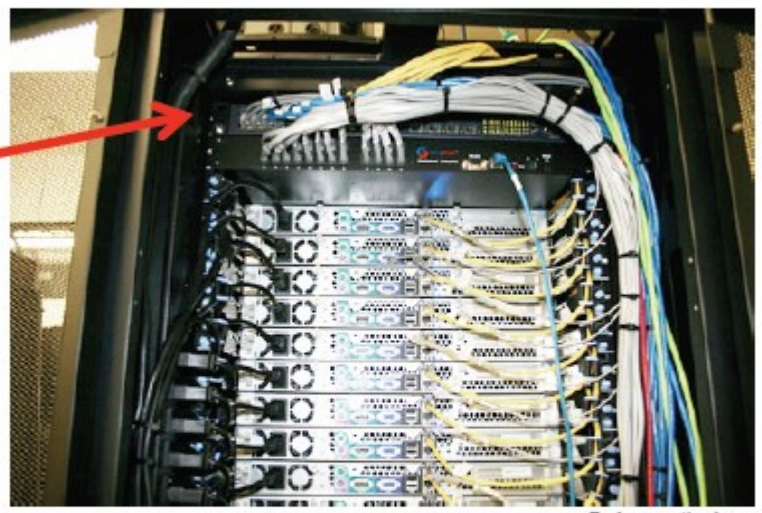
Data center services-> Common framework -> Products catalogue

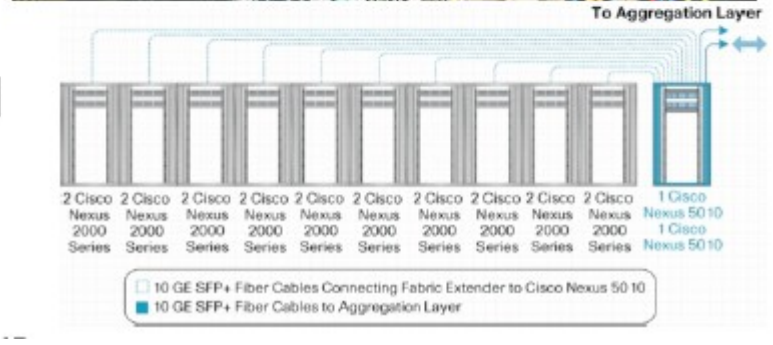
Building blocks of modern data centers



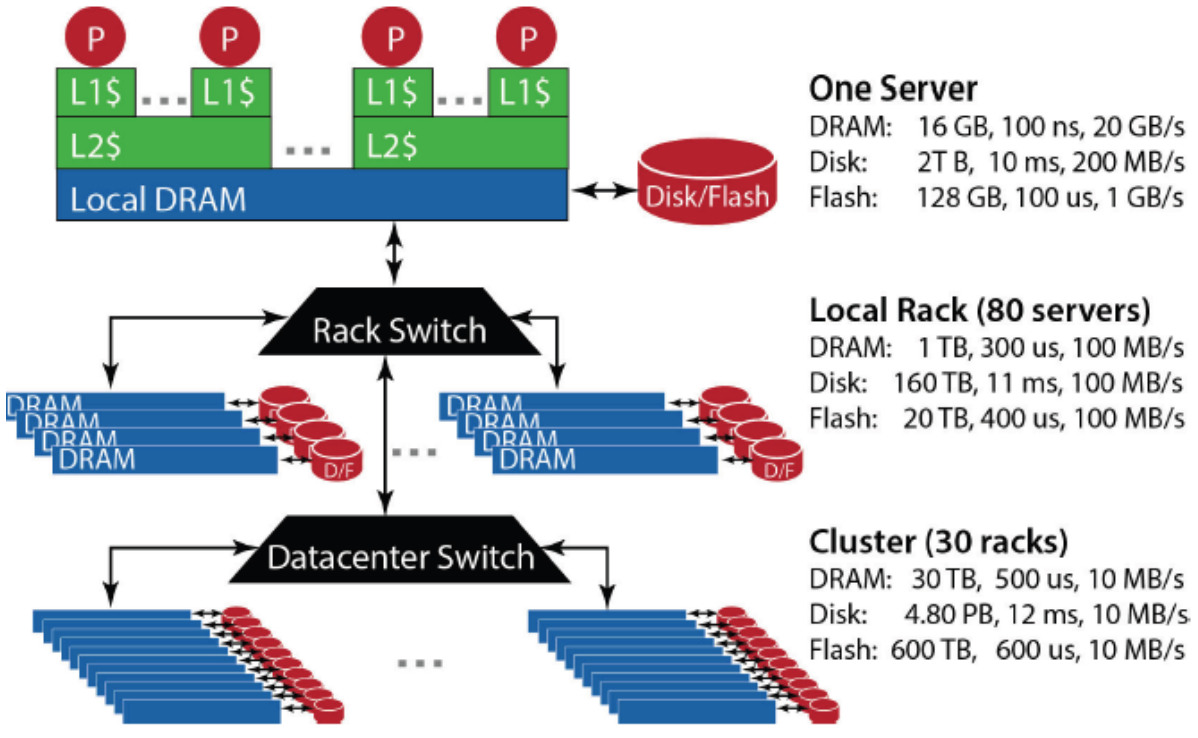
Top-of-Rack Architecture

- Rack of servers
  - Commodity servers
  - And top-of-rack switch
- Modular design
  - Preconfigured racks
  - Power, network, and storage cabling
- Aggregate to the next level

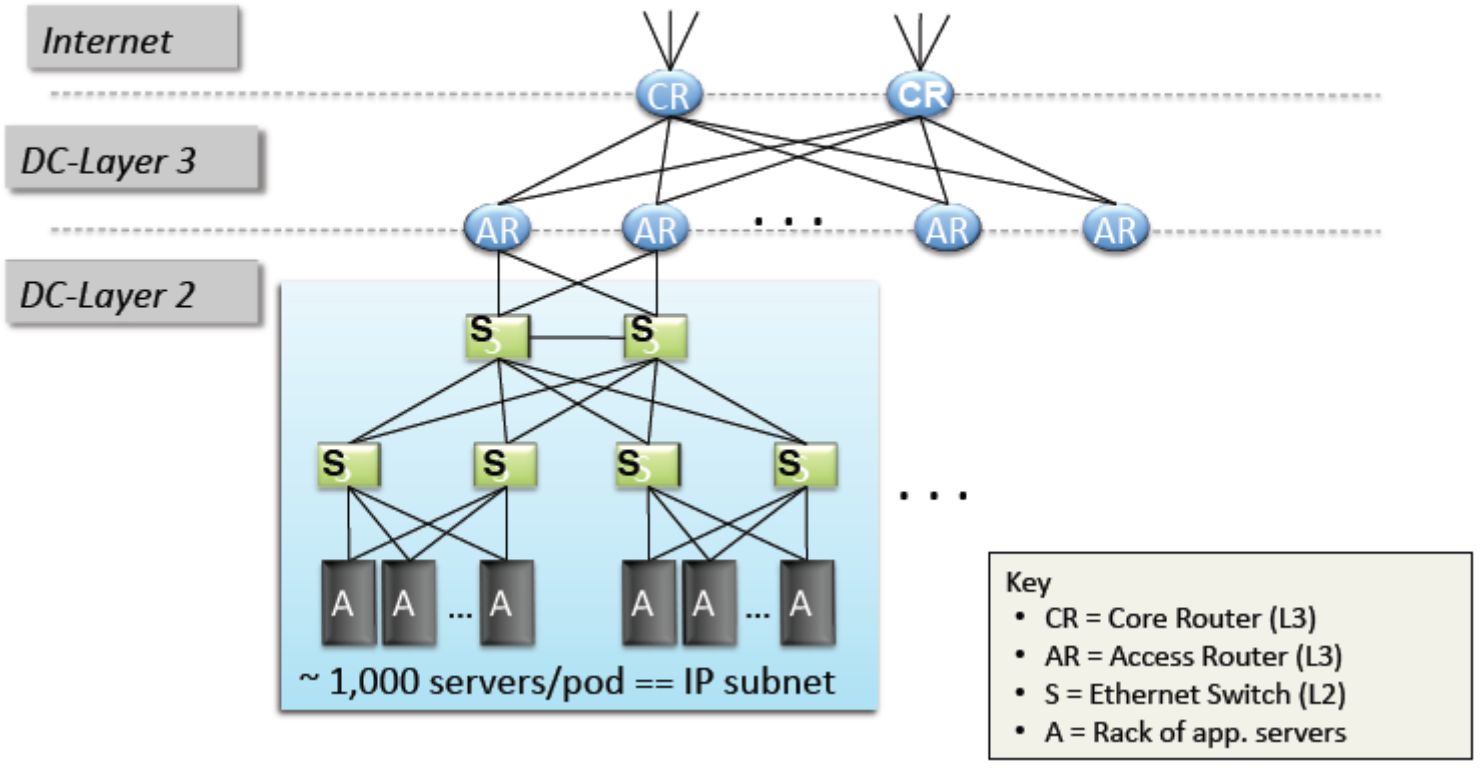




The storage hierarchy



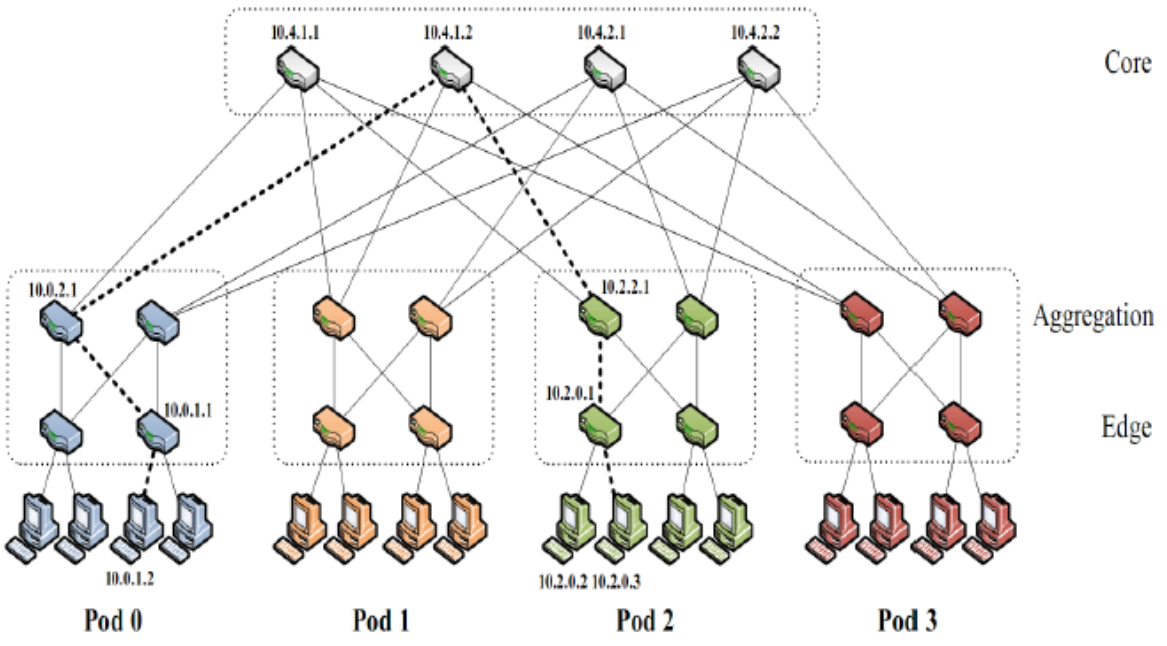
Layer 2 Pods w/L3 Backbone



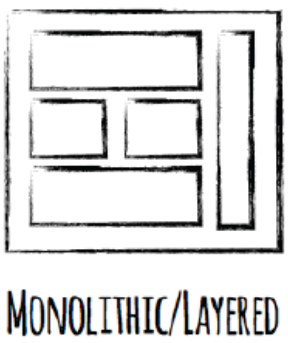
"Fat-Tree" Topology

FAT Tree-Based Solution

- An all Layer-3 solution
- Connect end-host together using a "fat-tree" topology
  - Infrastructure consist of cheap devices
    - Each port supports same speed as endhost
  - All devices can transmit at line speed if packets are distributed along existing paths
  - A k-port fat tree can support $k^3/4$ hosts



Microservices: Application Design is changing !!!



Properties of a Microservice

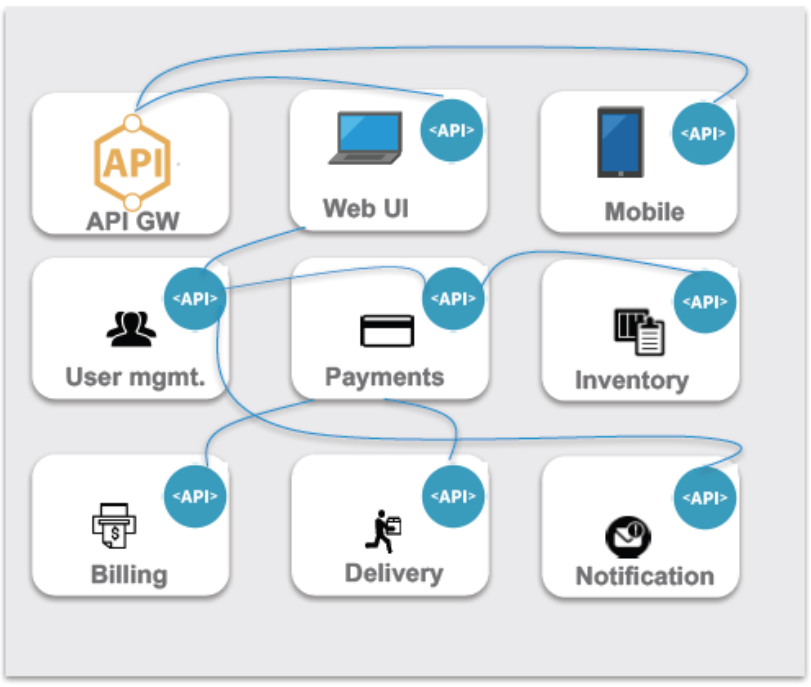
- ✓ Small code base
- ✓ Easy to scale, deploy and throw away
- ✓ Autonomous
- ✓ Resilient

Benefits of a Microservices Architecture

- ✓ A highly resilient, scalable and resource efficient application
- ✓ Enables smaller development teams
- ✓ Teams free to use the right languages and tools for the job
- ✓ Rapid application development

Cloud Native Application

Applications built using the "Microservices" architecture pattern

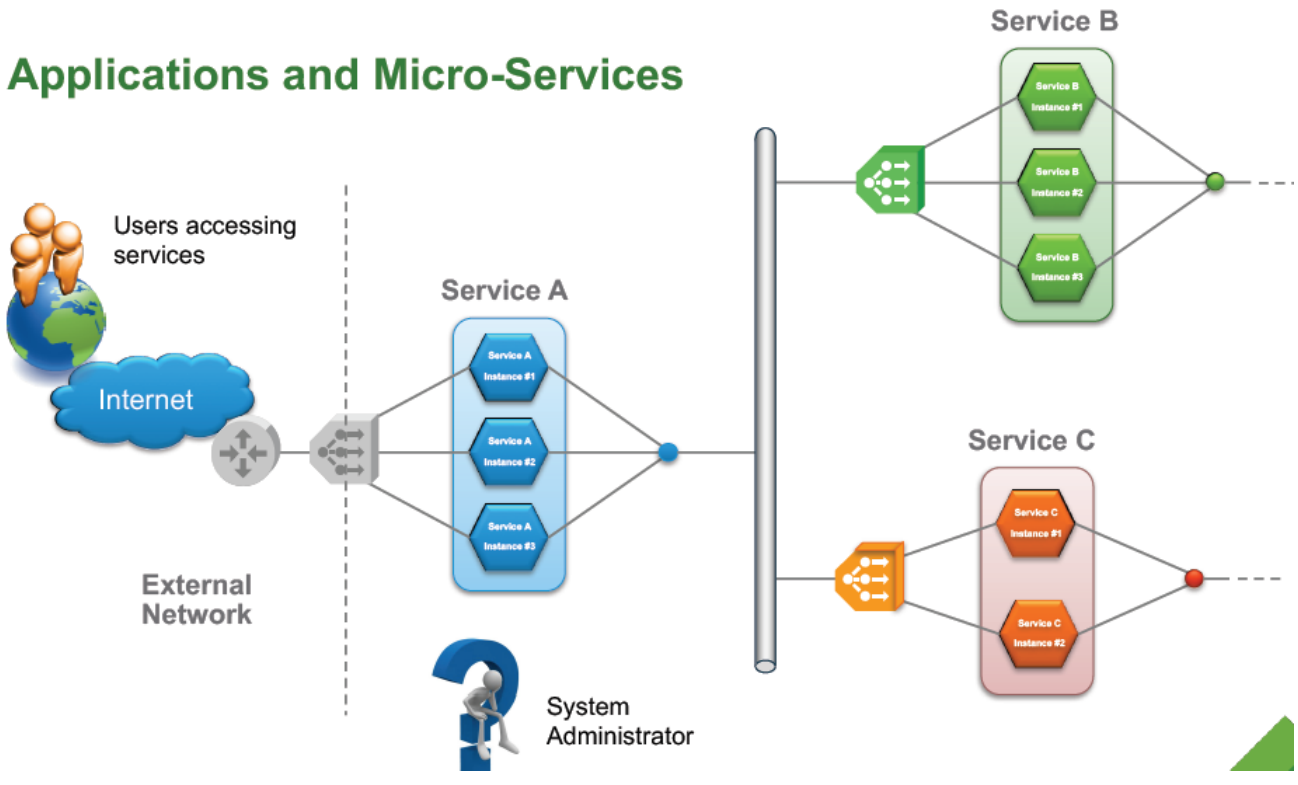


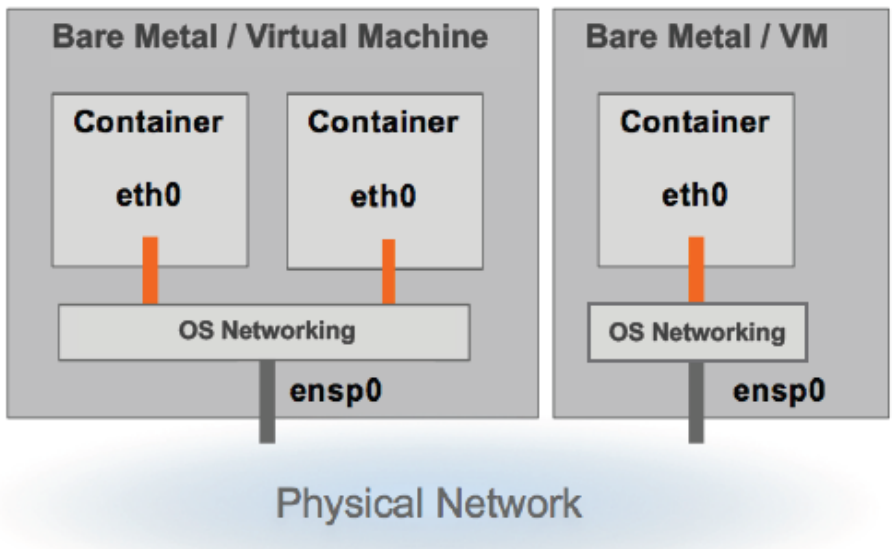
- **Loosely coupled distributed application**
  Application tier is decomposed into multiple web services
- **Datastore**
  Each micro service typically has its own datastore
- **Packaging**
  Each microservice is typically packaged in a "Container" image
- **Teams**
  Typically a team owns one or more Microservices

Challenges of running Microservices…

- Service Discovery
- Operational Overhead (100s+ of Services !!!)
- Distributed System... inherently complex
- Service Dependencies
  - service fan-out
  - dependency services running "hot"
- Traffic / Load each service can handle
- Service Health / Fault Tolerance
- Auto-Scale

Applications and Micro-Services

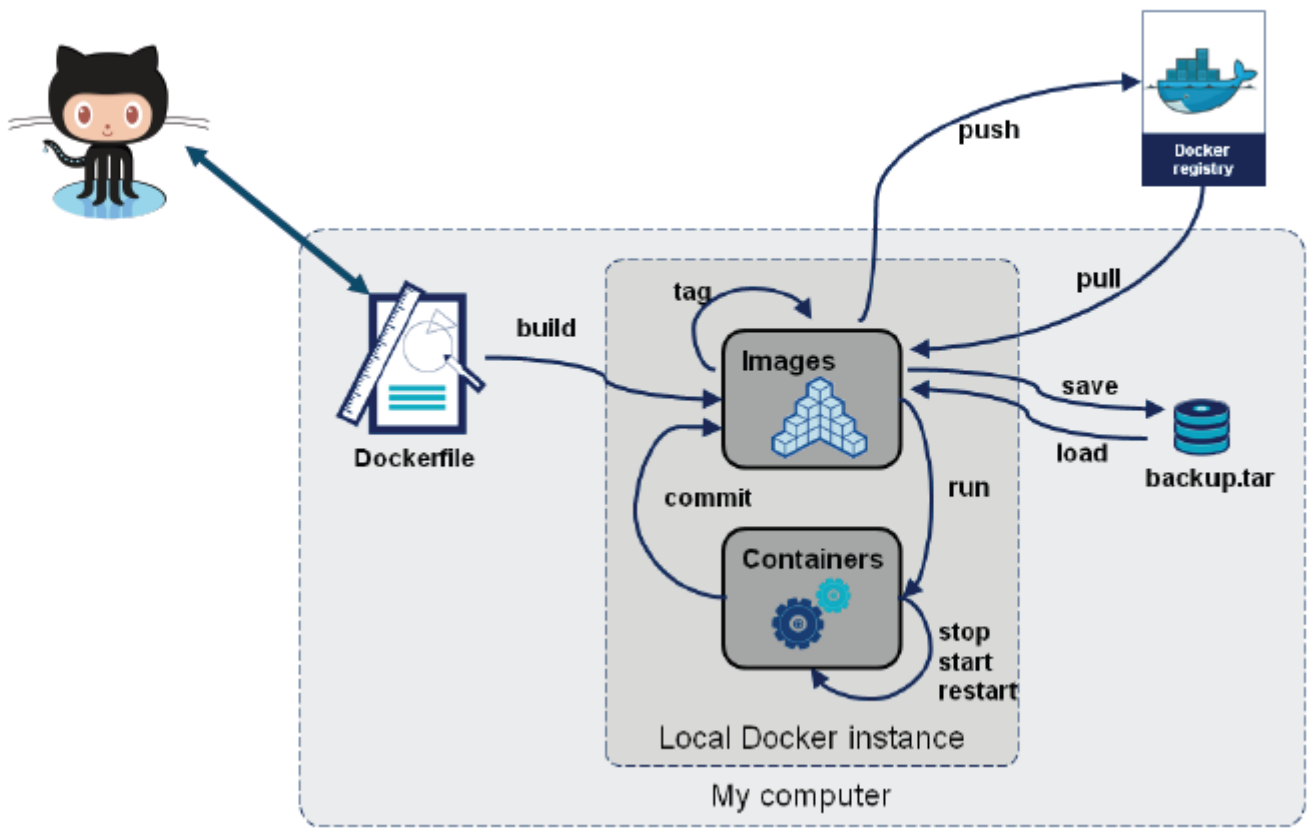




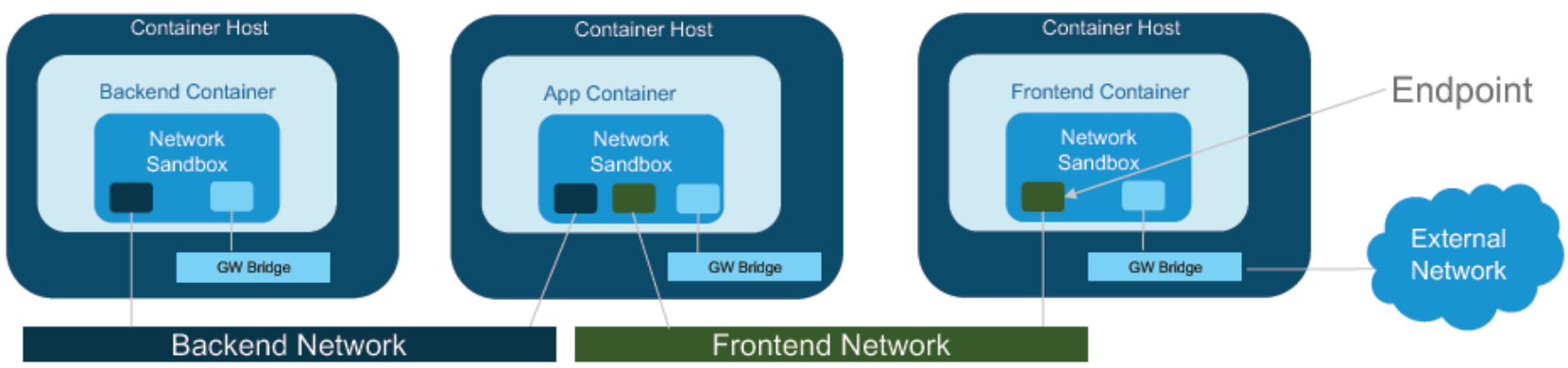
Minimalist Networking requirements:

- IP Connectivity in Container's Networl
- IP Address Management (IPAM) and Network Device Creation
- External Connectivity via Host NAT or Route Advertisement

Docker is a "Shipping Container" for Code



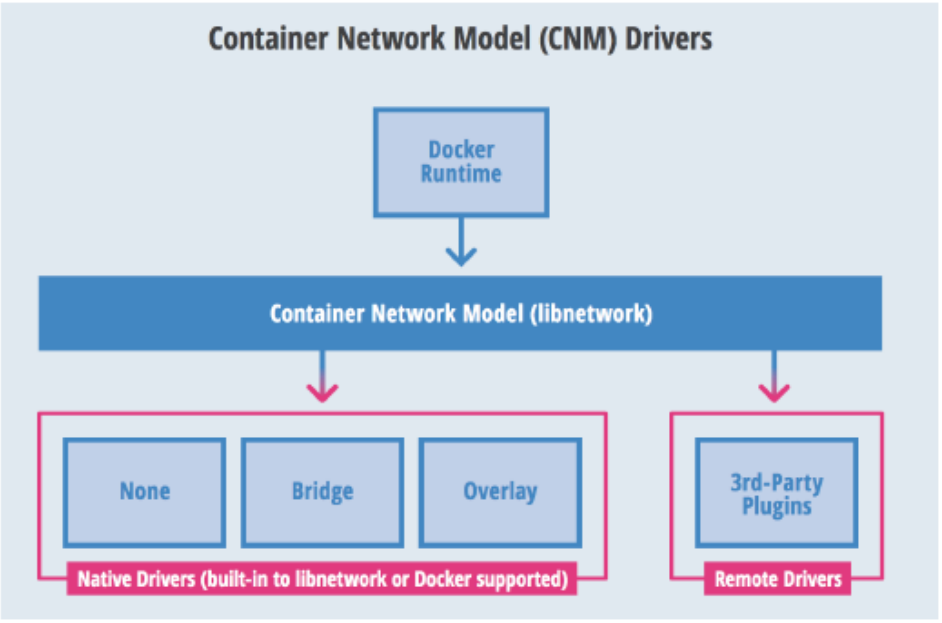
Docker: The Container Network Model (CNM) Interfacing



- **Sandbox**
  - A Sandbox contains the configuration of a container's network stack. This includes management of the container's interfaces, routing table and DNS settings. An implementation of a Sandbox could be a Linux Network Namespace, a FreeBSD Jail or other similar concept.
- **Endpoint**
  - An Endpoint joins a Sandbox to a Network. An implementation of an Endpoint could be a veth pair, an Open vSwitch internal port or similar
- **Network**
  - A Network is a group of Endpoints that are able to communicate with each-other directly. An implementation of a Network could be a VXLAN Segment, a Linux bridge, a VLAN, etc.

Container Network Model (CNM)

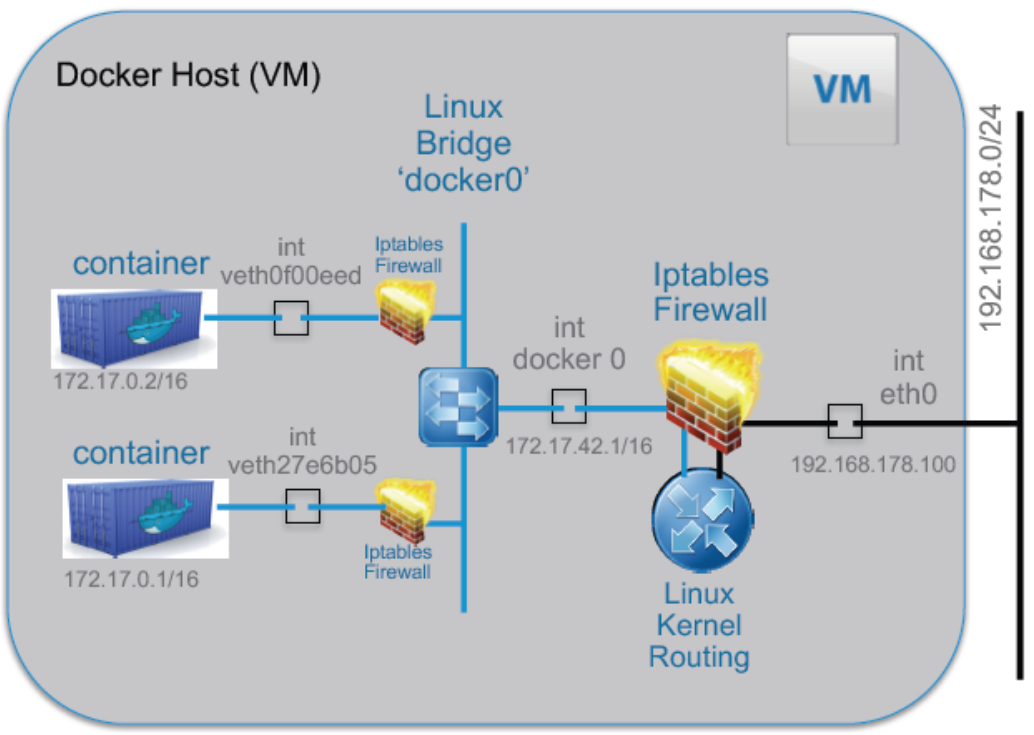
- The intention is for CNM (aka libnetwork) to implement and use any kind of networking technology to connect and discover containers
- Partitioning, Isolation, and Traffic Segmentation are achieved by dividing network addresses
- CNM does not specify one preferred methodology for any network overlay scheme



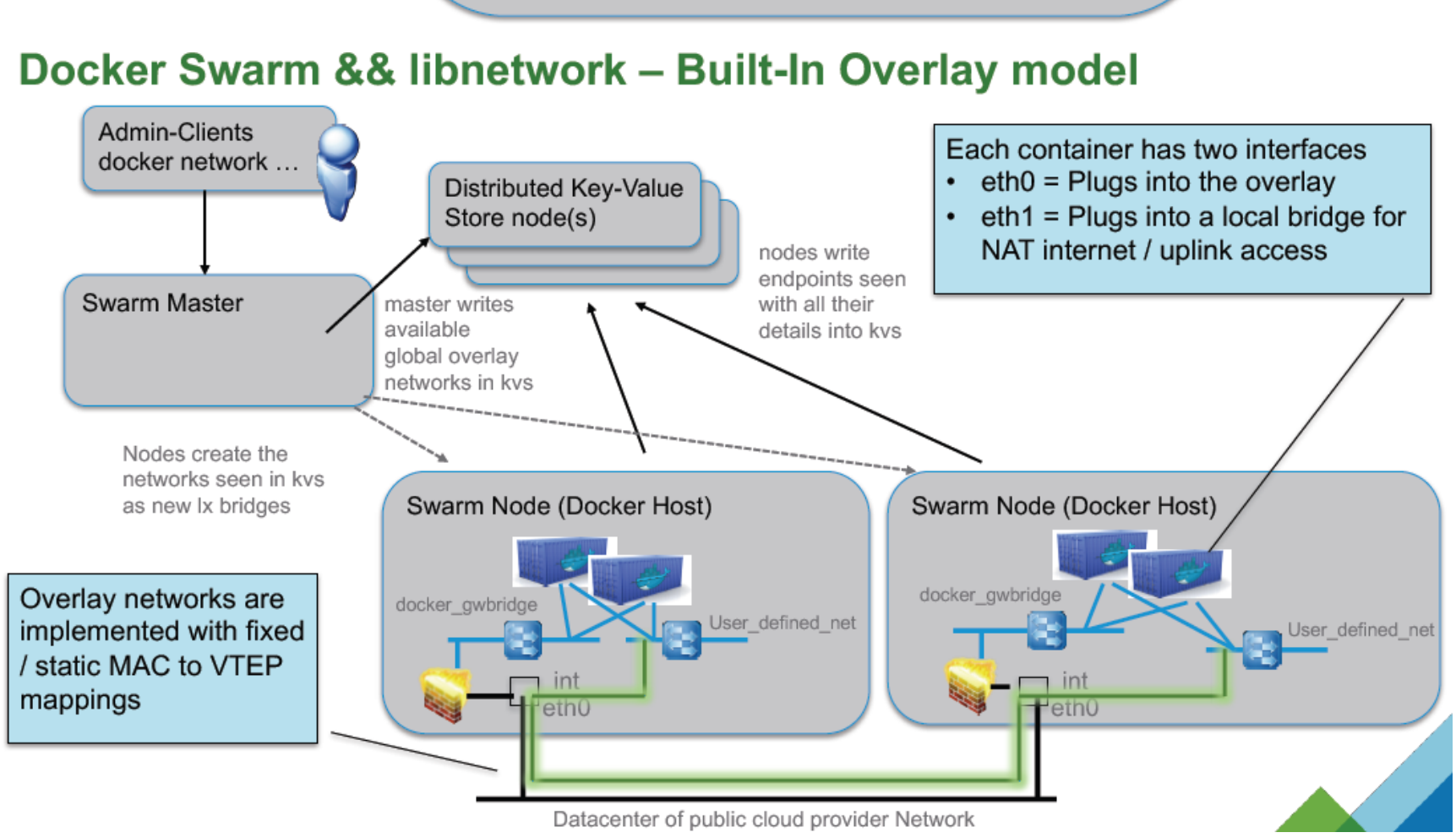

```
~/ docker network create \
   --driver overlay \
   --subnet 192.168.1.0/24 \
   multi-host-network
```

```
~/ docker network connect multi-host-network container1
```

Docker networking – Using the defaults



Docker Swarm && libnetwork – Built-In Overlay model



Docker Networking – key points

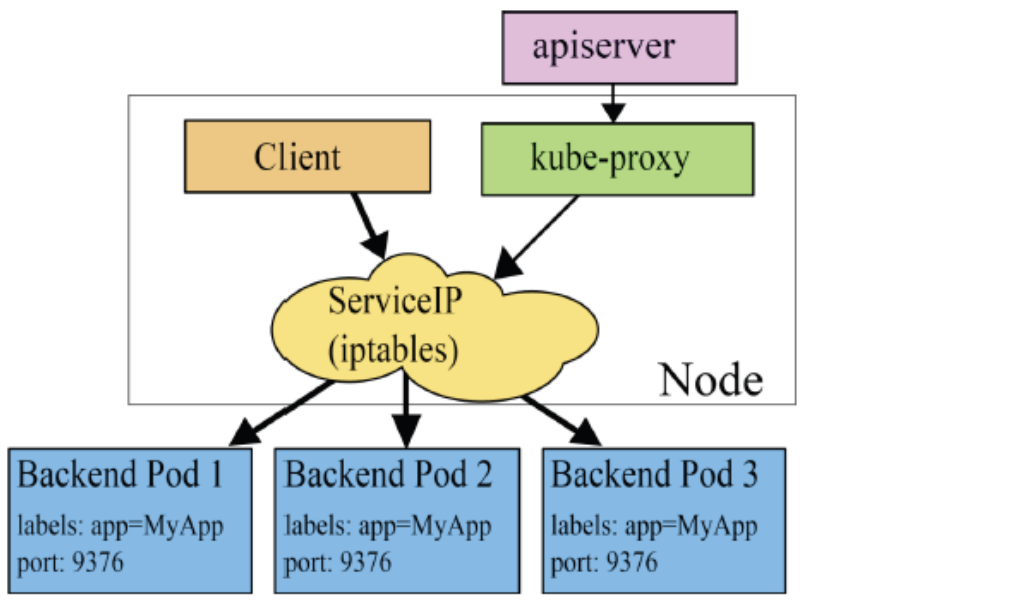
- Docker adopts the Container Network Model (CNM), providing the following contract between networks and containers:
  - All containers on the same network can communicate freely with each other
  - Multiple networks are the way to segment traffic between containers and should be supported by all drivers
  - Multiple endpoints per container are the way to join a container to multiple networks
  - An endpoint is added to a network sandbox to provide it with network connectivity
- Docker Engine can create overlay networks on a single host. Docker Swarm can create overlay networks that span hosts in the cluster
- A container can be assigned an IP on an overlay network. Containers that use the same overlay network can communicate, even if they are running on different hosts
- By default, nodes in the swarm encrypt traffic between themselves and other nodes. Connections between nodes are automatically secured through TLS authentication with certificates

Service Discovery

- Kubernetes provides two options for internal service discovery :
  - **Environment variable**: When a new Pod is created, environment variables from older services can be imported. This allows services to talk to each other. This approach enforces ordering in service creation.
  - **DNS**: Every service registers to the DNS service; using this, new services can find and talk to other services. Kubernetes provides the kube-dns service for this.

- Kubernetes provides several ways to expose services to the outside:
  - **NodePort**: In this method, Kubernetes exposes the service through special ports (30000-32767) of the node IP address.
  - **Loadbalancer**: In this method, Kubernetes interacts with the cloud provider to create a load balancer that redirects the traffic to the Pods. This approach is currently available with GCE
  - **Ingress Controller** : Since Kubernetes v1.2.0 it's possible to use Kubernetes ingress which includes support for TLS and L7 http-based traffic routing

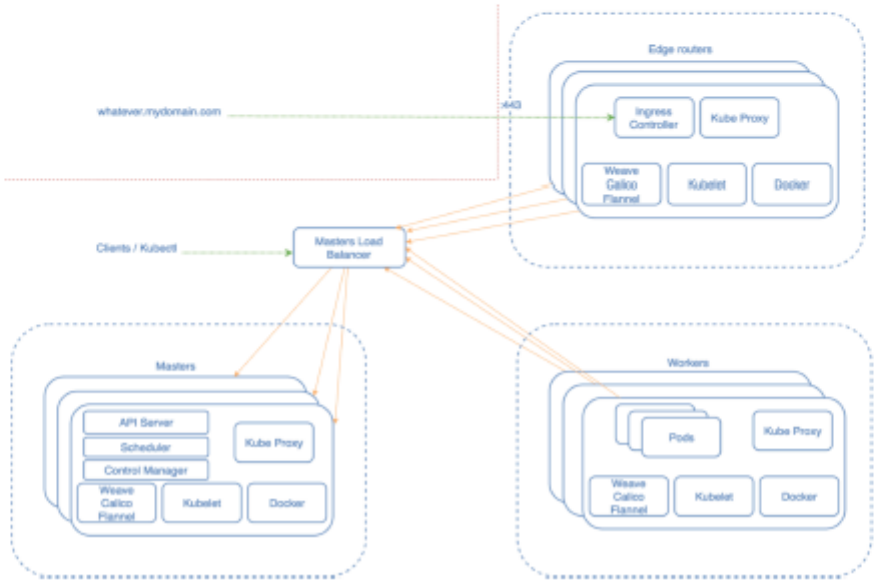
Internal Load Balancing



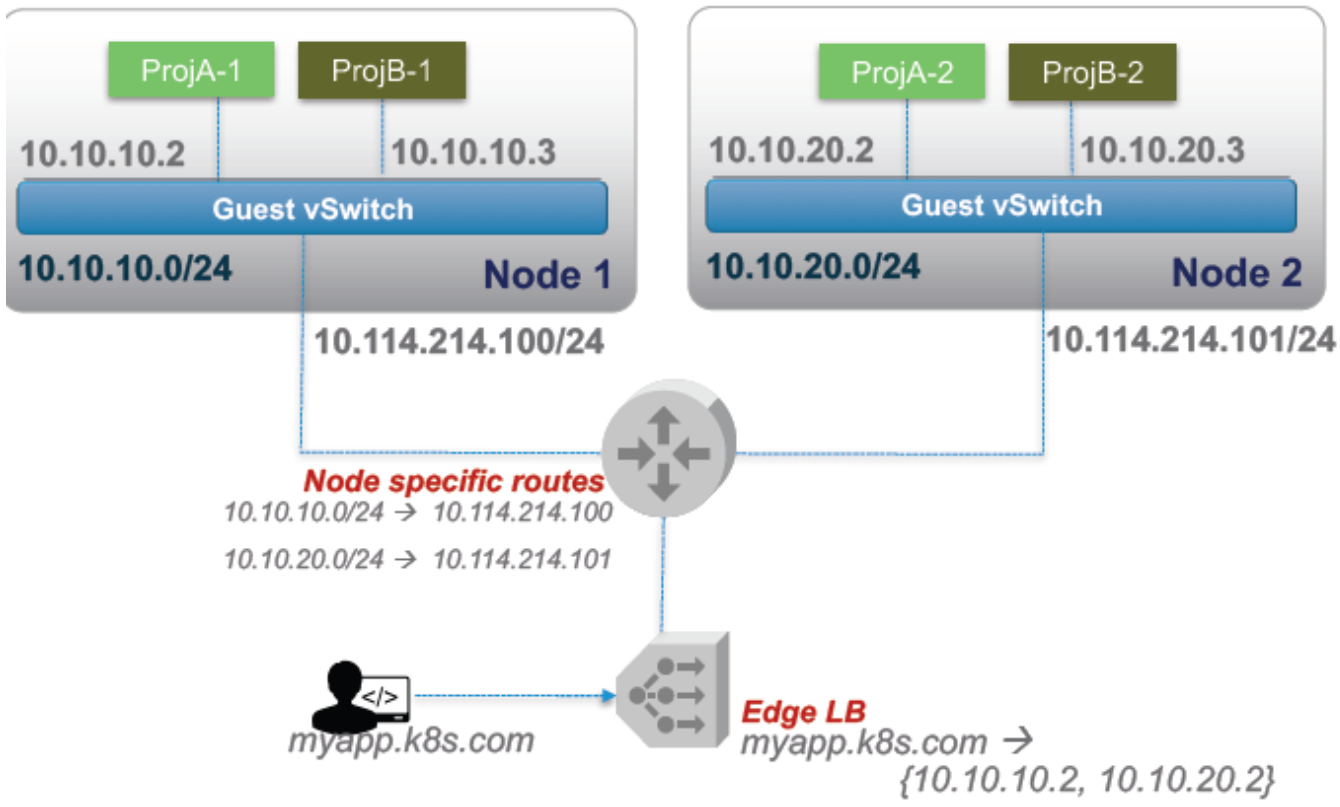
- Service name gets mapped to Virtual IP and port using Skydns
- Kube-proxy watches Service changes and updates IPtables. Virtual IP to Service IP, port remapping is achieved using IP tables
- Kubernetes does not use DNS based load balancing to avoid some of the known issues associated with it

Ingress Load Balancing w/t Ingress Controller

- An Ingress is a collection of rules that allow inbound connections to reach the cluster services
- It can be configured to give services externally-reachable urls, load balance traffic, terminate SSL, offer name based virtual hosting etc
  - Users request ingress by POSTing the Ingress resource to the API server.
- In order for the Ingress resource to work, the cluster must have an Ingress controller running. The Ingress controller is responsible for fulfilling the Ingress dynamically by watching the ApiServer's /ingresses endpoint.

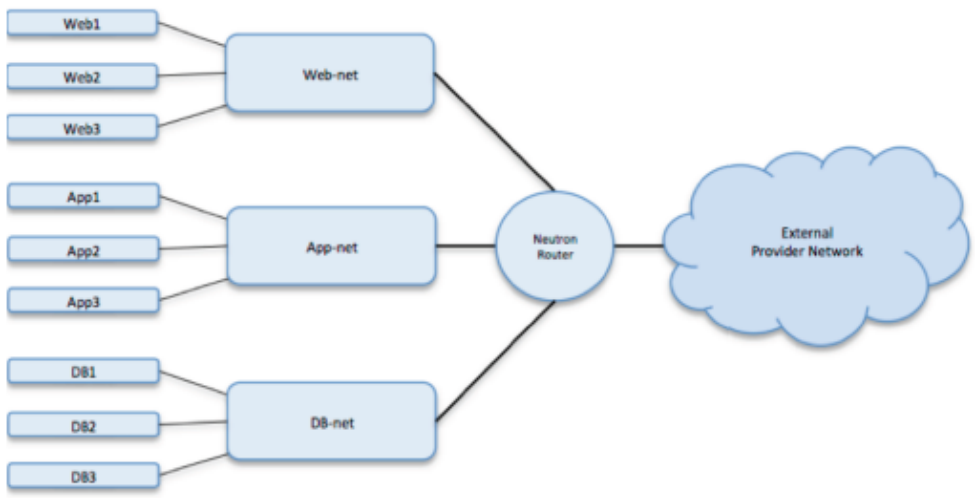


Networking for Services

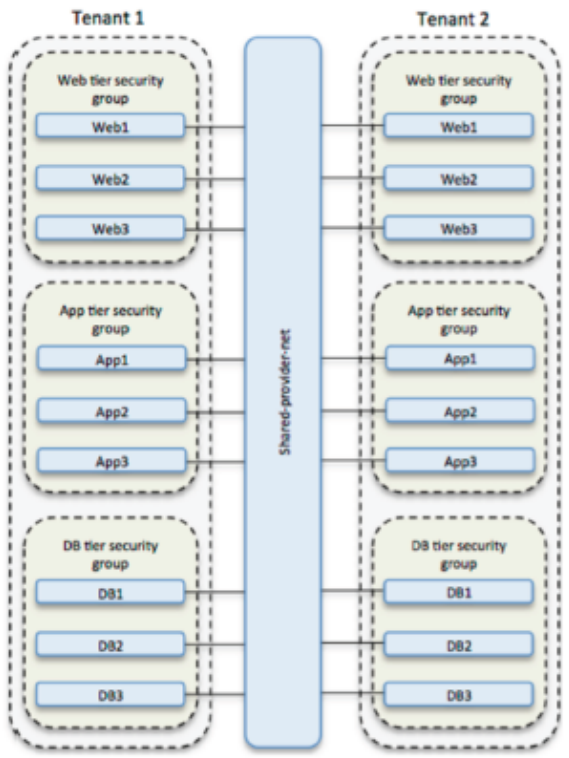


- K8s default networking configures
  - Routable IP per POD
  - Subnet per node / minion
- K8s **Service** provides East-West Load Balancing
- Provides DNS based service discovery – Service Name to IP
- Network Security Policy – in beta
- Not in K8s scope
  - Edge LB – e.g. external to frontend pods
  - Routing of a subnet to k8s node

Multi-Tenancy and Application tiering



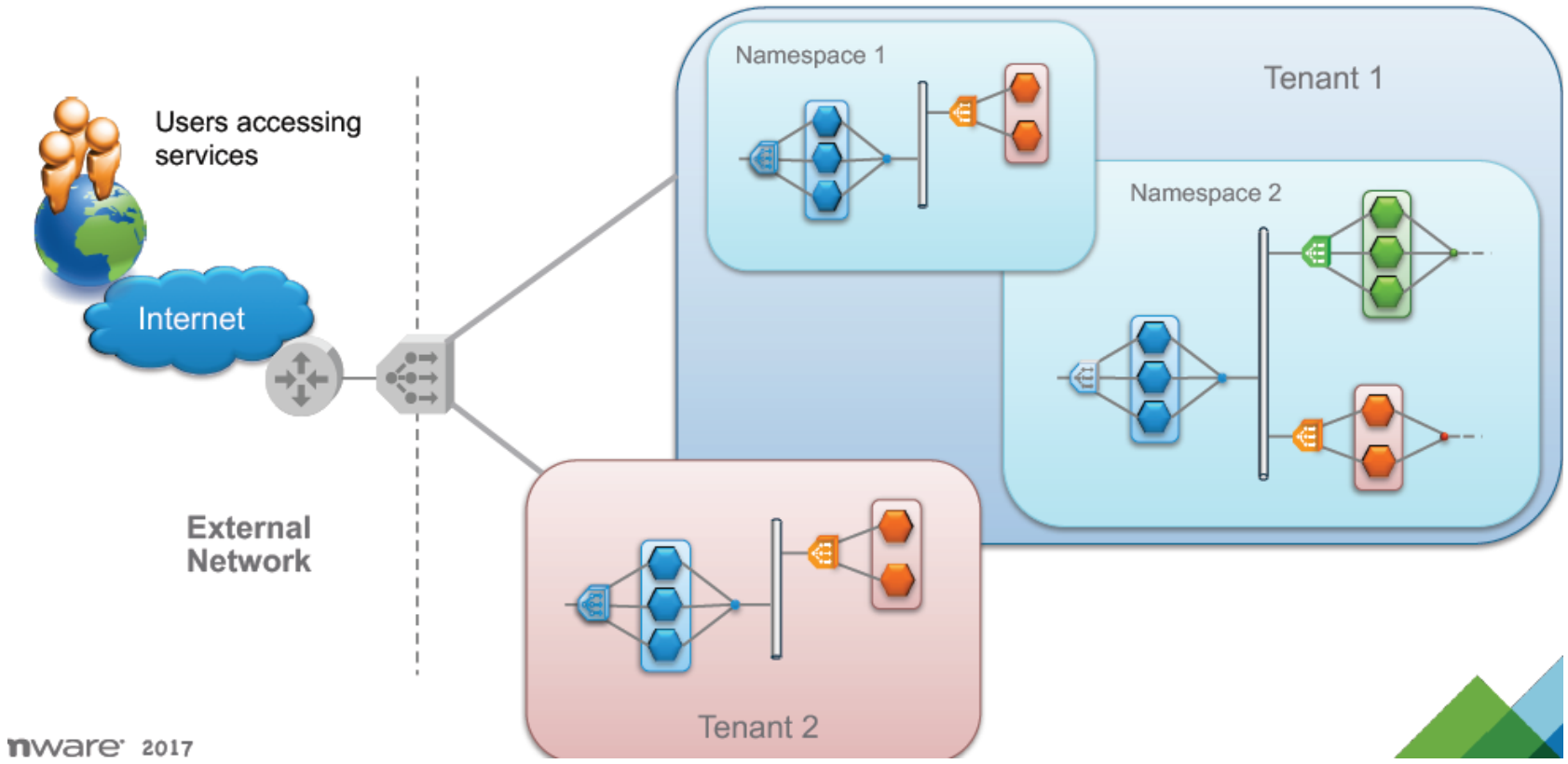
Traditional application tiering



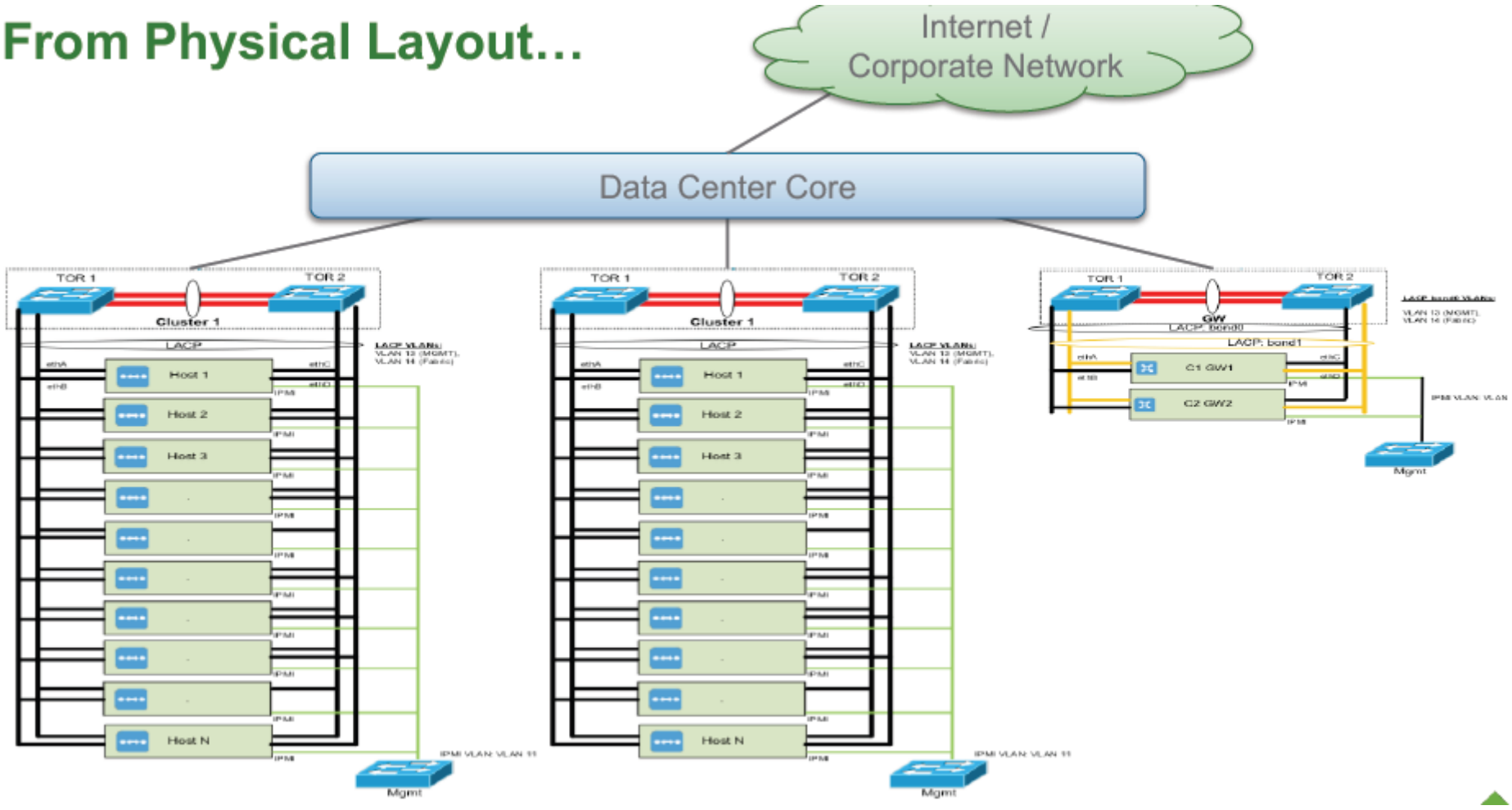
Application tiering/security zones using security groups

Example of Multi-Tenancy Model

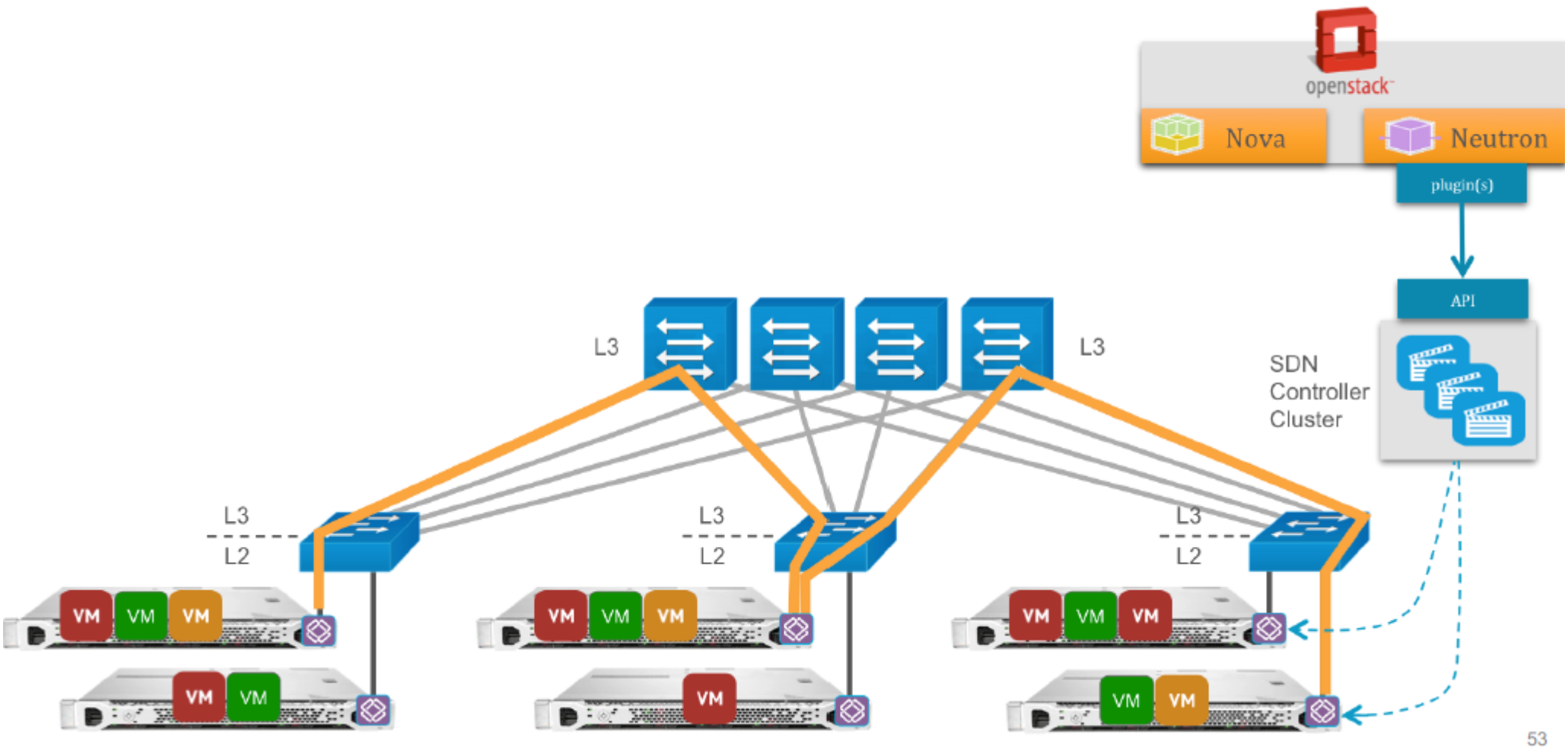
Multi-Tenancy, Namespaces && Microsegmentation



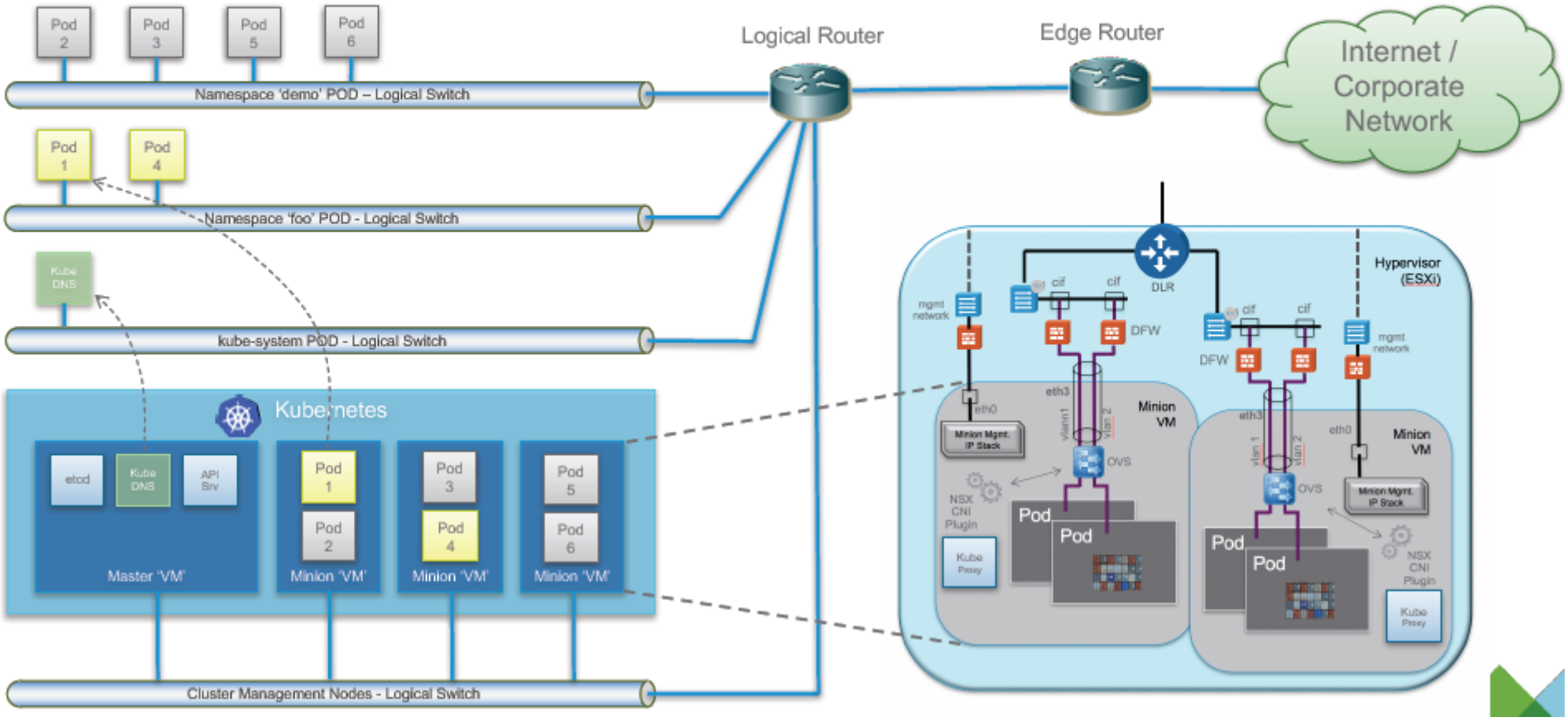
From Physical Layout…

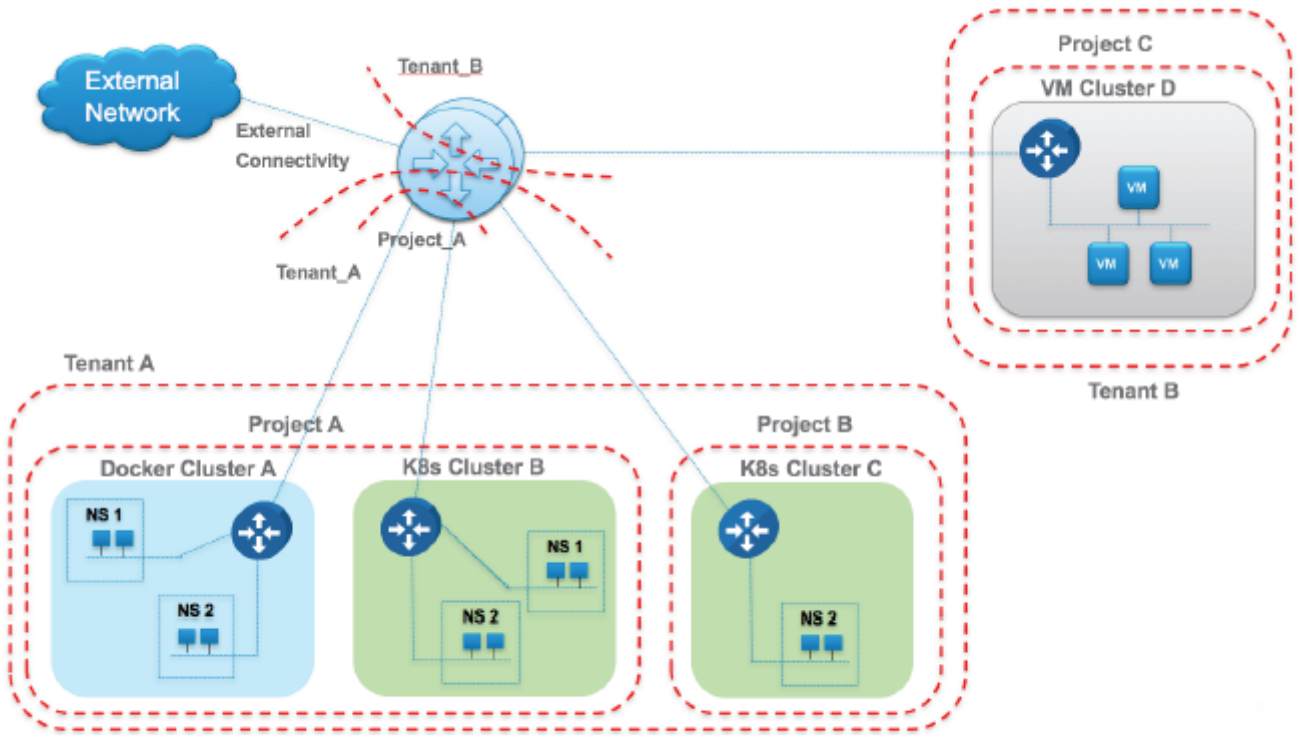


…to Overlay-based Networking Model…



…to Cluster Deployment on Logical Networks…





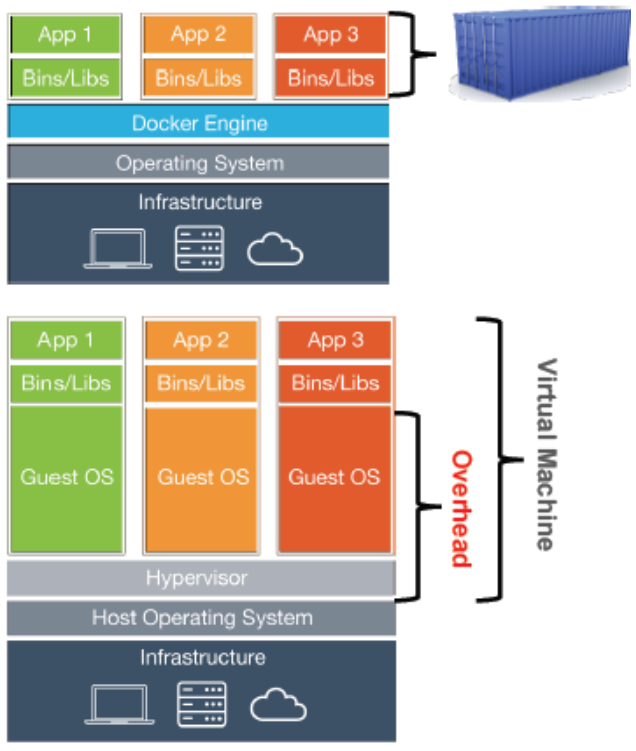
Multi-Tenancy deployment and Networking constrains

Containers

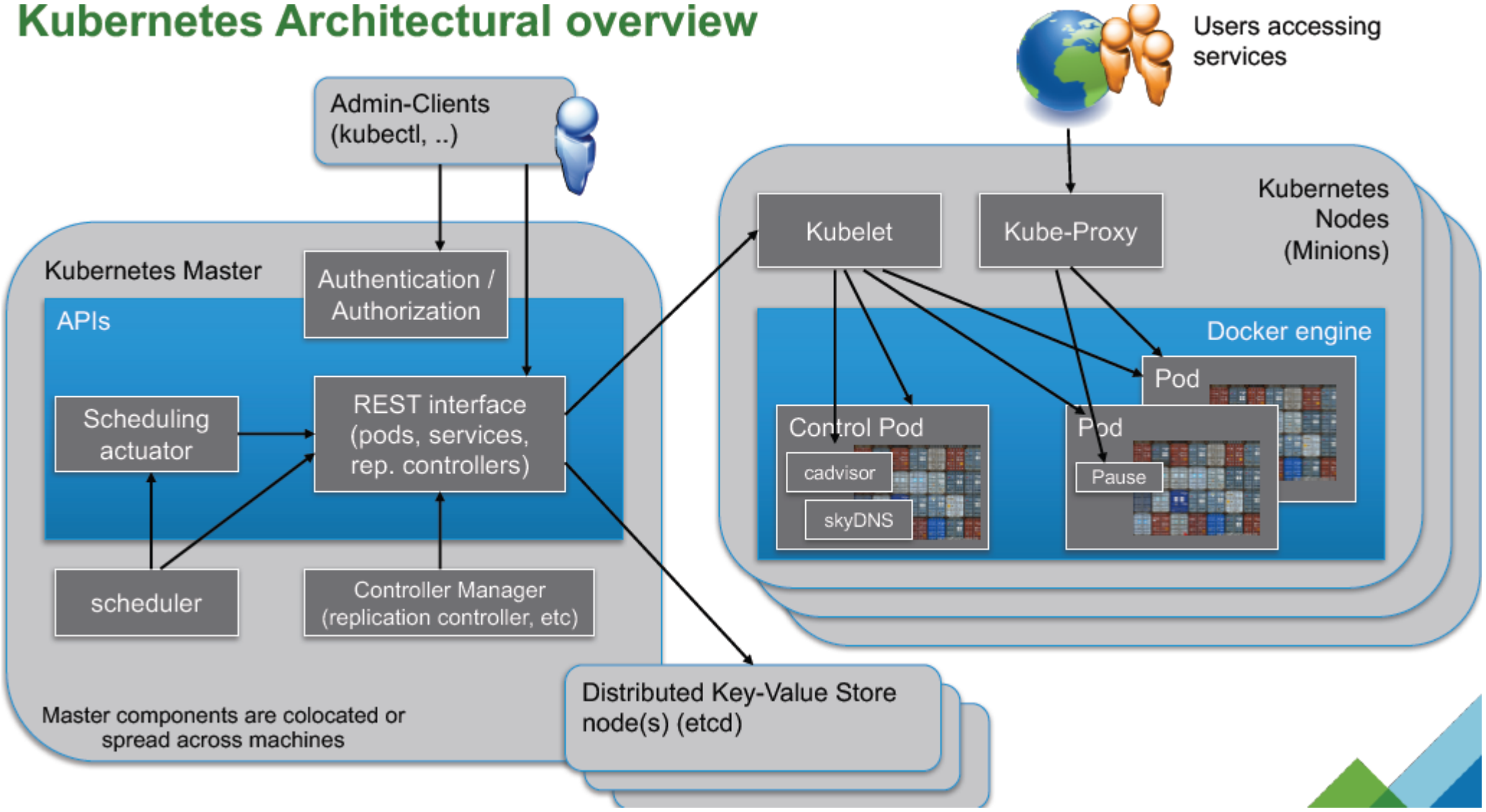
- Virtualization of application instead of hardware
- Runs on top of the core OS (Linux or Windows)
- Doesn't require dedicated CPU, Memory, Network —managed by core OS
- Optimizes Infrastructure—speed and density

"Containerization seems poised to offer both a complement and a viable alternative to server virtualization"

(1) IDC



Kubernetes Architectural overview



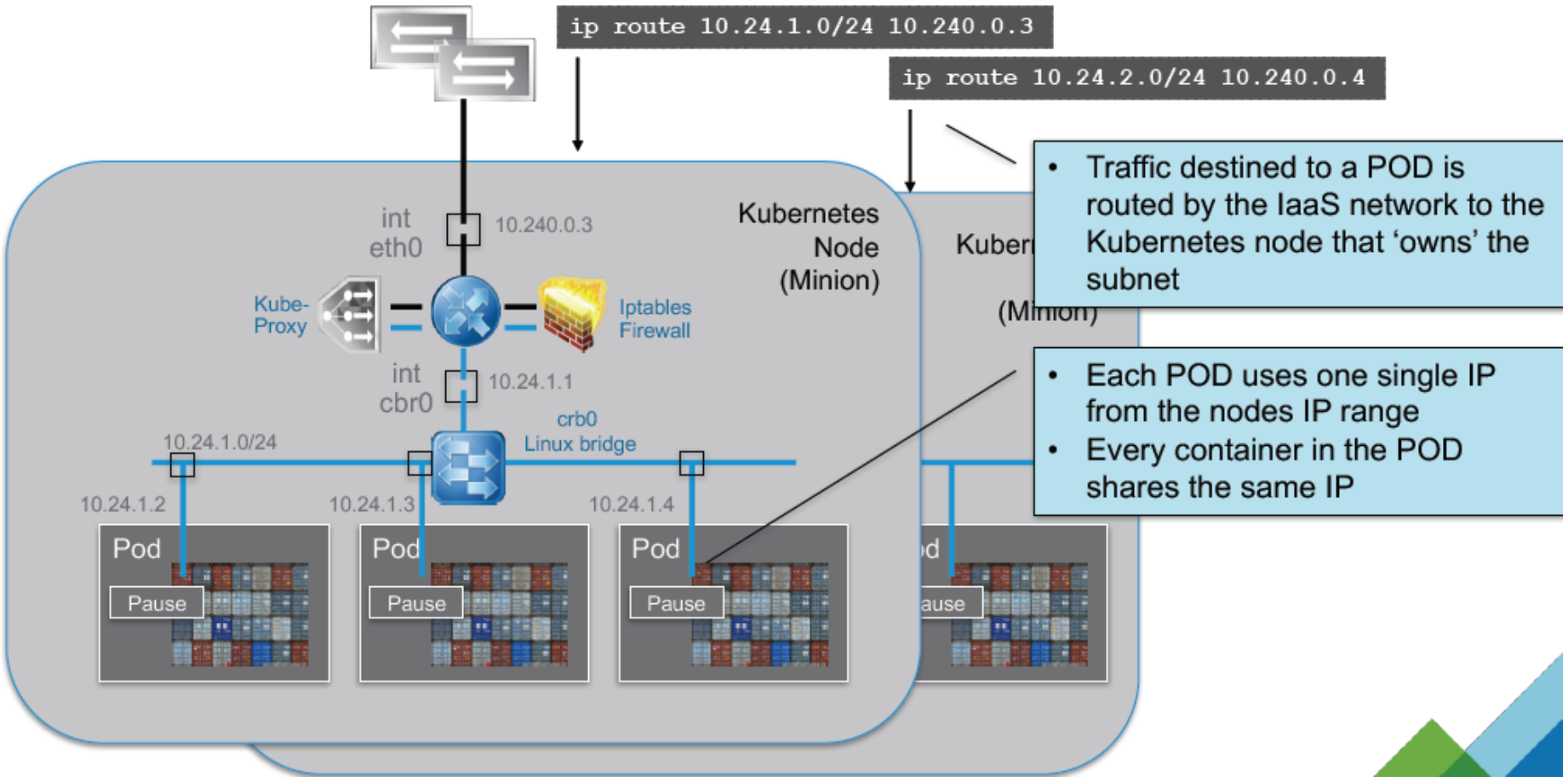
Quick Overview of Kubernetes

Kubernetes (k8s) = Open Source Container Cluster Manager

- **Pods:** tightly coupled group of containers
- **Replication controller:** ensures that a specified number of pod "replicas" are running at any one time.
- **Networking:** Each pod gets its own IP address
- **Service:** Load balanced endpoint for a set of pods with internal and external IP endpoints
- **Service Discovery**: Using env variable injection or SkyDNS with the Service

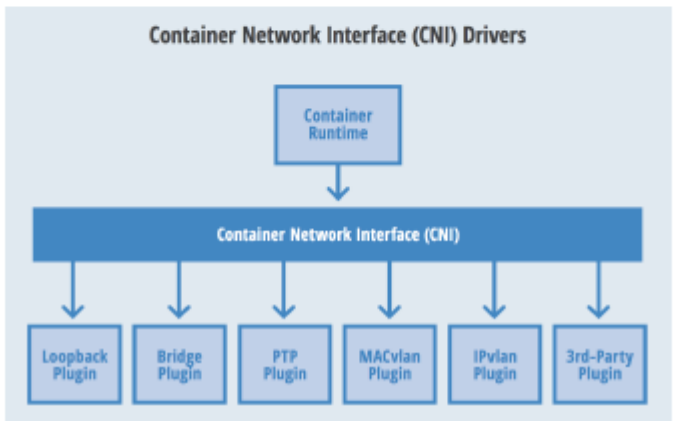
- Uses etcd as distributed key-value store
- Has its roots in ‘borg’, Google’s internal container cluster management

Kubernetes Node (Minion) – Docker networking details



Container Network Interface (CNI)

- Kubernetes uses the Container Network Interface (CNI) specification and plug-ins to orchestrate networking
- Very differently from CNM, CNI is capable of addressing other containers' IP addresses without resorting to network address translation (NAT)
- Every time a POD is initialized or removed, the default CNI plug-in is called with the default configuration
- This CNI plug-in creates a pseudo interface, attaches it to the relevant underlay network, sets IP Address / Routes and maps it to the POD namespace




```
{
    "name": "net",
    "type": "bridge",
    "bridge": "br-int",
    "isGateway": true,
    "ipMasq": false,
    "ipam": {
        "type": "host-local",
        "subnet": "10.96.0.64/26"
    }
}
```

/etc/cni/net.d/10-bridge.conf

Kubernetes Networking – key points

- Kubernets adopts the Container Network Interface (CNI) model to provide a contract between networks and containers

- From a user perspective, provisioning networking for a container involves two steps:
  - Define the network JSON
  - Connect container to the network

- Internally, CNI provisioning involves three steps:
  - Runtime create a network namespace and gives it a name
  - Invokes the CNI plugin specified in the "type" field of the network JSON. Type field refers to the plugin being used and so CNI invokes the corresponding binary
  - Plugin code in turn will create a veth pair, check the IPAM type and data in the JSON, invoke the IPAM plugin, get the available IP, and finally assign the IP address to the interface

Container Networking Specifications

Container Networking Model

CNM

- Specification proposed by Docker, adopted by projects such as libnetwork
- Plugins built by projects such as Weave, Project Calico and Kuryr
- Supports only Docker runtime

Container Networking Interface

CNI

- Specification proposed by CoreOS and adopted by projects such as Kubernetes, Cloud Foundry and Apache Mesos
- Plugins built by projects such as Weave, Project Calico, Contiv Networking
- Supports any container runtime

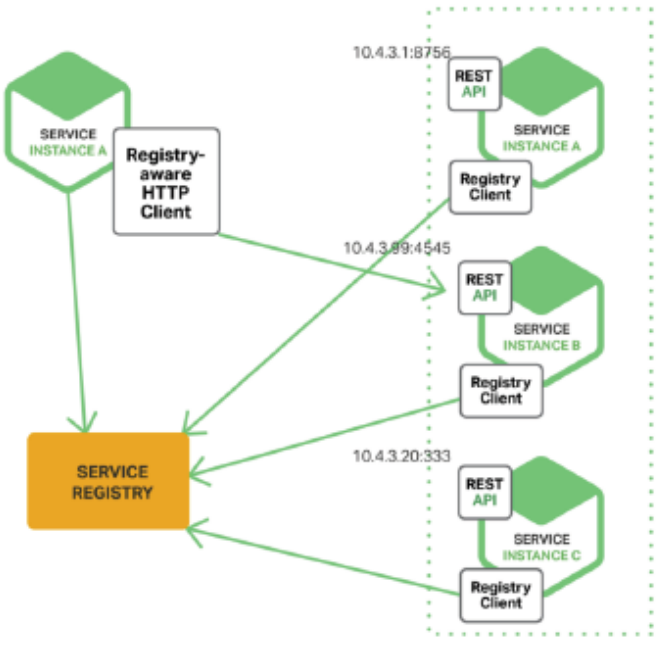
CNI and CNM commonalities…

- CNI and CNM models are both driver-based
  - provide "freedom of selection" for a specific type of container networking
- Multiple Network drivers can be active and used concurrently
  - 1-1 mapping among network type and network driver
- Containers are allowed to join one or more networks
- Container runtime can lunch network in its own namespace
  - delegate to the network driver the responsibility of connecting the container to the network

Container Networking Models	Container Network Interface (CNI)	Docker Libnetwork
Container Platform	kubernetes, OPENSHIFT ENTERPRISE, Pivotal Cloud Foundry, MESOSPHERE	docker SWARM
Pluggable Network Stack	YES, YES, WORK IN PROGRESS, WORK IN PROGRESS	YES

Client vs Server side Service discovery

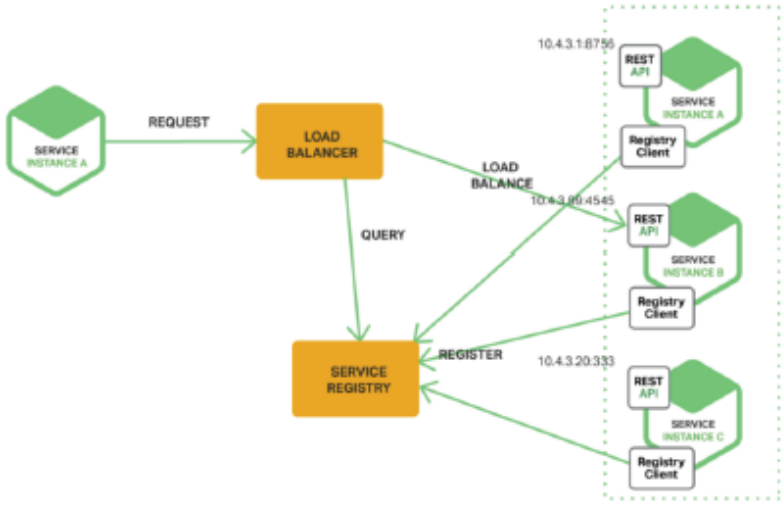
Client Discovery



- Client talks to Service registry and does load balancing.
- Client service needs to be Service registry aware.

eg: Netflix OSS

Server Discovery



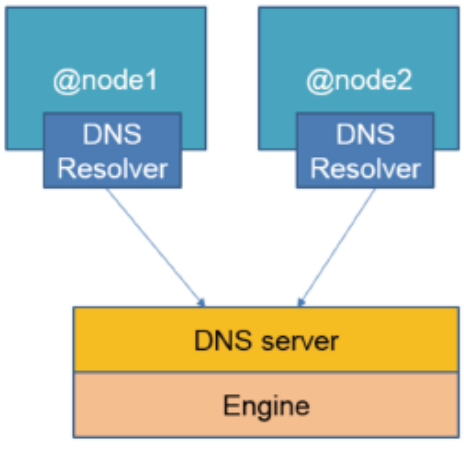
- Client talks to load balancer and load balancer talks to Service registry.
- Client service need not be Service registry aware

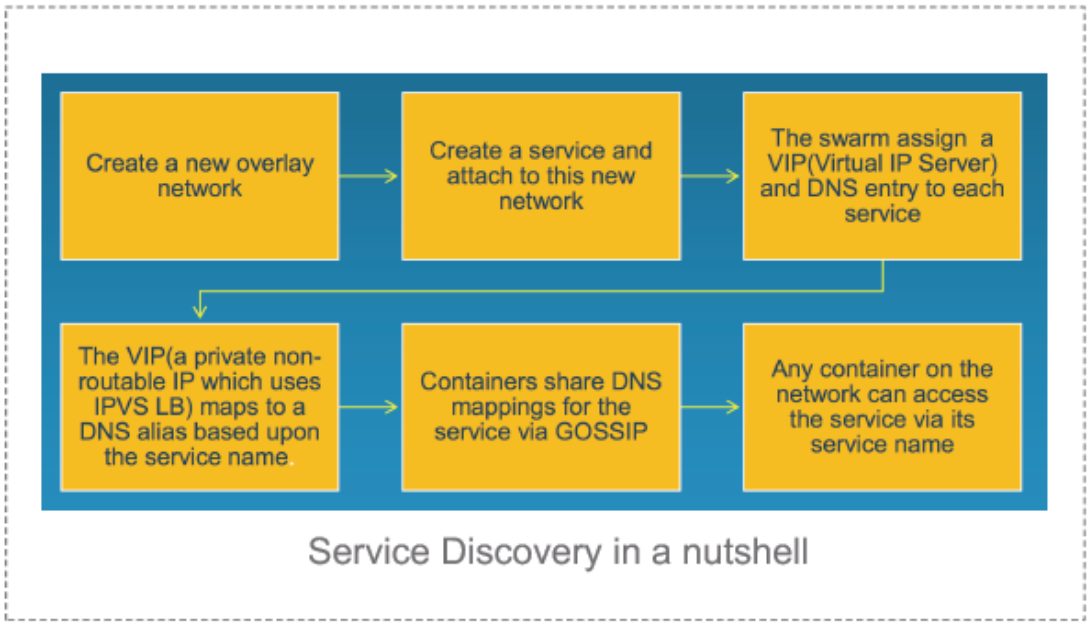
eg: Consul, AWS ELB, K8s, Docker

Service Discovery

Service Discovery:

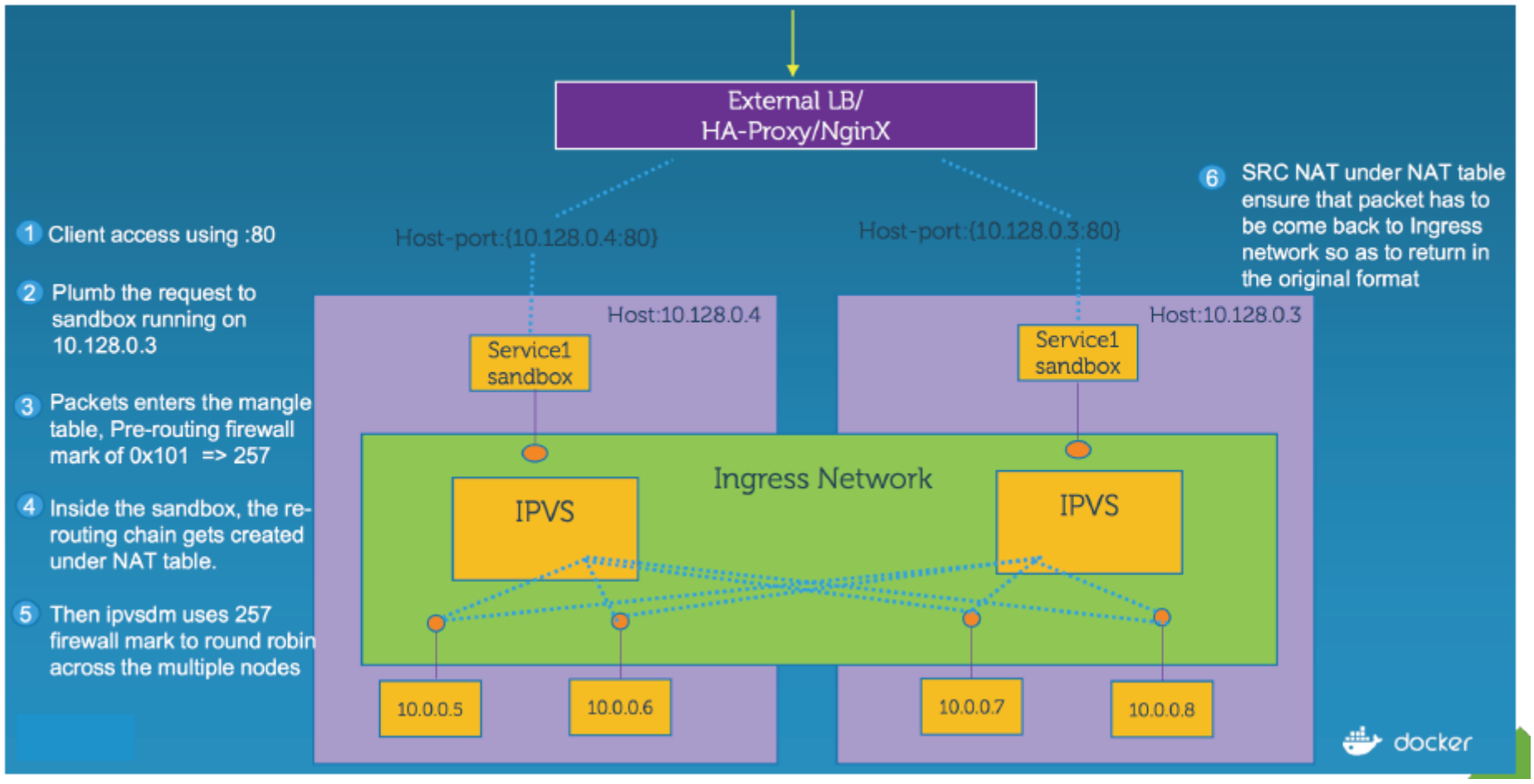
- Provided by Embedded DNS
- Highly Available
- Uses Network Control Plane to learn state
- Can be used to discover services and containers





Service Discovery in a nutshell

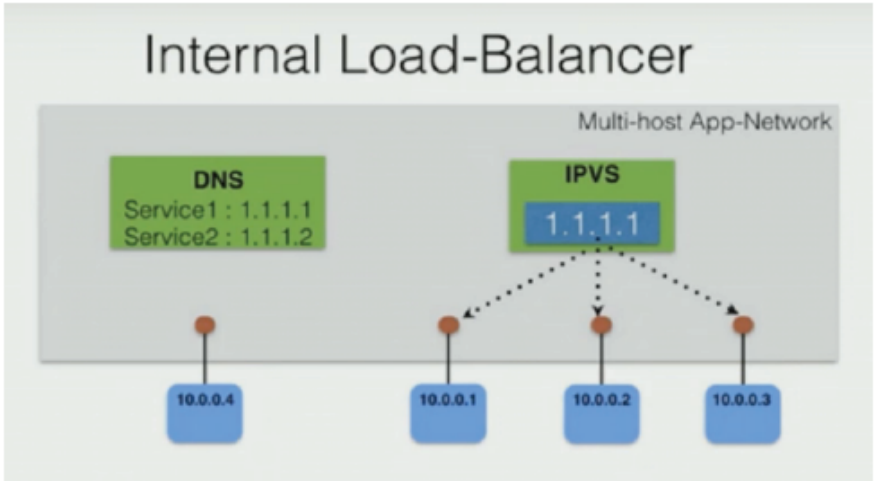
Ingress Load Balancing



TODO: Service Discovery

Internal Load Balancer - IPVS

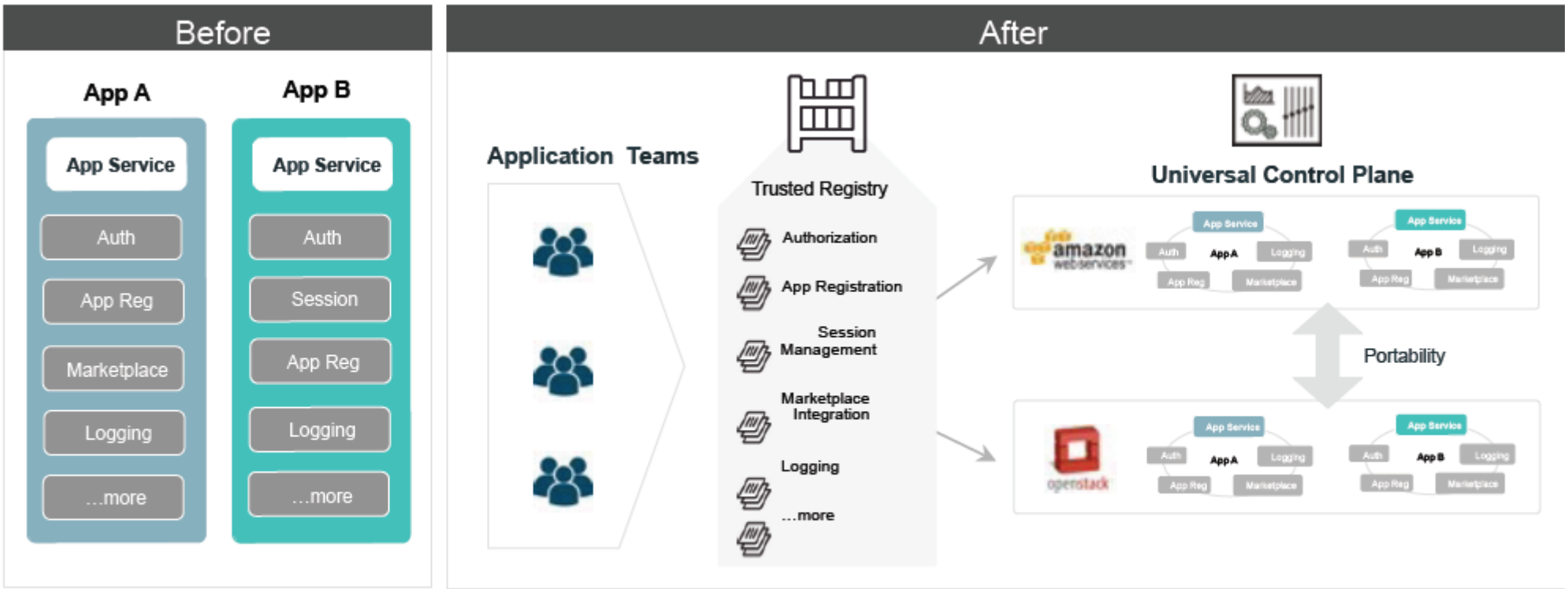
- IPVS (IP Virtual Server) implements transport-layer load balancing inside the Linux kernel, so called Layer-4 switching
- It's based on Netfilter and supports TCP, SCTP & UDP, v4 and v7
- IPVS is dynamically configurable, supports 8+ balancing methods, provides health checking



	Container	Virtual Machine	Bare-Metal x86 Server
Underlying Platform	OS on Virtual Machine or Bare-Metal x86 Server	Hypervisor on Bare-Metal x86 Server	N/A
Performance: Speed and Consistency	Average	Average	Fastest
Provisioning Time	Seconds	Minutes	Hours
Tenant Isolation Enforcement	OS Kernel	Hypervisor	Physical
Ideal Application Types	Mode 2	Mode 1 or Mode 2	Mode 1 or Mode 2
Configuration and Reconfiguration Flexibility	Highest	Medium	Lowest
Host Consolidation Density	Maximum	Average	None
Application Portability	Application Packaging/ Manifest*	VM Image, VM Migration Tools	Backup and Restore, ISO Images
Granularity	Extremely Small	Average	Largest

*While application portability is somewhat easier in container environments that are leveraging a container management and orchestration solution, portability should not assumed to be universal — differences in the underlying host OS below the containers could still present some interoperability challenges.

Source: Gartner (September 2015)



Common services in monoliths are turned turned into base applications stored in the Trusted Registry available to all app teams

Teams request into central IT maintained portal/registry to provision infrastructure and pull base images

Monoliths are now micro services applications. Each app has it's own containers based on the same base image

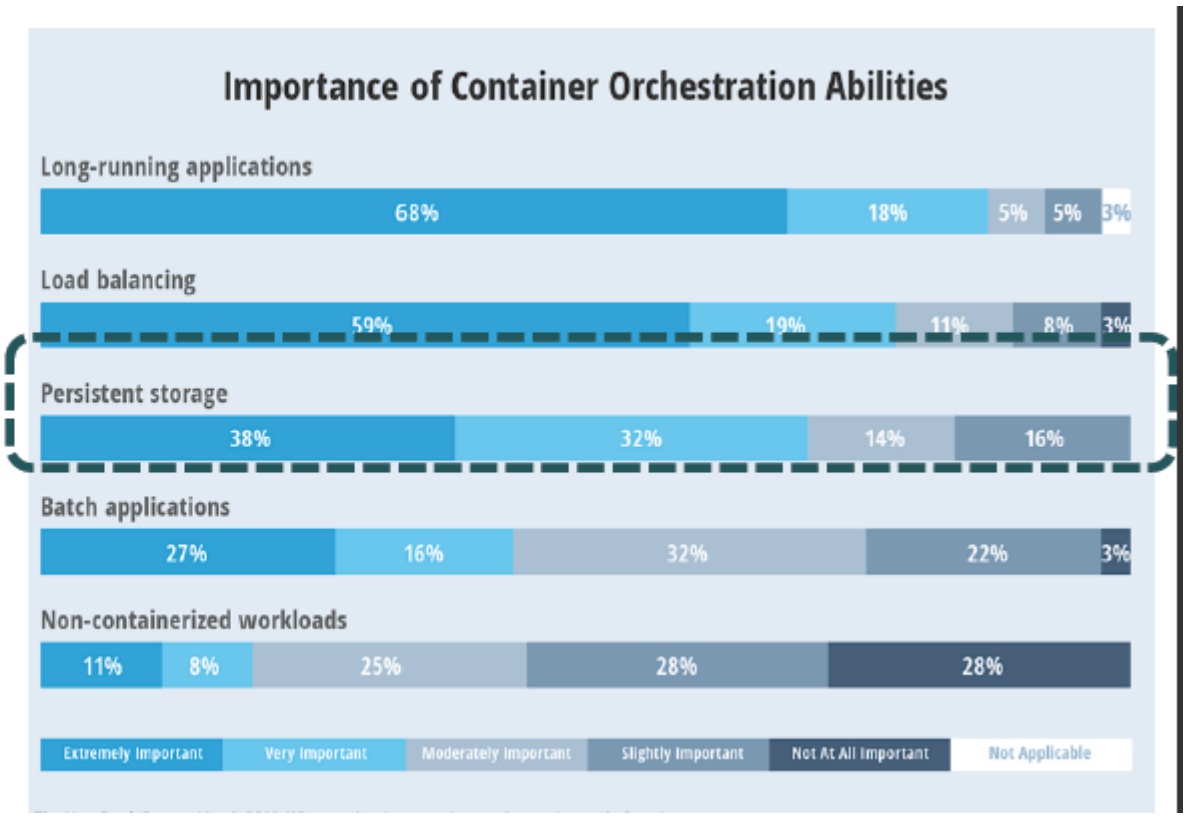
Scenario: Enabling Transformation to Microservices

The Docker Platform consists of multiple products/tools
– Docker Engine
– Docker Hub
– Docker Trusted Registry
– Docker Machine
– Docker Compose
– Docker for Windows/Mac
– Docker Datacenter

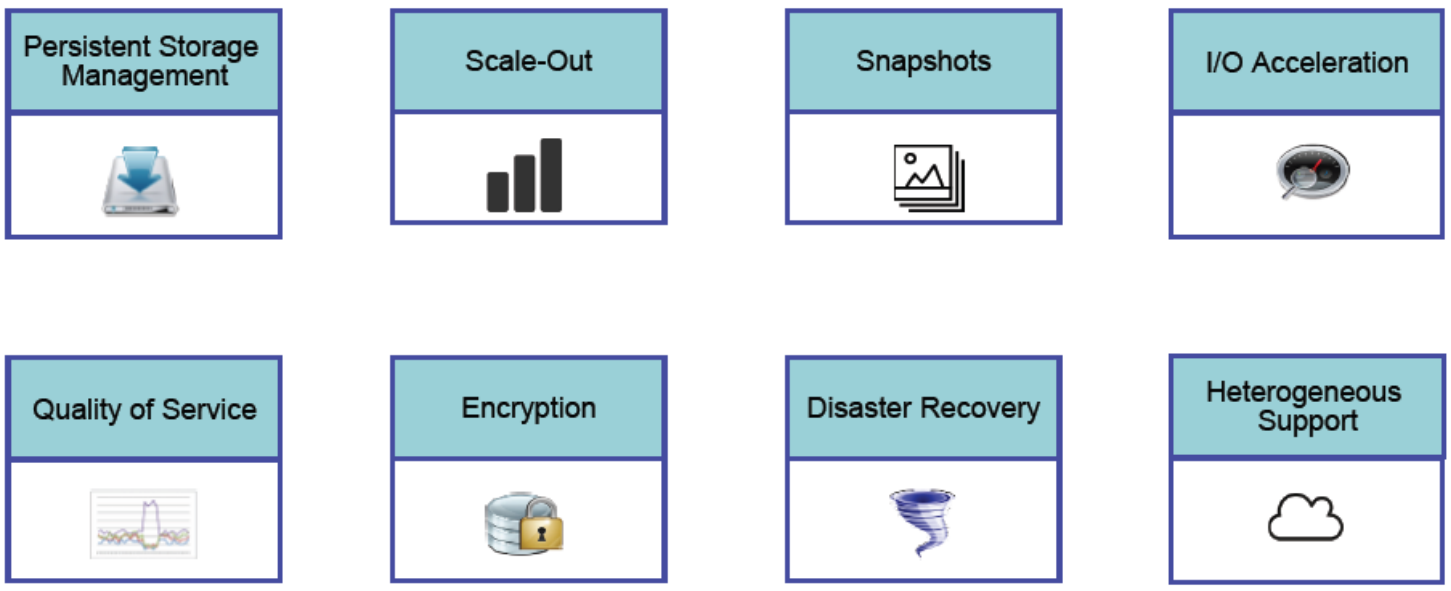
"**Stateful container** apps represent the next big **IT challenge**[1]"

"**Persistent storage** among top issues for container enterprise-readiness in **production**[2]"

"**Stateful Database** applications such as Redis, MySQL, MongoDB among **most pulled** images on Docker Hub[2]"



Stateful vs Stateless



Storage Services for Containers

Docker Storage Types

SNIA CSI | CLOUD STORA

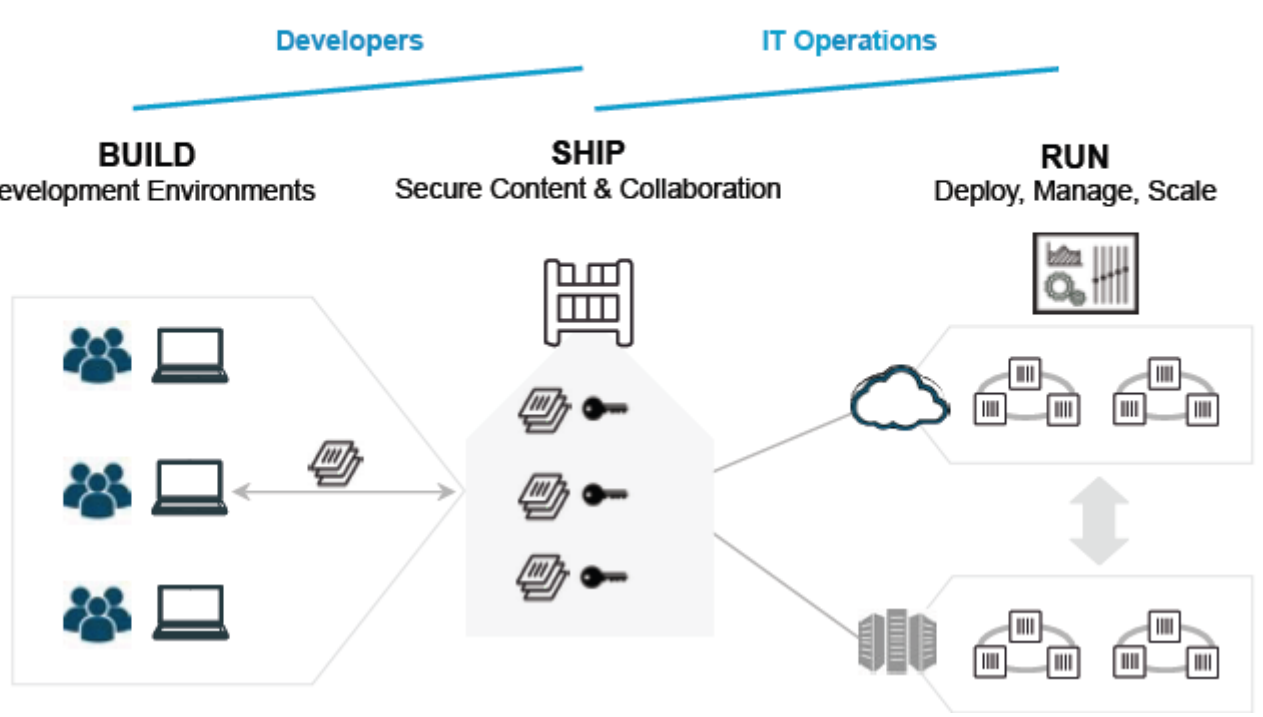
Registry — Cold storage of container images

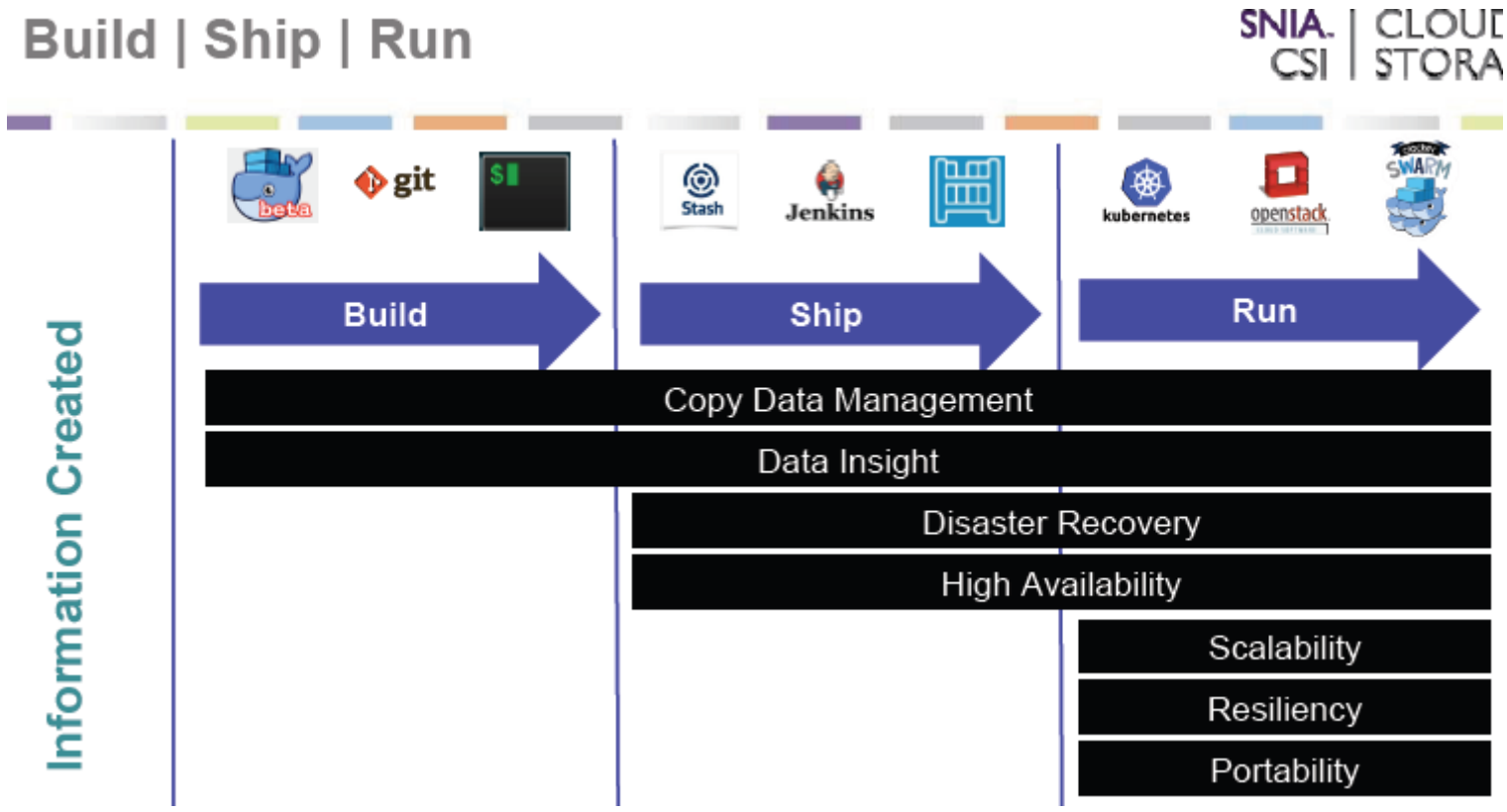
Graph — Active storage of running container images

Volume — Persistent block storage for data

Docker Datacenter CaaS workflow

SNIA CSI | C S





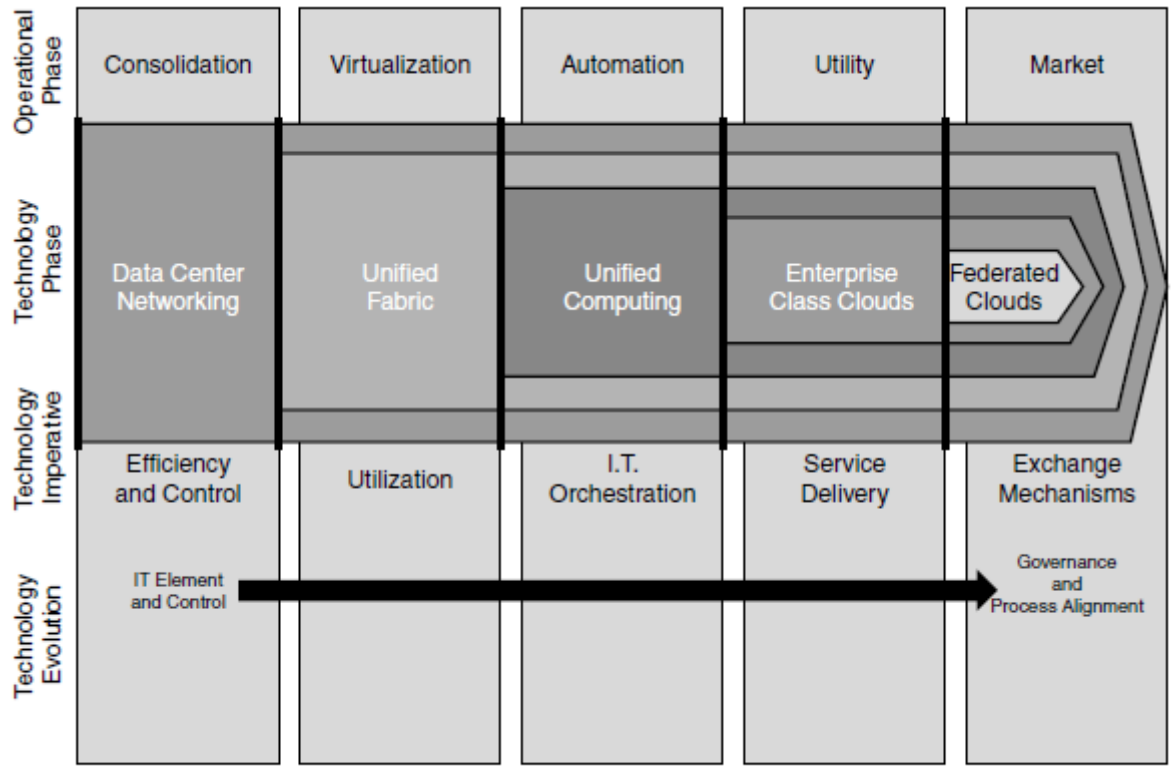


Figure 3-4 *Operational and Technological Evolution Stages of IT*

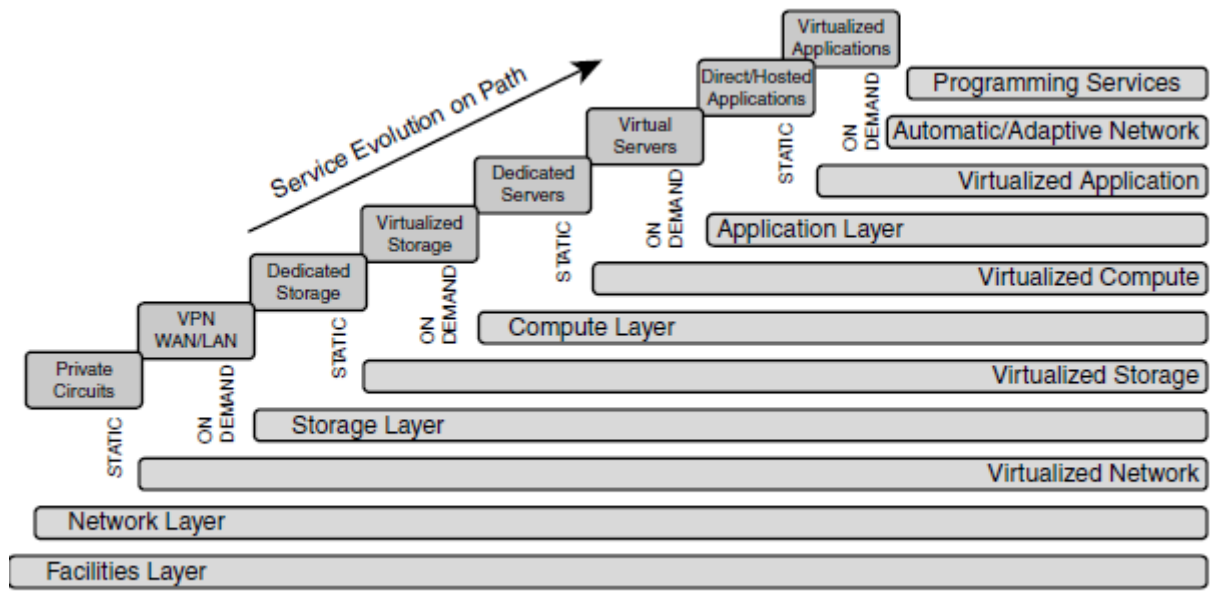


Figure 3-5 *IT Service Enablement Through Abstraction/Virtualization of IT Domains*

Introducing Layer 2 Multi-Pathing (L2MP): IETF Transparent Interconnection of Lots of Links (TRILL) is a new Layer 2 topology based capability. With the Nexus 7000 switch, Cisco already supports a prestandards version of TRILL called *FabricPath*, enabling customers to benefit from this technology

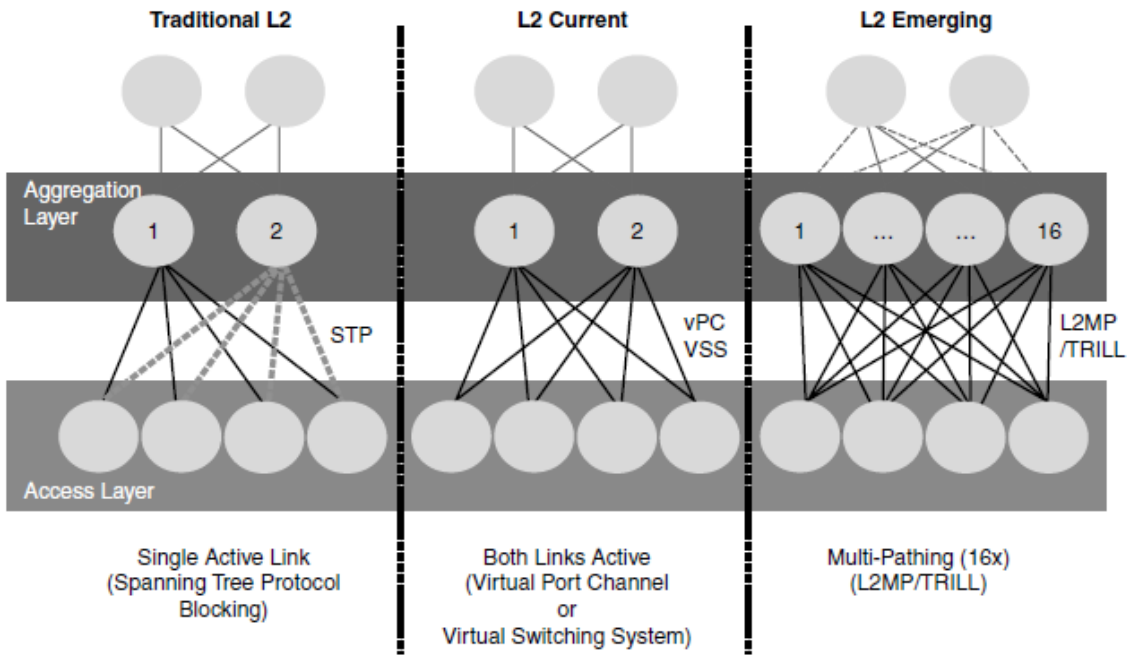


Figure 3-6 *Evolution of OSI Layer 2 in the Data Center*

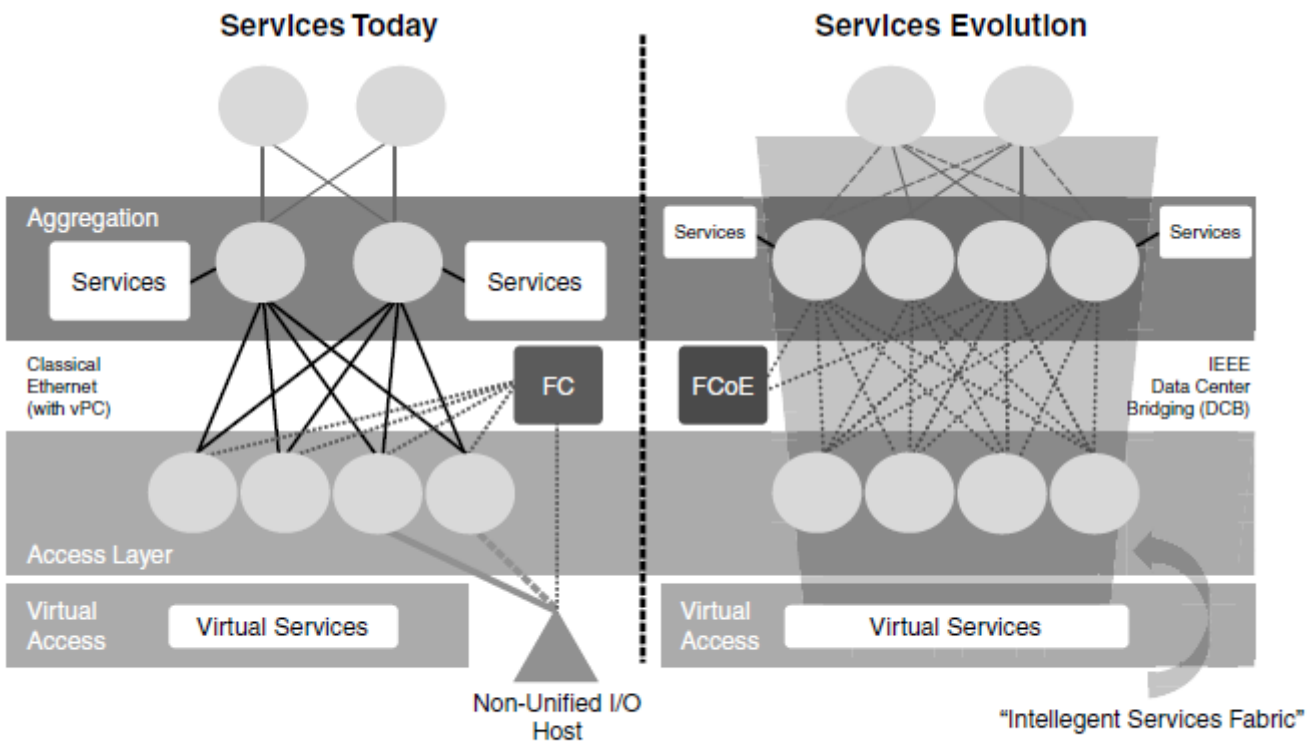


Figure 3-7 *Evolution of I/O Fabric and Service Deployment in the DC*

Table 3-2 *Features and Benefits of Data Center Bridging*

Feature	Benefit
Priority-based Flow Control (PFC) (IEEE 802.1 Qbb)	Provides the capability to manage a bursty, single-traffic source on a multiprotocol link
Enhanced Transmission Selection (ETS) (IEEE 802.1 Qaz)	Enables bandwidth management between traffic types for multiprotocol links
Congestion Notification (IEEE 802.1 Qau)	Addresses the problem of sustained congestion by moving corrective action to the network edge
Data Center Bridging Exchange (DCBX) Protocol	Allows autoexchange of Ethernet parameters between switches and endpoints

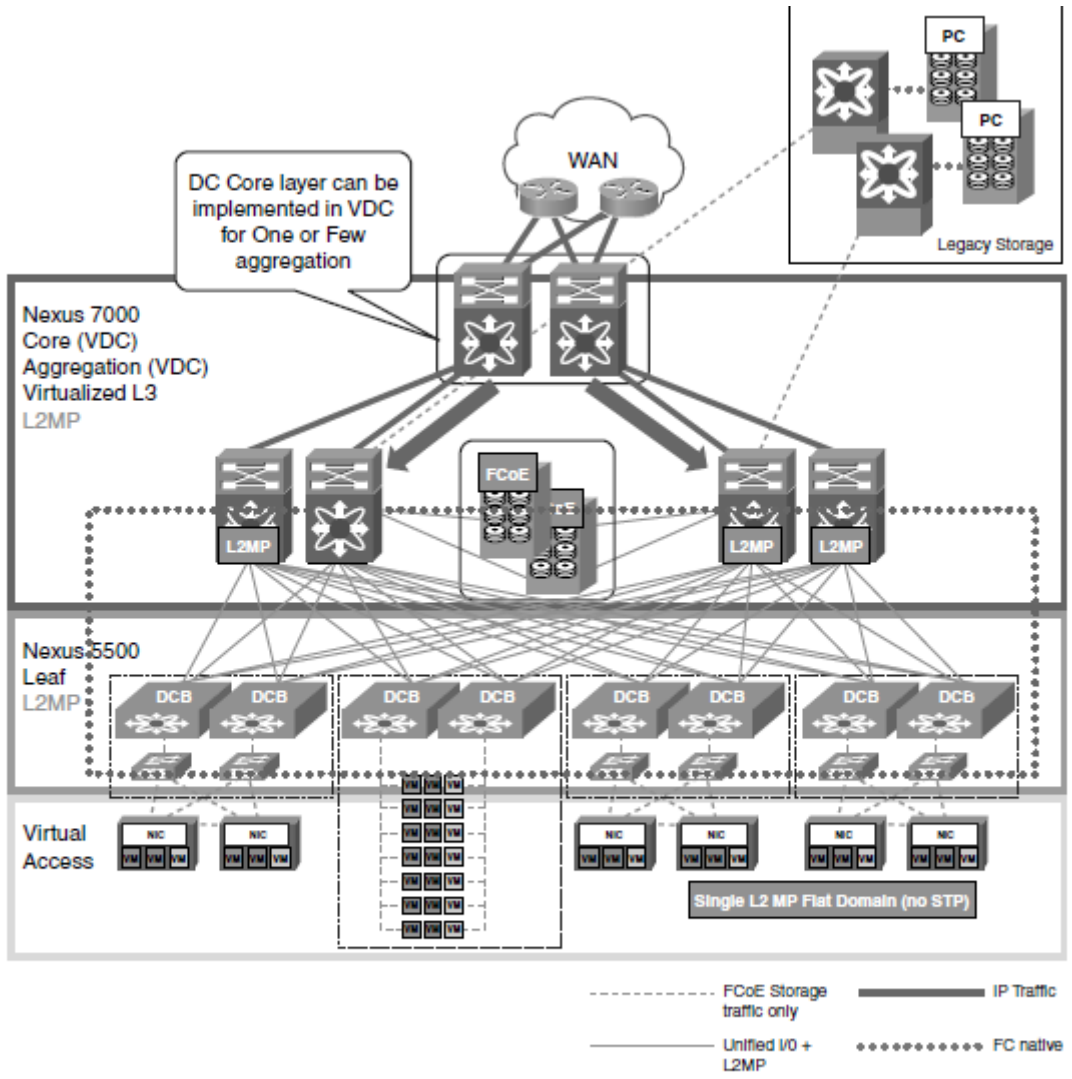


Figure 3-8 *Collapsing of the Vertical Hierarchy with Nexus 7000 Virtual Device Contexts (VDC)*

Multi-Protocol Label Switching (MPLS). Therefore, multitenancy in the DC is an evolution of a well-established paradigm, albeit with some additional technologies such as VLANs and Virtual Network Tags (VN-Tag) combined with virtualized network services (for example, session load balancers, firewalls, and IPS PEP instances).

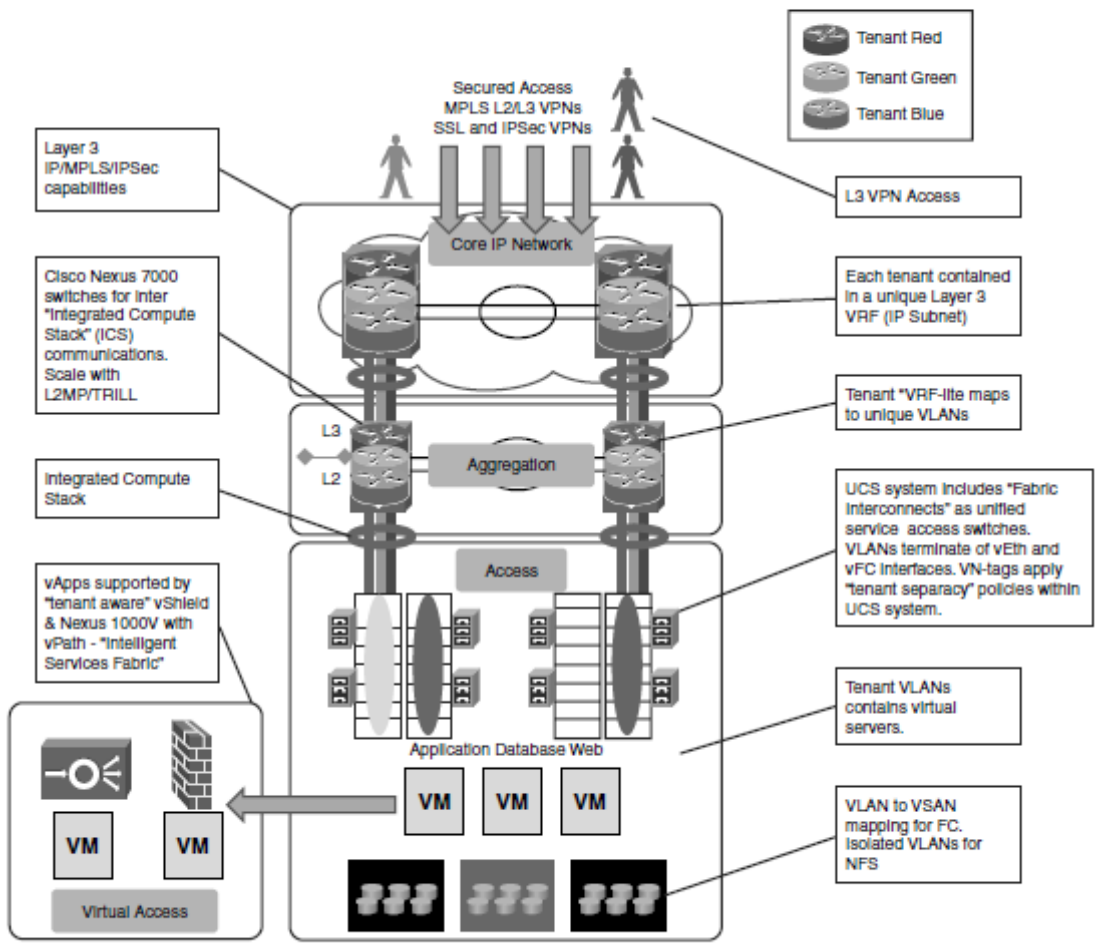


Figure 3-9 *End-to-End "Separacy"—Building the Multitenant Infrastructure*

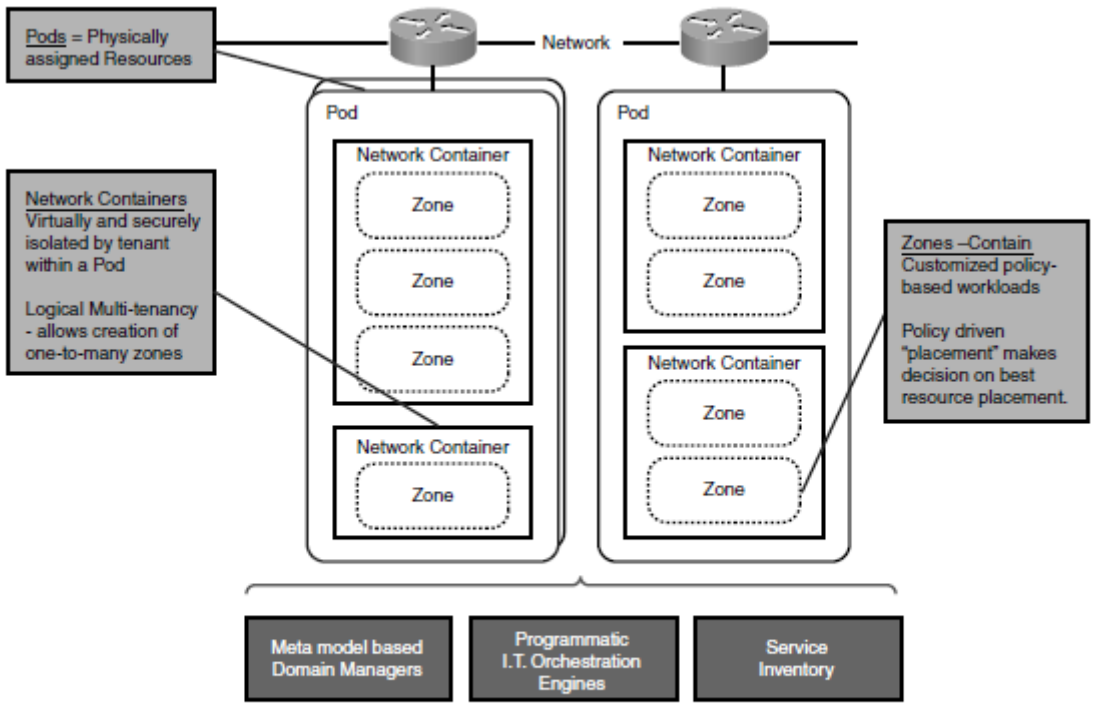


Figure 3-10 *Example of a Hierarchical Architecture Incorporating Multitenancy, Multitier, and Multizoning Attributes for an IaaS Platform (Source: Cisco Systems VMDC 2.0 Solution)*

This structured approach requires a predefined infrastructure modeling of the capacity and capability of service elements and their proximity and adjacency to each other. A predefined service chain, known more colloquially as a network container, can therefore be activated on the infrastructure as a known unit of consumption. A service chain is a group of technical topology building blocks, as illustrated in Figure 3-11.

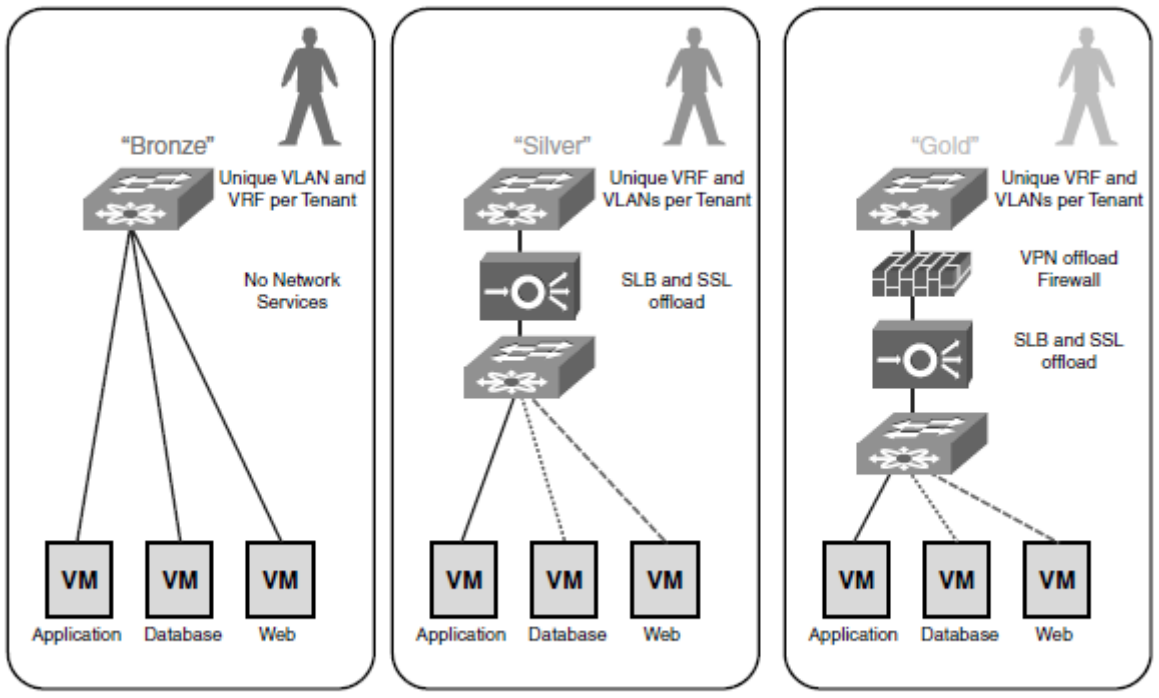


Figure 3-11 *Network Containers for Virtual Private Cloud Deployment*

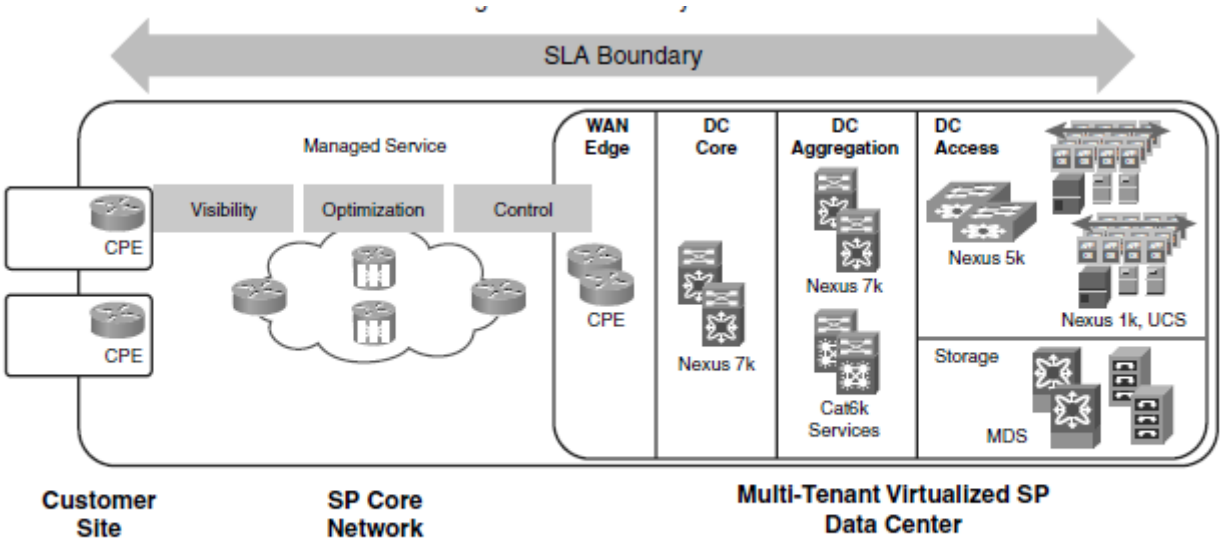


Figure 3-12 *Expanding the SLA Boundary*

An IaaS assurance deployment requires a real-time and extensible data model that can support the following:

- Normalized object representation of multiple types of devices and domain managers, their components, and configuration
- Flexible enough to represent networking equipment, operating systems, data center environmental equipment, standalone and chassis servers, and domain managers such as vSphere, vCloud Director, and Cisco UCS
- Able to manage multiple overlapping relationships among and between managed resources
- Peer relationships, such as common membership in groups
- Parent-child relationships, such as the relationship between a UCS chassis and blade
- Mobile dependency relationships, such as the relationship between a VM and its current host system
- Cross-silo discovered relationships, such as the relationship between a virtual host and a logical unit number (LUN) that represents network attached logical storage volume
- Linkages between managed objects and management data streams, such as event database and performance metrics
- Security boundaries between sets of managed objects and subsets of users to enable use in multitenant environments
- Developer-extensible to allow common capabilities to be developed for all customers
- Field-extensible to enable services teams and customers to meet uncommon or unique requirements

■ Fixed dependency relationships, such as the relationship between a process and an operating system

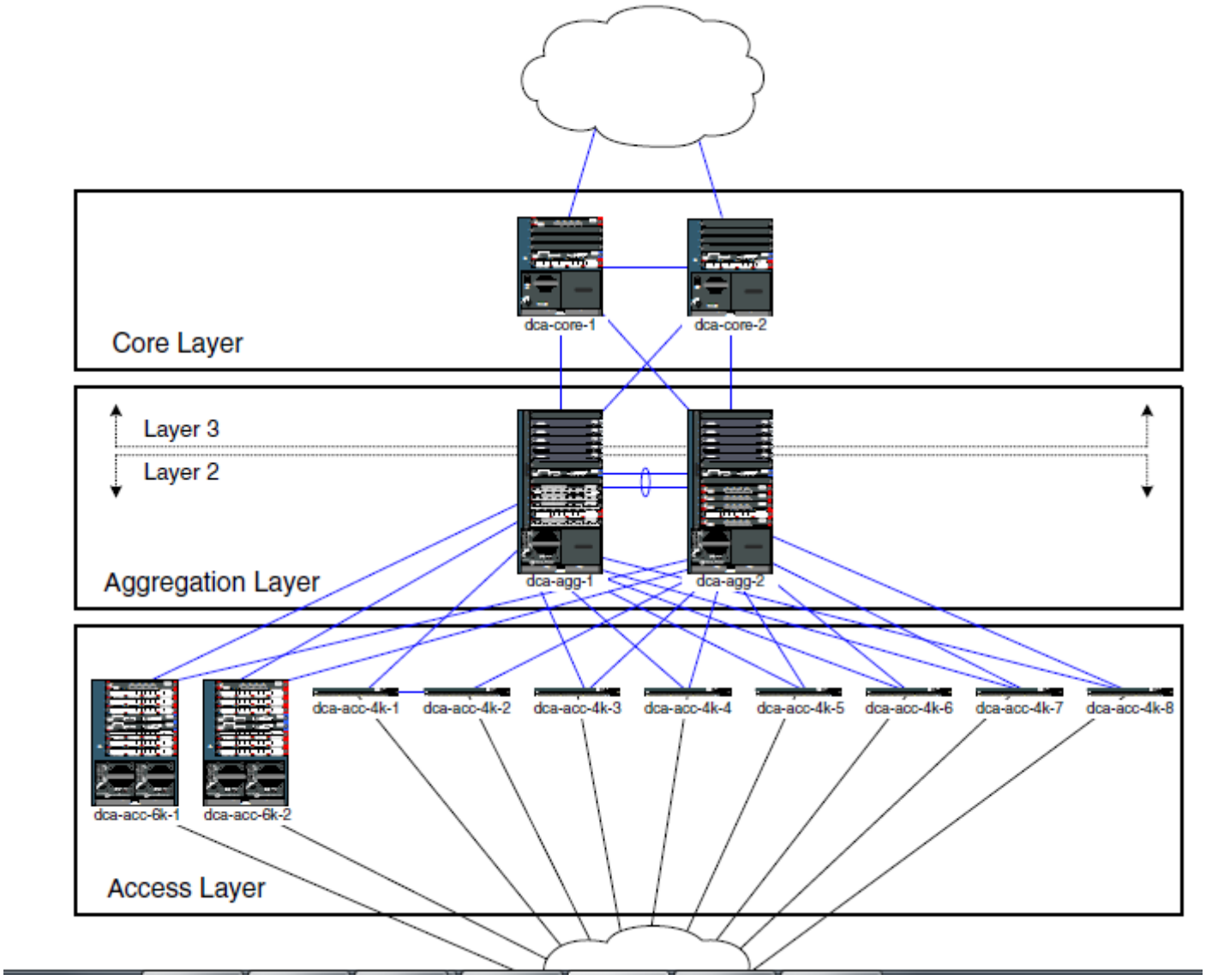
Layer 2-3 Infrastructure Overview

The DCAP 4.0 test topology consists of two separate data centers, DC-A and DC-B. Each data center has its own LAN, SAN and storage components. The tests performed with regards to LAN Layer 2-3 Infrastructure verification were executed against the LAN topology in DC-A. Figure 1-1 shows this portion of the test topology. It is divided into three distinct, logical layers called the Core, Aggregation, and Access Layers offering the Layer 2-3 services listed in Table 1-1.

Table 1-1 Logical Layer Services

Logical Layer	Services
Core	OSPF, CEF
Aggregation	Default Gateway Redundancy (HSRP), OSPF, Rapid PVST+ Spanning-Tree, UDLD, LACP, 802.1q Trunking
Access	Rapid PVST+ Spanning-Tree, 802.1q Trunking

Figure 1-1 Cisco DCAP 4.0 DCa Topology



Layer 4-7 Services Overview

There are several Layer 4-7 services that are employed as part of the DCAP 4.0 test topology. They include load balancing, firewalling, SSL offloading, intrusion detection and prevention, global site load balancing and application acceleration.Table 1-2 shows these Layer 4-7 services and the Cisco product line that DCAP uses to provide them.

Table 1-2 Layer 4-7 Solutions Used in DCAP 4.0

Layer 4-7 Service	DCAP Solution
Load balancing	ACE, CSM
Firewall	FWSM
SSL offloading	SSLM, ACE
Global site load balancing	GSS
Application optimization	WAAS
Intrusion prevention/detection	IDSM

Of the Cisco products used in this list, several are service modules:

- ACE – Application Control Engine
- CSM – Content Switching Module
- FWSM – Firewall Services Module
- SSLM – SSL Module
- IDSM – Intrusion Detection Services Module

Data center services-> Data center Engineering services -> Products catalogue

http://clusterdesign.org/fat-trees/

Data center services-> Data center IT services -> Products catalogue

Data center services-> Data center security services-> Products catalogue

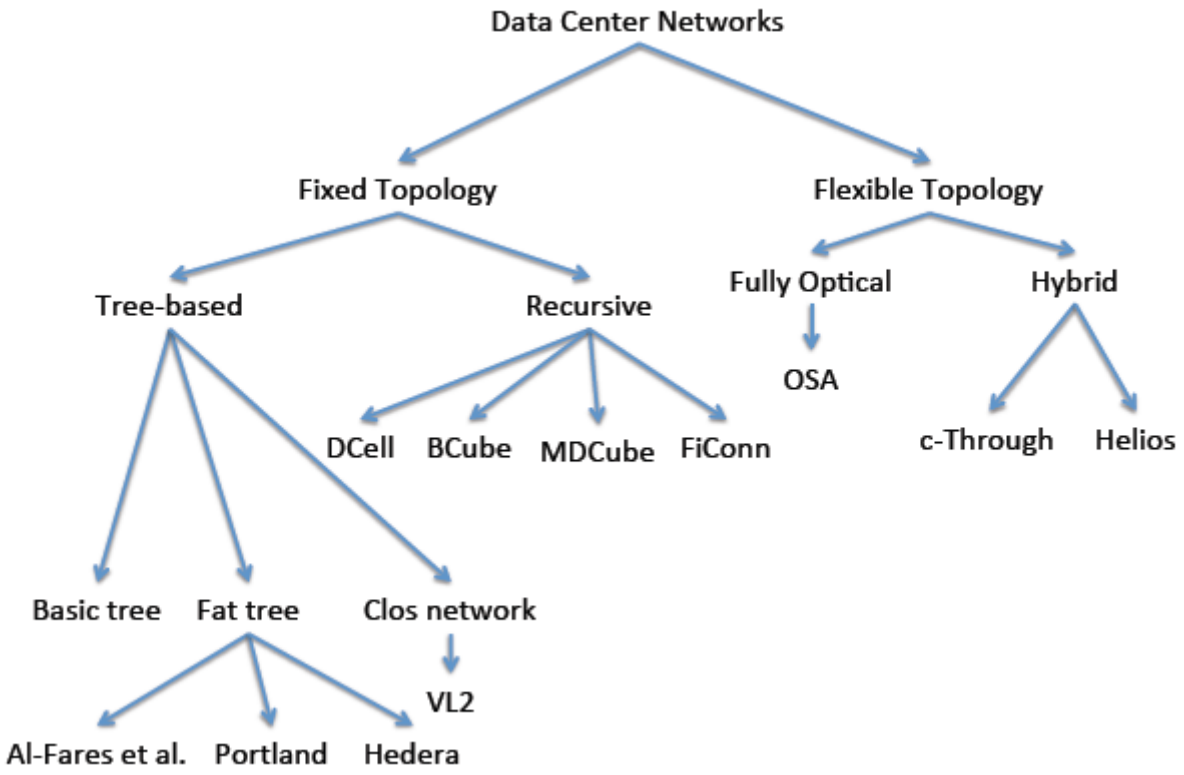
Data center services-> Data center QA services-> Products catalogue

Data center services-> Data center cloud services-> Products catalogue

Data center services-> Data center compliances services-> Products catalogue

Data center services-> Data center business services-> Products catalogue

Data center services-> Networking services -> Products catalogue



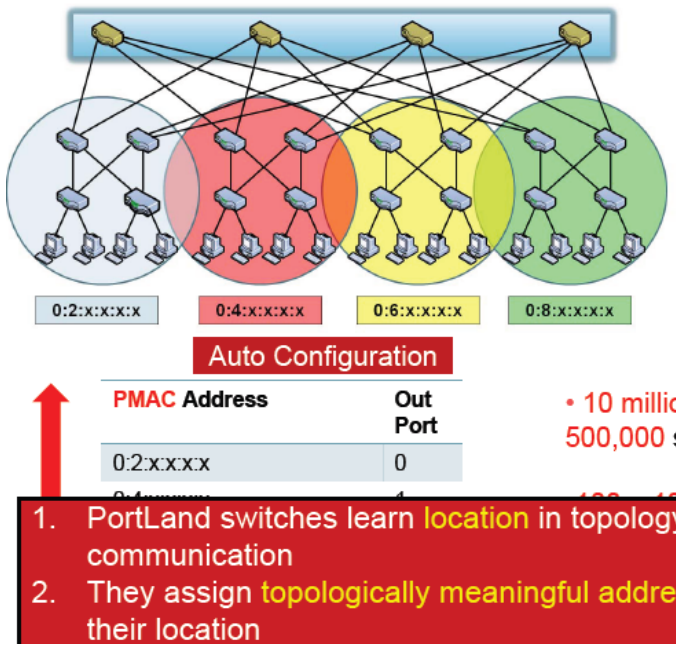
Taxonomy of Data Center Topologies

Source: A Survey of Data Center Network Architectures.pdf

PortLand In A Nutshell

- PortLand is a single logical layer 2 data center network fabric that scales to millions of endpoints
- PortLand internally separates host identity from host location
  - Uses IP address as host identifier
  - Introduces "Pseudo MAC" (PMAC) addresses internally to encode endpoint location
- PortLand runs on commodity switch hardware with unmodified hosts

PortLand: Plug and Play + Small Switch State



+ Pair-wise communication

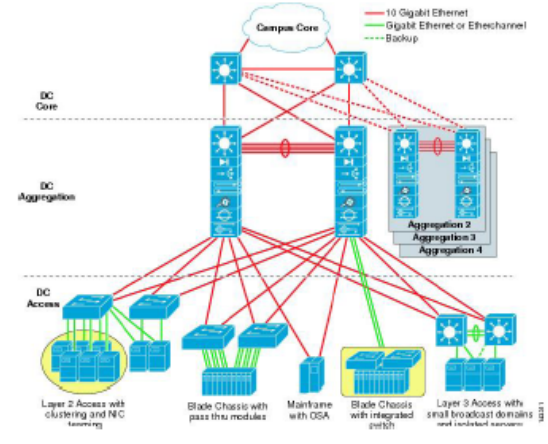
- 10 million virtual endpoints in 500,000 servers in data center

1. PortLand switches learn location in topology using pair-wise communication
2. They assign topologically meaningful addresses to hosts using their location

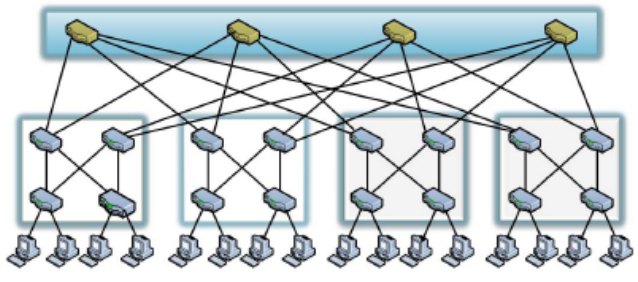
PortLand: Main Assumption

Hierarchical structure of data center networks:

They are multi-level, multi-rooted trees



Cisco Recommended Configuration

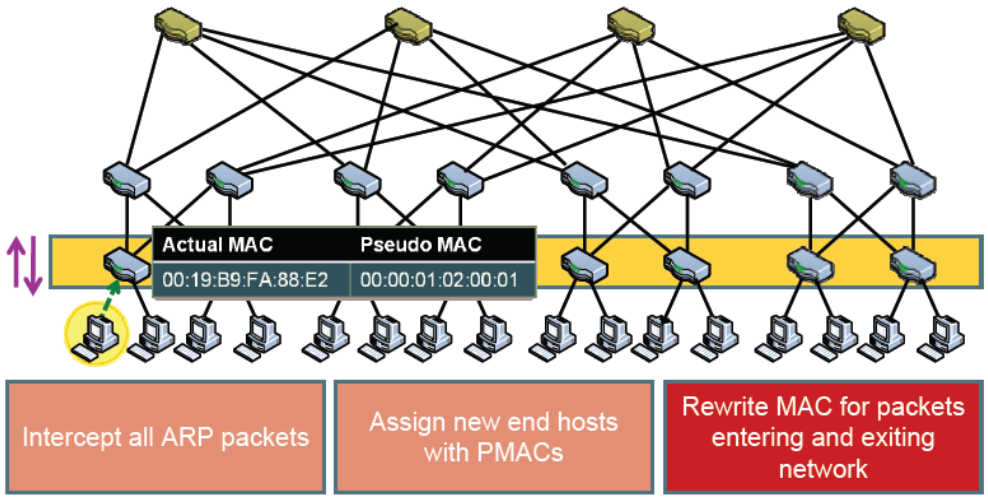


Fat Tree

PROXY-BASED ARP

- When an edge switch sees a new AMAC, it assigns a PMAC to the host
- It then communicates the PMAC to IP mapping to the fabric manager.
- The fabric manager servers as a proxy-ARP agent, and answers ARP queries

PortLand: Name Resolution



PROVABLY LOOP-FREE FORWARDING

- Switches populate their forwarding tables after establishing local positions
- Core switches forward according to pod numbers
- Aggregation switches forward packets destined to the same pod to edge switches, to other pods to core switches
- Edge switches forward packets to the corresponding hosts

FAULT TOLERANT ROUTING

- LDP exchanges serve as keepalive
- A switch reports a dead link to the fabric manager (FM)
- The FM updates its faulty link matrix, and informs affected switches the failure
- Affected switches reconfigure their forwarding tables to bypass the failed link
- → No broadcasting of the failure

Network Design Goals

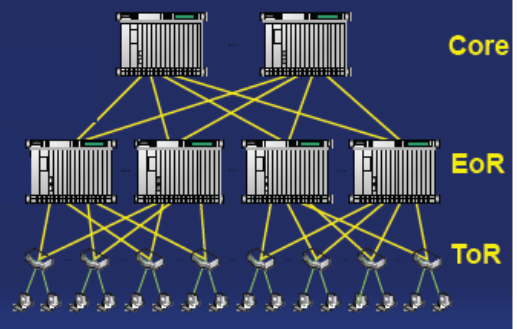
- Scalable interconnection bandwidth
  - Full bisection bandwidth between all pairs of hosts
    Aggregate bandwidth = # hosts × host NIC capacity
- Economies of scale
  - Price/port constant with number of hosts
  - Must leverage commodity merchant silicon
- Single network fabric
  - Support Ethernet and IP without end host modification
- Management
  - Modular design
  - Avoid actively managing 100's-1000's network elemer

Scale Out Networking

- Advances toward *scale out* computing and storage
  - Aggregate computing and storage grows linearly with t number of commodity processors and disks
  - Small matter of software to enable functionality
  - Alternative is *scale up* where weaker processors and smaller disks are *replaced* with more powerful parts
- Today, no technology for scale out networking
  - Modules to expand number of ports or aggr BW
  - No management of individual switches, VLANs, subnet

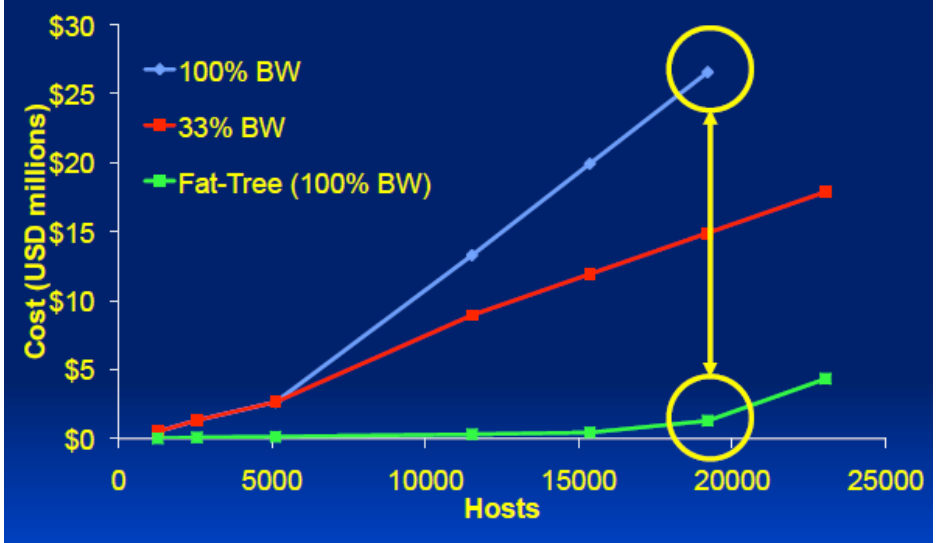
Current Data Center Topologies

- Edge hosts connect to 1G Top of Rack (ToR) switch
- ToR switches connect to 10G End of Row (EoR) switches
- Large clusters: EoR switches to 10G core switches
  - Oversubscription of 2.5:1 to 8:1 typical in guidelines
- No story for what happens as we move to 10G to the edge



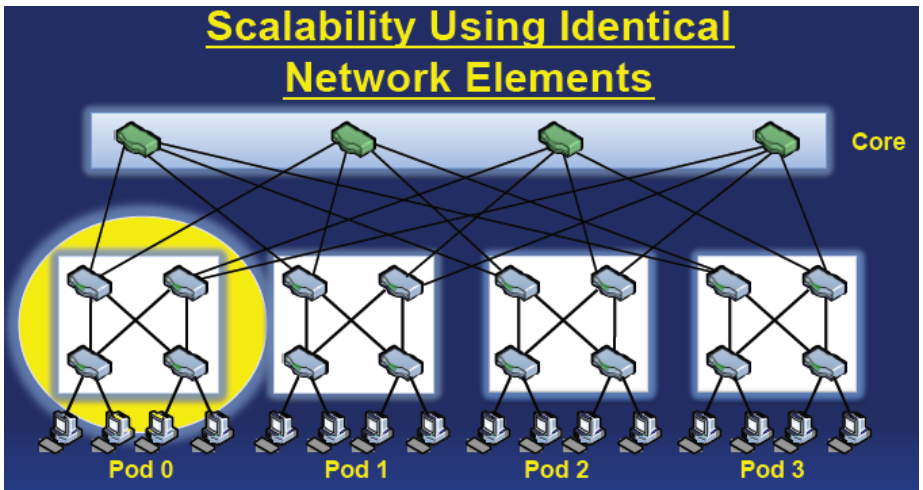
➡ Key challenges: performance, cost, routing, energy, cabling

Cost of Data Center Networks



- Factor of 10+ price difference between traditional approach and proposed architecture

Scalability Using Identical Network Elements

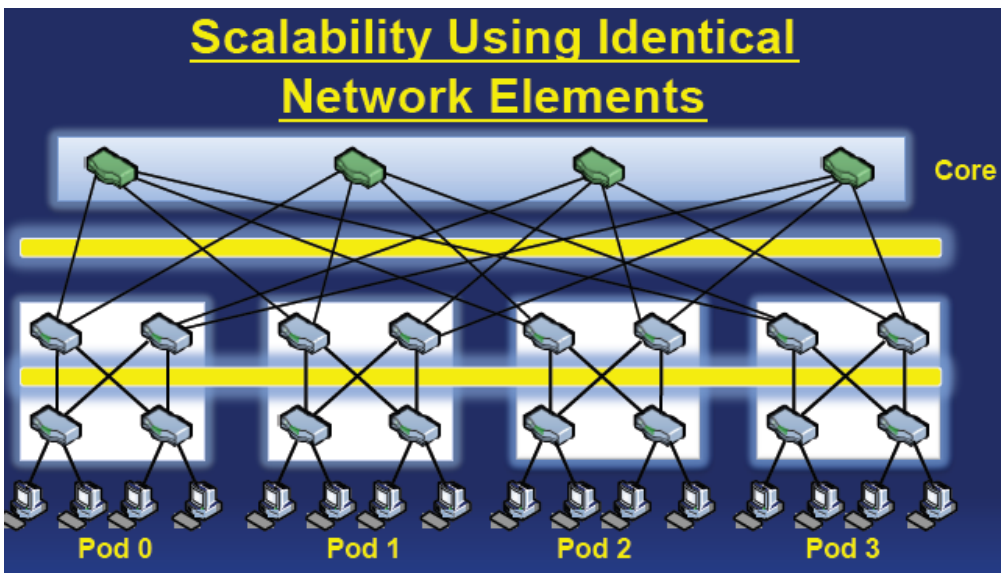


- Support 16 hosts organized into 4 pods
  - Each pod is a 2-ary 2-tree
  - Full bandwidth among hosts directly connected to pod

Fat tree built from 4-port switches

- *$(5k^2/4)$ k*-port switches support *$k^3/4$* hosts
  - 48-port switches: 27,648 hosts using 2,880 switches
- Critically, approach scales to 10 GigE **at the edge**

Scalability Using Identical Network Elements



- Full bisection bandwidth at each level of fat tree
  - Rearrangeably Nonblocking
  - Entire fat-tree is a 2-ary 3-tree

Our Work

- Switch Architecture [SIGCOMM 08]
- Cabling, Merchant Silicon [Hot Interconnects 09]
- Virtualization, Layer 2, Management [SIGCOMM 09]
- Routing/Forwarding [ongoing]
- Energy, Optics [ongoing]

- Take advantage of regularity of fat tree structure to simplify protocol design and improve performance

PortLand Goals

- Single Layer 2 fabric to 100k ports, 1M end hosts
  - VM migration while maintaining IP address, external connectivity and session state
- Toward zero configuration at deployment
  - No subnet, IP address, wiring information ,etc.
- No forwarding loops
- Rapid and efficient failure detection
- Native multicast support
- First class support for multipath forwarding

PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric

System Comparison

System	Topology	Forwarding		Routing	ARP	Loops	Multicast
		Switch State	Addressing				
TRILL	General	O(# of global hosts)	Flat: MAC in MAC encap	Switch broadcast	Switch maps MAC address to remote switch	TRILL header with TTL	ISIS, extensions based on MOSPF
SEATTLE	General	O(# of global hosts)	Flat	Switch broadcast	One-hop DHT	Unicast loops possible	New construct: groups
PortLand	Multi-rooted tree	O(# of ports)	Hierarchical	LDP: Fabric Manager for faults	Fabric Manager	Provably loop free; no extra encap	Broadcast free routing; Multi-rooted spanning trees

Design Overview

- Separate node location from node identifier
  - Host IP address: node identifier
  - *Pseudo MAC* (PMAC): node location
- *Fabric Manager*
  - Maintains IP→PMAC mapping
  - Facilitates fault tolerance
- PMAC sufficient to perform positional *forwarding*
- Location Discovery Protocol (LDP) for decentralized, zero-configuration *routing* and *forwarding*

Layer 2 versus Layer 3 Data Center Fabrics

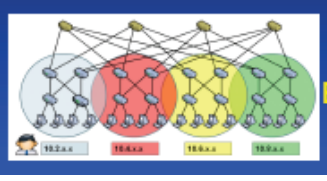
Technique	Plug and play	Scalability	Small Switch State	Seamless VM Migration
Layer 2: Flat MAC	+	-	-	+
Layer 3: IP Addrs	-	~	+	-

Flooding

Host MAC Address	Out Port
9a:57:10:ab:6e	1

No change

Location-based addresses mandate manual configuration



LSR – Broadcast based

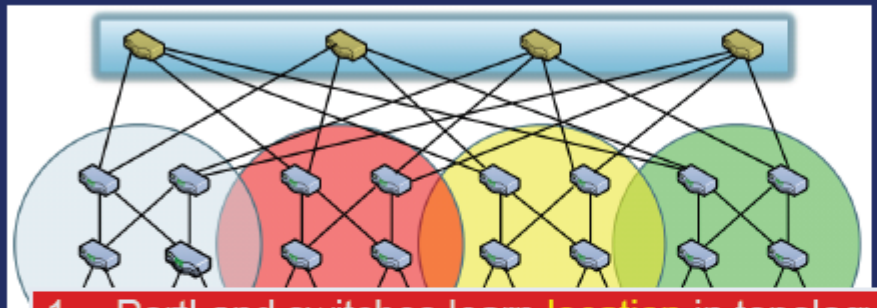
Host IP Address	Out Port
10.2.x.x	0
10.4.x.x	1

IP endpoint changes

Cost Consideration: Flat vs. Hierarchical Addresses

- Commodity switches today have ~640 KB of low latency, power hungry, expensive on chip memory
  - Stores 32 – 64 K flow entries
- 10 million virtual endpoints in 500k servers in data center
- Flat addresses → 10 million address mappings → ~100 MB on chip memory → ~150 times the memory size that can be put on chip today
- Location based addresses → 100 – 1000 address mappings→ ~10 KB of memory → easily accommodated in switches today

PortLand: Plug and Play + Small Switch State



+ Pair-wise communication

1. PortLand switches learn location in topology using pair-wise communication
2. They assign topologically meaningful addresses to hosts using their location

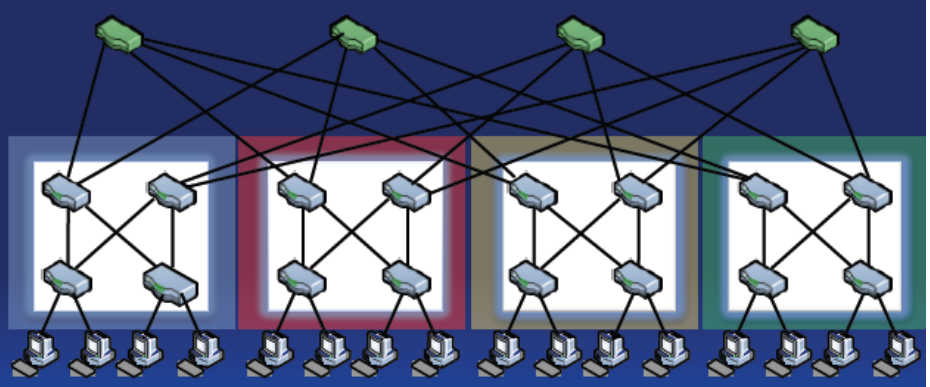
PMAC Address	Out Port
0:2:x:x:x:x	0
0:4:x:x:x:x	1
0:6:x:x:x:x	2
0:8:x:x:x:x	3

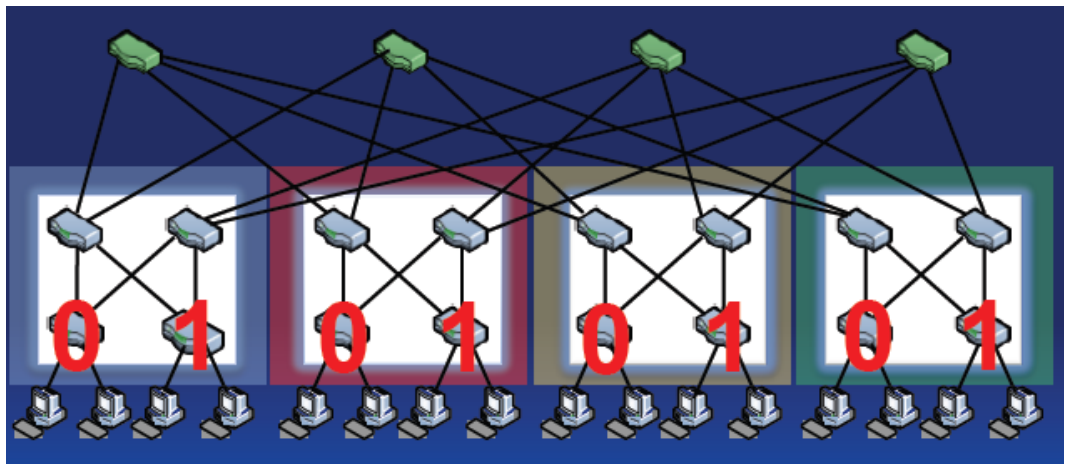
- 10 million virtual endpoints in 500,000 servers in data center
- 100 – 1000 address mappings→ ~10 KB of memory → easily accommodated in switches today

PortLand: Layer 2 Scalability Challenges

Challenge	State Of Art
Address Resolution	Broadcast based
Routing	Broadcast based
Forwarding	Large switch state

Layering Hierarchy On A Multi-Rooted Tree

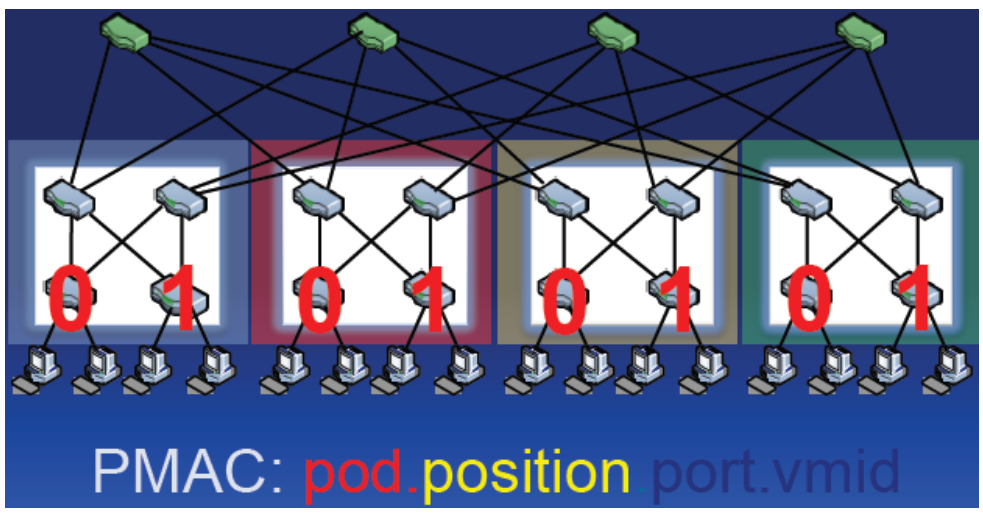




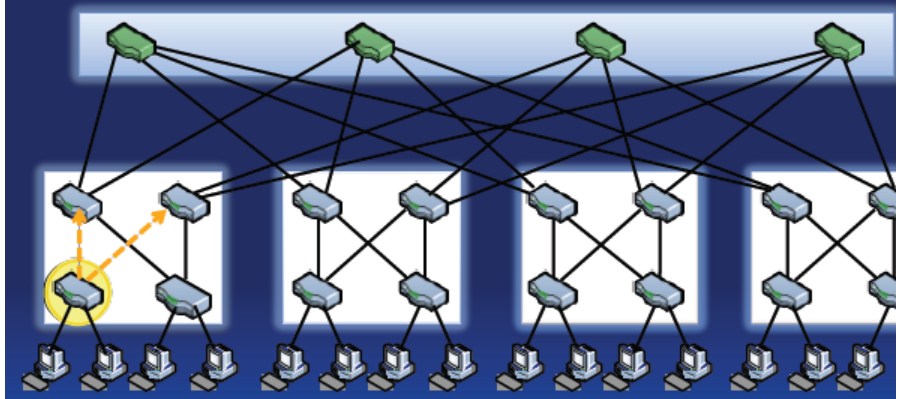
PortLand: Location Discovery Protocol

- Location Discovery Messages (LDMs) exchanged between neighboring switches
- Switches self-discover location on boot up

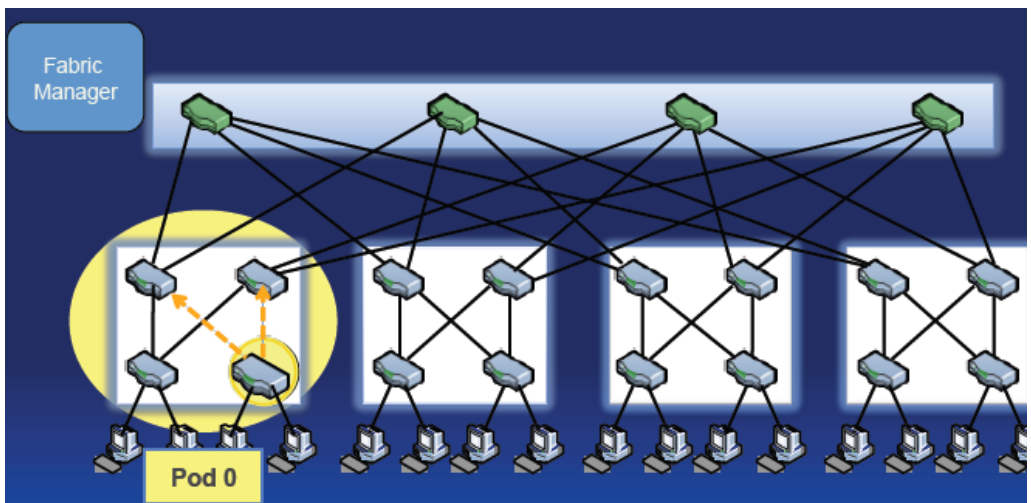
Location characteristic	Technique
1) Tree level / Role	Based on neighbor identity
2) Pod number	Aggregation and edge switches agree on pod number
3) Position number	Aggregation switches help edge switches choose unique position number



PortLand: Location Discovery Protocol

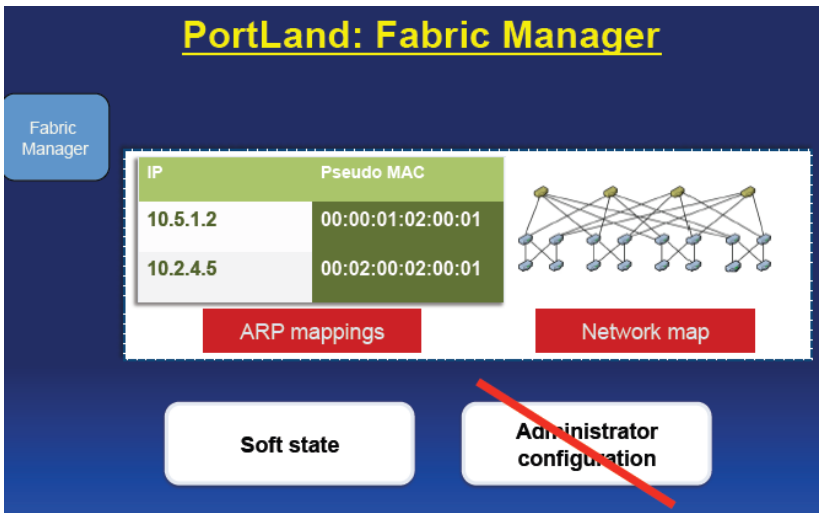


Switch Identifier	Pod Number	Position	Tree Level
A0:B1:FD:56:32:01	??	??	??

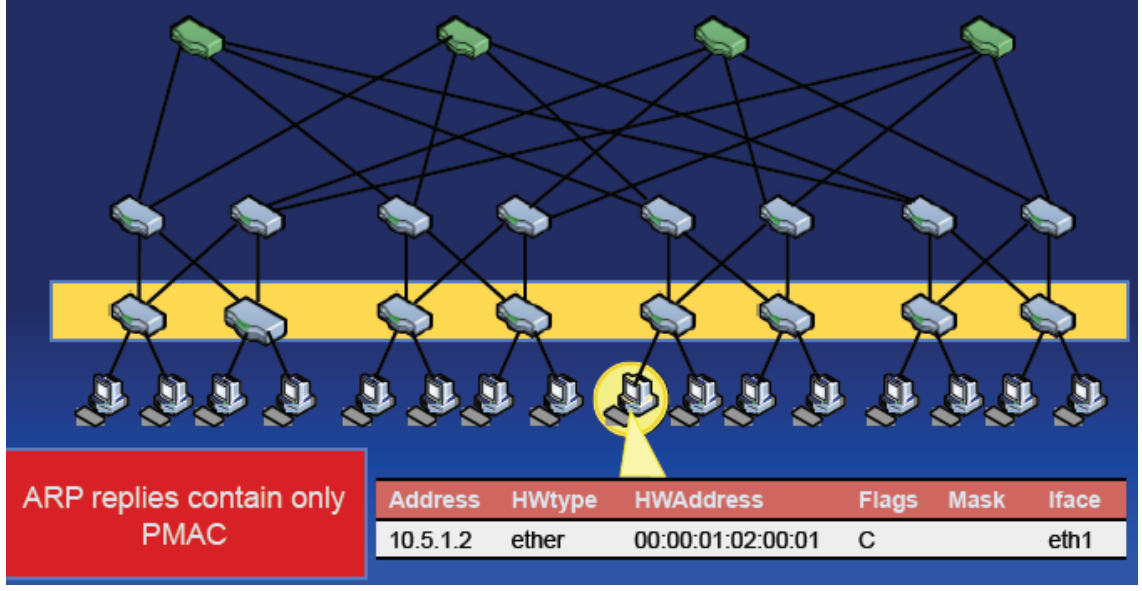


Switch Identifier	Pod Number	Position	Tree Level
D0:B1:AD:56:32:01	0	0	0

PortLand: Fabric Manager



PortLand: Name Resolution



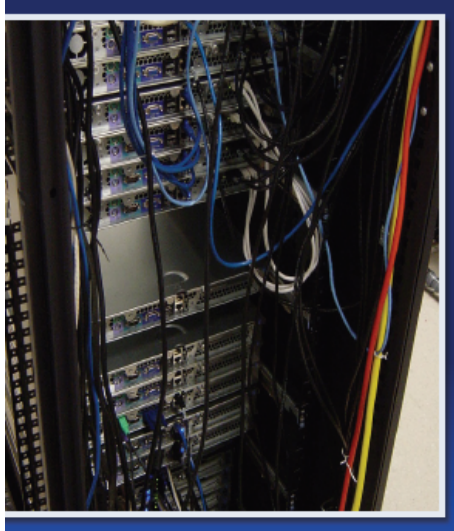
Design: Provably Loop Free Forwarding

- Up-Down semantics [AutoNet]
- Constraint: A switch must never forward a packet along an upward facing port when the ingress port for that packet is also an upward facing port
- Traffic can never change its direction more than once

Broadcast Free Routing Protocol

- No Link State Routing
- Liveness Monitoring by LDP exchanges
- Switches convey fault information to Fabric Manager
- Fabric Manager informs affected switches of failure

PortLand Prototype



- 20 OpenFlow NetFPGA switches
- TCAM + SRAM for flow entries
- Software MAC rewriting
- 3 tiered fat-tree
- 16 end hosts
- Implementations underway for HP, Broadcom, and Fulcrum switches

PortLand: Evaluation

Measurements	Configuration	Results
Network convergence time	Keepalive frequency = 10 ms Fault detection time = 50 ms	65 ms
TCP convergence time	RTO_{min} = 200ms	~200 ms
Multicast convergence time		110ms
TCP convergence with VM migration	RTO_{min} = 200ms	~200 ms – 600 ms
Control traffic to fabric manager	27,000+ hosts, 100 ARPs / sec per host	400 Mbps
CPU requirements of fabric manager	27,000+ hosts, 100 ARPs / sec per host	70 CPU cores

Related Work

- Floodless in SEATTLE
  - Lookup scheme to enable layer 2 routing based on DHT
- DCell
  - Alternative topology for data center interconnect
  - Fewer modifications to switch, but end host modifications required
  - Topology is not rearrangeably nonblocking
- Rbridges/TRILL
  - Layer 2 routing protocol with MAC in MAC encapsulation

Related Work

- Much work in interconnection networks comes from Massively Parallel Processing (MPP) world
  - Many supercomputers are organized as fat-trees
  - Thinking Machines CM-5, SGI Altix, etc.
- Specialized networks targeting HPC clusters deploy fat-tree topologies
  - InfiniBand, Myrinet, Quadrics
  - Most use source routing

Host Centric vs. Network Centric Approach

VL2

- Network architecture that scales to support huge data centers
- Layer 3 routing fabric used to implement a virtual layer 2
- Scale Layer 2 via end host modifications
  - Unmodified switch hardware and software
  - End hosts modified to perform enhanced resolution to assist routing and forwarding

PortLand

- Modify network fabric to
  - Work with arbitrary operating systems and virtual machine monitors
  - Maintain the boundary between network and end-host administration
- Scale Layer 2 via network modifications
  - Unmodified switch hardware and end hosts

Conclusions

Data Center Network Requirements

- Cost and absolute performance
- Packaging
- Energy/heat
- Fault tolerance

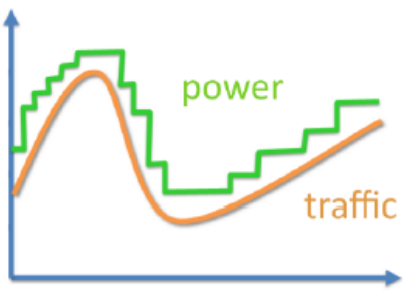
Fat-tree as basis for delivering on above challenges

- Regularity of the structure allows simplifying assumptions to address challenges above
- Fat-tree built from 48-port switches support 27k hosts
- Holds promise for cost, performance, energy

GOAL: ENERGY PROPORTION NETWORKING

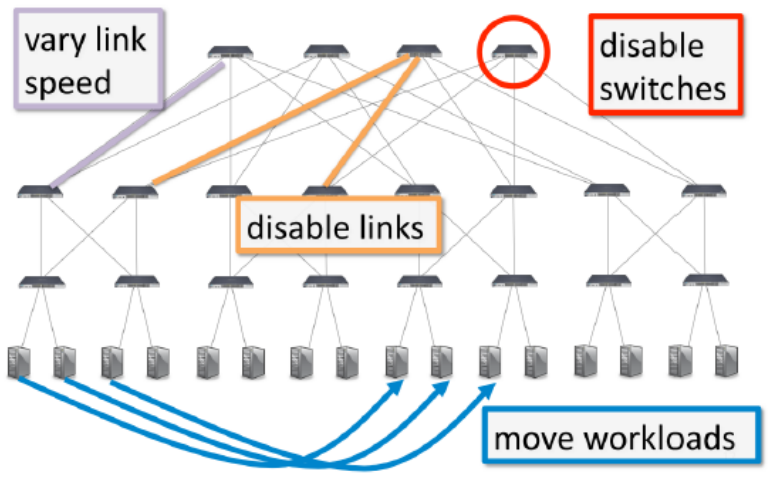
End goal:

Create an
energy-proportional data center **network**
from non-proportional components.



APPROACH: TURN OFF UNNEEDED LINKS AND SWITCHES CAREFULLY AND AT SCALE

Today's Network Power Knobs

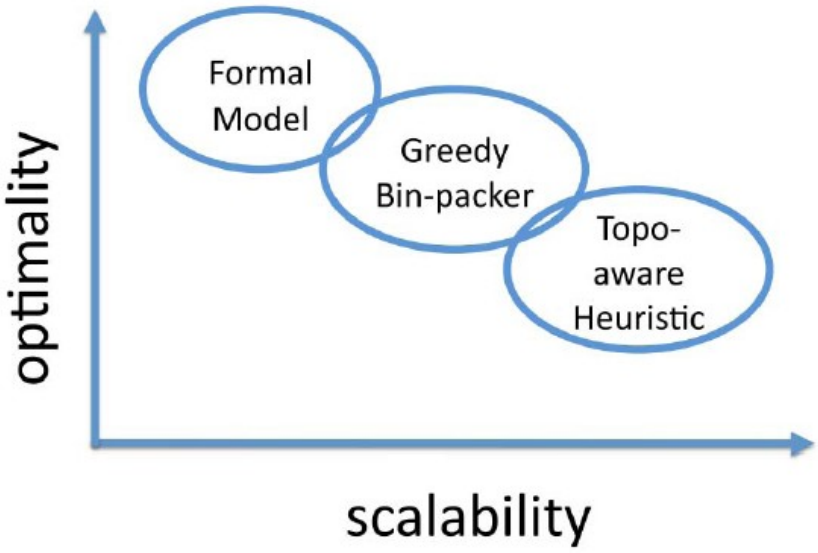


ELASTIC TREE ARCHITECTURE

Optimize for
Power Efficiency

Later, balance:
+ Fault Tolerance
+ Utilization Bounds

THREE OPTIMIZERS



FORMAL MODEL: MCF

Variables

Type	Description
Real	Amount of each flow along each link
Boolean	Switch power state
Boolean	Link power state

Optimization Goal

minimize Σ (link + switch power)

Constraints

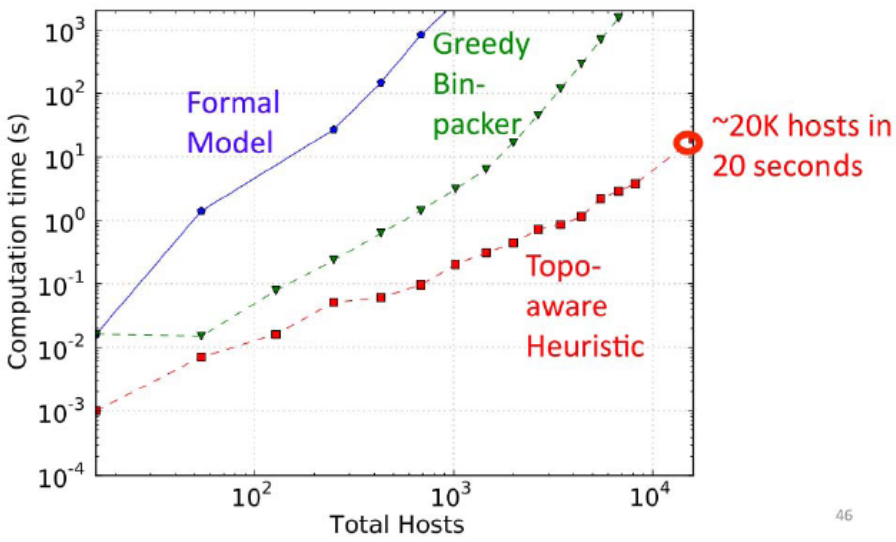
Type	Constraint	Description
Multi-Commodity Flow	Capacity	traffic <= link rate?
	Flow Conservation	packets in = packets out?
	Demand Satisfaction	bandwidth >= demand?
Our Additions	Flow on active links only	link off ←→ no flow
	Connect switches and links	switch off ←→links off

Does not scale

GREEDY BIN PACKING

- For each flow, evaluates all possible flows, and chooses the left-most one with sufficient capacity

SCALABILITY



TOPOLOGY-AWARE HEURISTICS

- Active switches == total bandwidth demand / capacity per switch
- Determine which switches are active, and pack flows to the active switches
- Add more switches for fault tolerance and connectivity

Cabling best practices

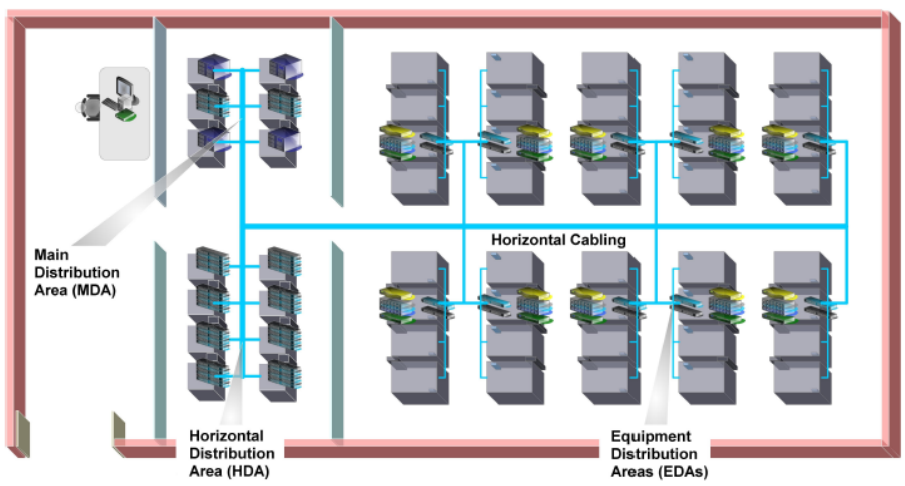


Figure 1. Top view of a data center layout showing a Main Distribution Area (MDA), a Horizontal Distribution Area (HAD) and several Equipment Distribution Areas (EDA).

Using a Structured Approach

The structured approach to cabling involves designing cable runs and connections to facilitate identifying cables, troubleshooting, and planning for future changes. In contrast, spontaneous or reactive deployment of cables to suit immediate needs often makes it difficult to diagnose problems and to verify proper connectivity.

Using a structured approach means establishing a Main Distribution Area (MDA), one or several Horizontal Distribution Areas (HDAs), and two-post racks for better access and cable management. The components selected for building the MDA and the HDA should be of good quality and able to handle anticipated and future loads, as this area will house the bulk of the cabling. Include horizontal and vertical cable managers in the layout. The MDA will house the main cross-connects as well as the core networking equipment. The HDA will house the cross-connects for distributing cables to the Equipment Distribution Areas (EDAs). Patch cables will be used to connect equipment such as servers and storage using the patch panels at their designated EDA.

Plan the layout of the equipment racks within the data center. Cables will be distributed from the HDA to the EDA using horizontal cabling. Dynamic data center environments call for a great deal of flexibility in connectivity, and the objective is to implement a cabling system with copper and fiber media capable of transmitting Ethernet, Fiber Channel, and any other protocols specific for the environment. Ensure that you address both current and future port counts and applications needs.

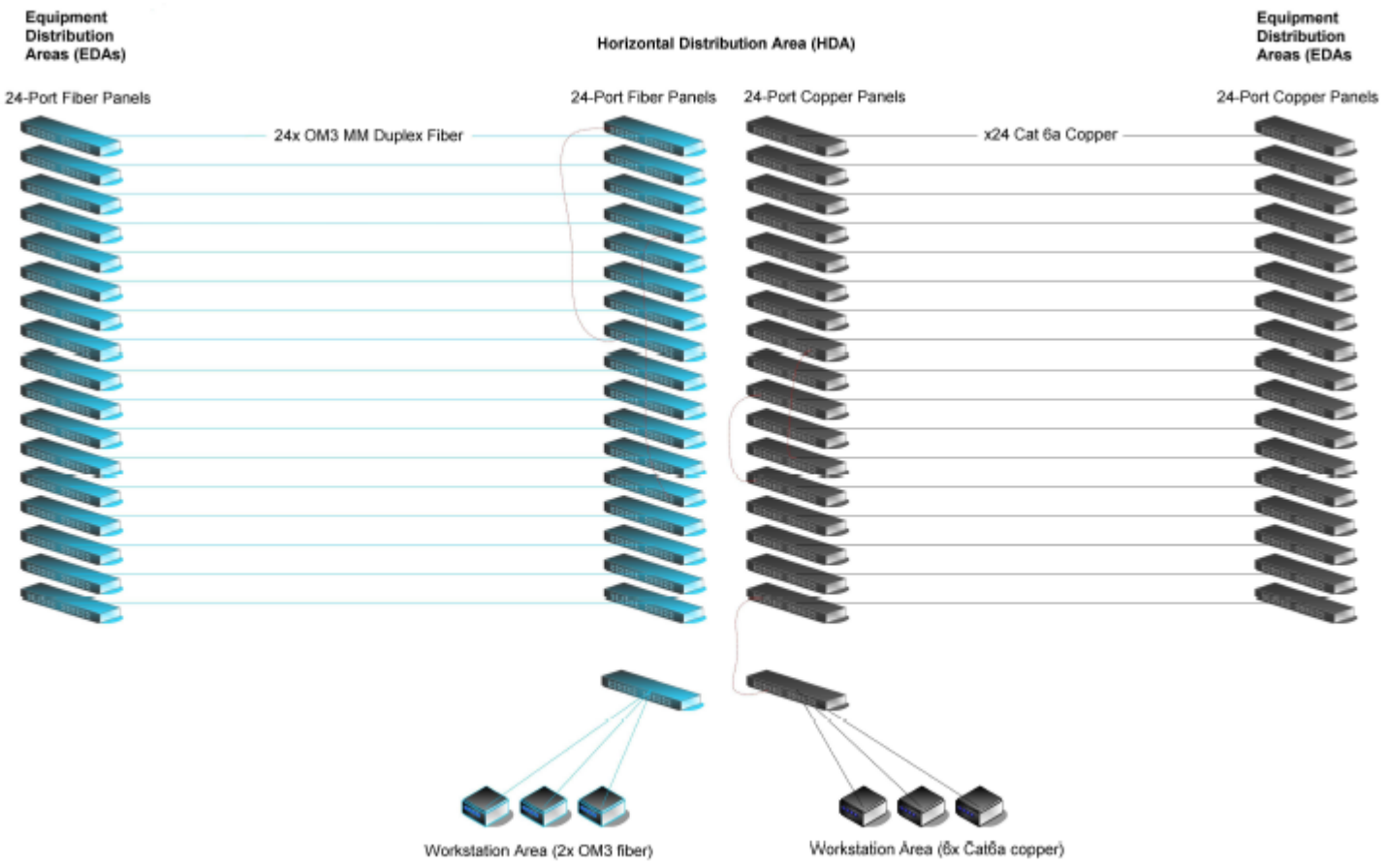


Figure 2. Sample logical diagram of the cable distribution between an HDA and several EDAs using horizontal cables.

Each scenario brings about its own challenges and customization requirements. It is important to digest the TIA-942 and the TIA/EIA-568 industry guidelines and to establish the cabling into some sort of structure. Each cabling component has an important role in the overall infrastructure and the trick is to carefully select and apply the right mix.

- Start with the proposed network topology that includes the network components in the data center.
- Next, identify common cable distribution points in the cable layout diagram such as network switches, server concentration areas, and workstation areas—and their locations. These will help identify the required cabling distribution areas (for example, MDA) and cabling components within these distribution areas.
- The logical cabling should eventually map to a physical map of the cabling for the data center. Plan your current and future port counts and cable media, and use that information to calculate quantities.
- Work with a reputable cabling contractor to survey the data center environment and to establish the exact locations for the proposed cable distribution points. Start with the Main Distribution Area and gradually expand out to the Equipment Distribution Areas.

NOTE: According to TIA-942 recommendations, there must one Main Distribution Area, one or more Horizontal Distribution Areas, and one or more Equipment Distribution Areas within a data center.

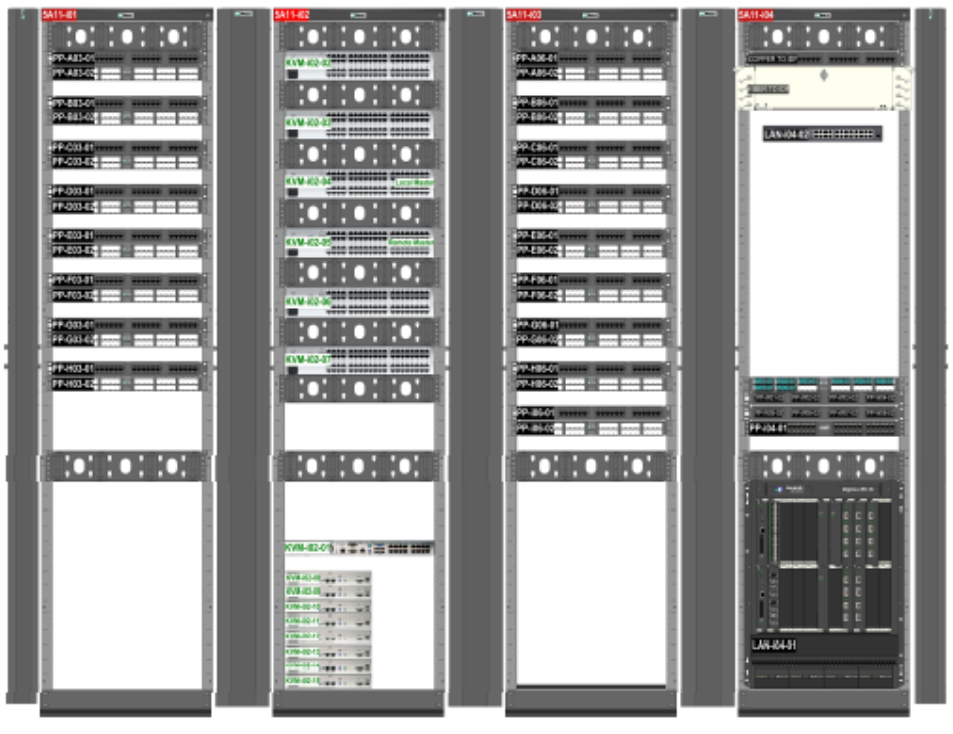


Figure 3. Layout of a Horizontal Distribution Area that includes patch panels, cable managers, and network equipment.

Modular Data Cabling

Modular cabling systems for fiber and copper connectivity are gaining in popularity. Modular cabling introduces the concept of plug-and-play, simplifying the installation of cables and drastically reducing labor time and costs. Cables are usually pre-terminated and tested at the factory.

As equipment prices continue to drop, vendors continue to build better options. The main difference to consider currently is the cost of modular components versus the cost of labor for a non-modular but structured offering. Although modular cabling saves you time and money when you want to modify the infrastructure yourself, the tradeoff is less flexibility and a potential commitment to stay with the chosen vendor for continued compatibility.

Cabling High Density, High Port Count Fiber Equipment

As networking equipment becomes denser and port counts in the data center increase to several hundred ports, managing cables connected to these devices becomes a difficult challenge. Traditionally, connecting cables directly to individual ports on low port-count equipment was considered manageable. Applying the same principles to high-density and high-port-count equipment makes the task more tedious, and it is nearly impossible to add or remove cables connected directly to the equipment ports.

Using fiber cable assemblies that have a single connector at one end of the cable and multiple duplex breakout cables at the other end is an alternative to alleviate cable management. Multifiber Push-On (MPO) cable assemblies are designed to do just that. The idea is to pre-connect the high-density, high-port-count Lucent Connector (LC) equipment with LC-MPO fan-out cable (shown in Figure 4) to dedicated MPO modules within a dedicated patch panel. Once fully cabled, this patch panel functions as if it were “remote” ports for the equipment. These dedicated patch panels ideally should be located above the equipment whose cabling they handle for easier access to overhead cabling. Using this strategy drastically reduces equipment cabling clutter and improves cable management.

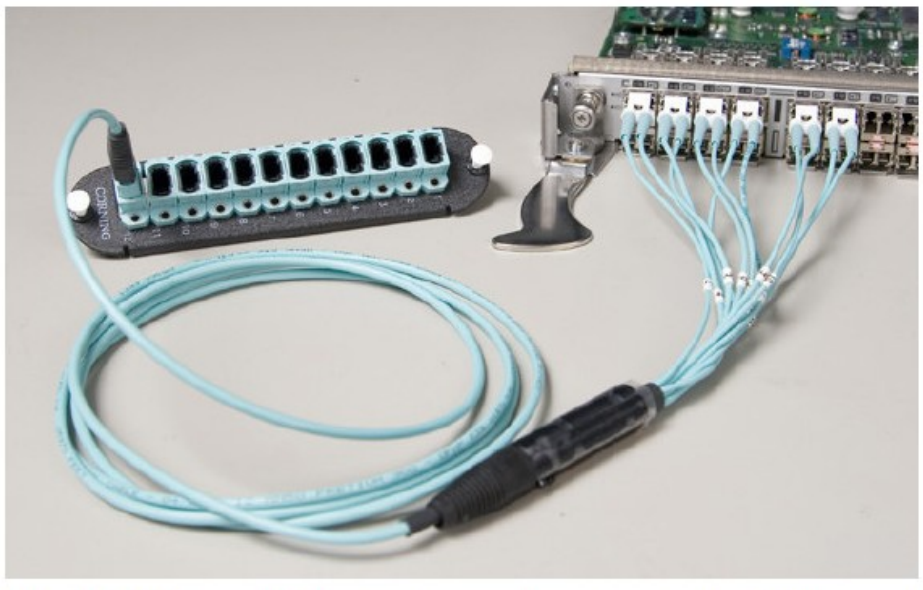


Figure 4. LC-MPO fan-out cable consolidates six duplex LC ports into one MPO connection.

As an example, the MPO module shown in Figure 4 is housed into a modular patch panel installed above a Fiber Channel director switch at the EDA. MPO trunk cables are used to link this patch panel to another modular patch panel located at the HDA. The patch panel at the HDA converts the MPO interface back to the LC interfaces using MPO-to-LC cassettes. MPO trunk cables can accommodate up to 72 individual fibers in one assembly; providing 36 duplex connections.

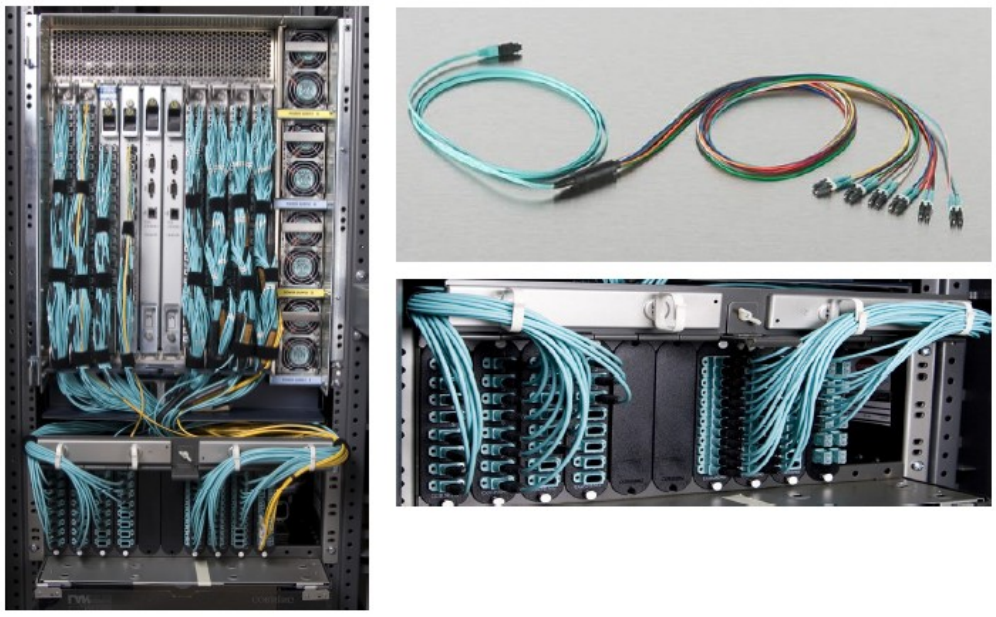


Figure 5. Director-class switch using MPO fan-out cabling structure.

There are a number of standards organizations and standards. The three best-known cabling standards organization are listed below:

- United States — ANSI/TIA/EIA-568 from the Telecommunications Industry Association (TIA)
- International — ISO/IEC IS 11801 (also referred to as Generic Customer Premises Cabling)
- International — TIA-942 from the TIA

NOTE: Cabling standards are reviewed and changed every five to ten years, which allows them to keep pace with technology advances and future requirements. Know and trust the standards, and apply common sense when designing, implementing, testing, and maintaining data center cabling.

Using Color to Identify Cables

Color provides quick visual identification. Color coding simplifies management and can save you hours when you need to trace cables. Color coding can be applied to ports on a patch panel: patch panels themselves come with different color jacks or have colored inserts that surround the jack. Cables are available in many colors (the color palette depends on the cable manufacturer). Apply these colors to identify the role/function of a cable or the type of connection.

Below is an example color scheme for patch cables.

Color	Type	Application (connections may by through patch panels)
Aqua	OM3 fiber	LAN/SAN device to device
Yellow	Single Mode Fiber	LAN/SAN device to device over long distance
Orange	OM1 or OM2 fiber	LAN/SAN device to device
Blue	Copper	LAN device to device
Green	Copper	KVM host to KVM switch, KVM switch to LAN switch, KVM switch to KVM switch
Yellow	Copper	Serial host to Terminal Server, Terminal Server to LAN switch
White	Copper	Power strip to LAN switch

Establishing a Naming Scheme

Once the logical and physical layouts for the cabling are defined, apply logical naming that will uniquely and easily identify each cabling component. Effective labeling promotes better communications and eliminates confusion when someone is trying to locate a component. Labeling is a key part of the process and should not be skipped. A suggested naming scheme for labeling and documenting cable components is suggested below (examples appear in parentheses):

- Building (**SJ01**)
- Room (SJ01-**5D11**)
- Rack or Grid Cell: Can be a grid allocation within the room (SJ01-5D11-**A03**)
- Patch Panel: instance in the rack or area (SJ01-5D11-A03-**PP02**)
- Workstation Outlet: Instance in the racks or area (SJ01-5D11-A01-**WS02**)
- Port: Instance in the patch panel or workstation outlet (SJ01-5D11-A03-PP02_**01**)
- Cable (each end labeled with the destination port)

(You can exclude Building and Room if there is only one instance of this entity in your environment.

Once the naming scheme is approved, you can start labeling the components. Be sure to create a reference document that will become part of the training for new data center administrators.

Patch Panels

Patch panels allow easy management of patch cables and link the cabling distribution areas. Multimedia patch panels, which allow several different cable connectors to be used in the same patch panel, are ideal. The main types of connectors that should be considered are LC for fiber and RJ-45 for copper. Although mixing cable types within the same patch panel is not best practice, it is good to have this flexibility for housing ad-hoc cable types. The best practice is to separate the fiber cabling from the copper cabling, using separate patch panels.

Colored jacks or bezels in the patch panel allow easy identification of the ports and the applications they are intended for. Patch panels also come in modular styles, for example, for an MPO structured system. The tradeoff for the higher cost of materials is this: some of this cost is recovered from faster installation and thus lower labor cost.

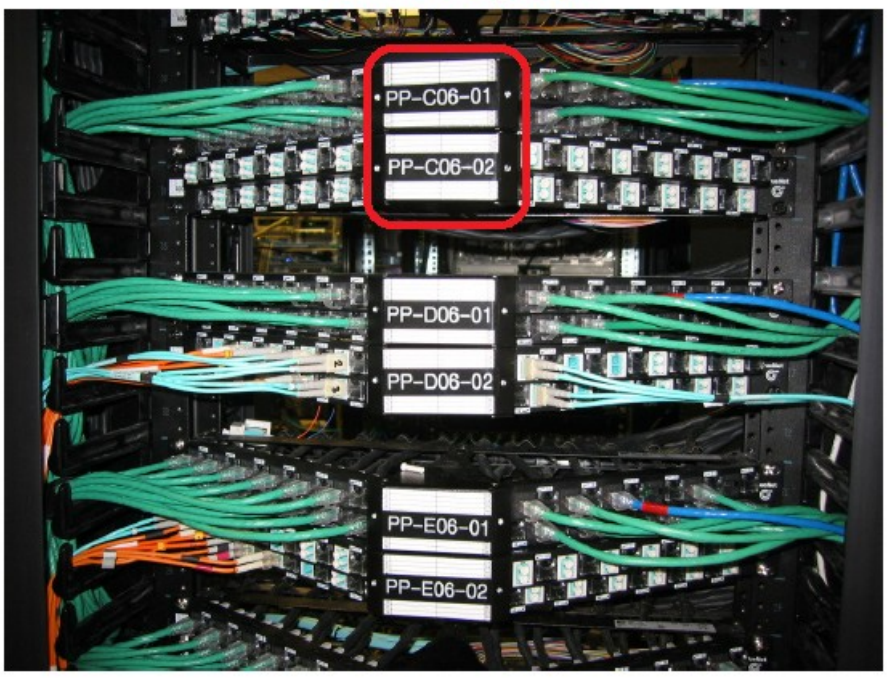


Figure 6. Angled patch panels allow cables to be routed directly into the vertical cable managers.

Angled patch panels, such as those shown in Figure 6, are ideal for high-density areas, as they do not require additional cable managers to be installed above and below the patch panel. They also allow for higher concentration of cables. When selecting patch panels, consider the following:

- √ Spacing between ports aids insertion and removal of cables
- √ Sturdy connectors: some panels have loose connectors and tend to fall out during cable installation and removal
- √ Orientation of ports in the panel: the top row and bottom row cable clips should face outward. Test these connections with your patch cables.
- √ One-piece dust covers for ports (recommended for high traffic areas)
- √ Density supported (24 ports or 48 ports per 1U panel)
- √ Compatibility with your racks
- √ Space for labeling on the front of the panel
- √ Compatibility with Industry standard connectors and racks
- √ Added cable support for the intended cable types on the back of the panel. This is critical and overlooked by many manufacturers.

Horizontal and Backbone Cables

Choose the fire-rated plenum type. These cables may not be as flexible as the patch cords, because they are meant for fairly static placements, for example, between the EDA and the HDA. There are no high-density copper solutions, but you can choose a modular cabling system such as the MRJ21 connector system. For fiber, high density involving 24-strand to 96-strand cables is adequate. Fiber breakout cables provide additional protection, but add to the diameter of the overall cable bundle. For fiber, MPO trunk cables (up to 72 fiber strands can be housed in one MPO connection) can be installed if you are using MPO style cabling.

Evaluate the cost of materials and labor for terminating connections into patch panels. These cables will most likely end up under raised floors, or over the ceiling, or in overhead cable pathways—out of view and touch from end users.

Horizontal Cable Managers

Horizontal cable managers (shown in Figure 7) allow neat and proper routing of the patch cables from equipment in racks and protect cables from damage. These cable managers take up the much-needed space in racks, so a careful balance between cable manager height and cable density supported is important. 1U and 2U horizontal cable managers are the most common varieties. The density supported varies with the height and depth of the manager. Horizontal cable managers come in metal and flexible plastic—choose the ones that work best for you. The ideal cable manager has a big enough lip to easily position and remove cables, and has sufficient depth to accommodate the quantity of cables planned for that area. Note that you should allow 30% space in the cable managers for future growth.

Choose these cable managers carefully so that cable bend radius is accommodated. Make sure that certain parts of the horizontal cable manager are not obstructing equipment in the racks, and that those individual cables are easy to add and remove. Some cable managers come with dust covers. For dynamic environments, however, dust covers can be an obstacle when quick cable changes are required.

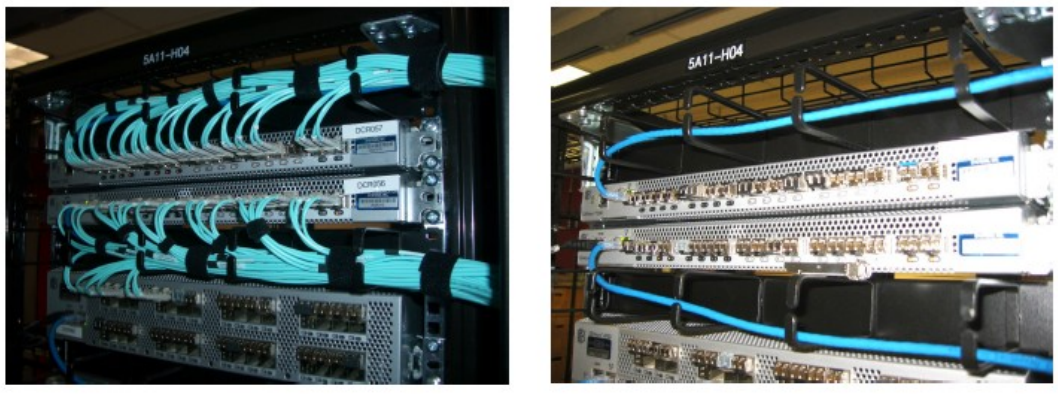


Figure 7. Cables from devices are routed through the horizontal cable managers.

Vertical Cable Managers

For vertical cable managers, look for the additional space required to manage the slack from patch cords, and ensure that they can easily route the largest cable diameter in your plan. The most convenient managers available on the market have hinged doors on both sides of the manager for pivoting the door from either side, and allow complete removal of the doors for unobstructed access. Allow for 50 percent growth of cables when planning the width (4" width for edge racks and 6" width for distribution racks are typical) and depth (6" depth is typical) of the vertical cable manager. Additionally, use d-rings type cable managers to manage cables on the back side of the racks in dynamic environments. For static environments, you can consider installing another vertical cable manager behind the racks, which does not block access to components in the space between the racks.

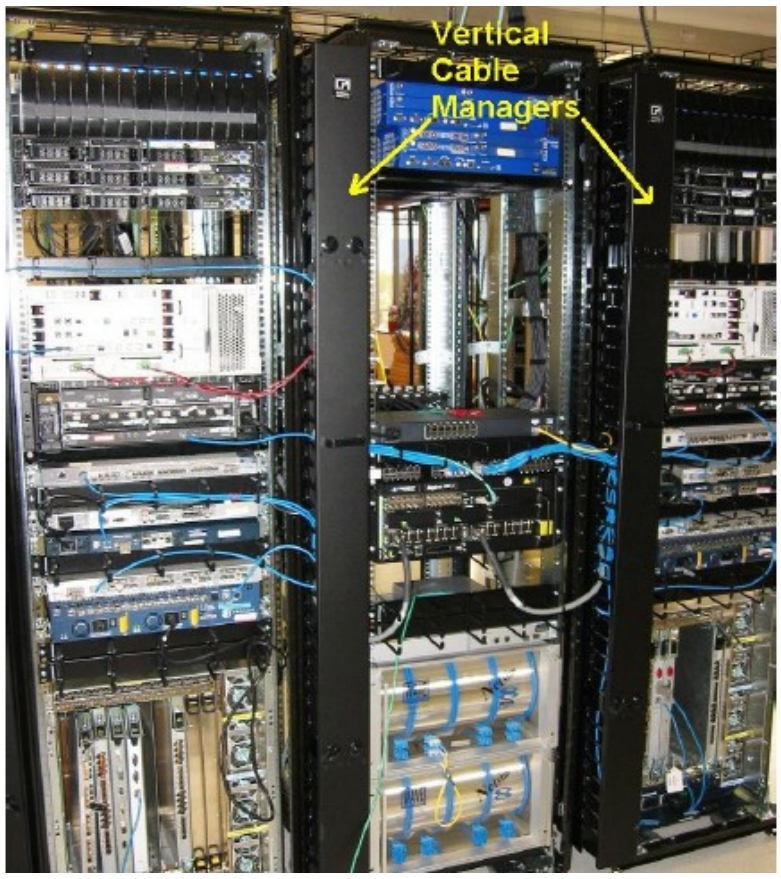


Figure 8. Vertical cable managers are installed between racks. Cables are routed from the horizontal cable manager into the vertical cable managers.

Cable Ties

Use cable ties to hold a group of cables together or to fasten cables to other components. Choose Velcro-based cable ties versus zip ties, as there is a tendency for users to over-tighten zip ties. Over tightening can crush the cables and impact performance. Velcro cable ties come in a roll or in pre-determined lengths. Bundle groups of relevant cables with ties *as you install*, which will help you identify cables later and facilitate better overall cable management.

Labelers

Labelers are used to print sticky labels for devices and cables. Here are some considerations when you choose a hand-held labeler:

- √ Should be capable of operating using batteries
- √ Can print labels on smooth, textured, flat, and curved surfaces
- √ The actual label material should resist solvents, chemicals, and moisture
- √ Labels are durable and resist fading
- √ Adhesive should be long-lasting

If you choose a labeler with bundled software, install it on a client workstation. You can then customize labels, print labels in batches, and store the formats for future printing.

Overhead Cable Pathways

Overhead cable pathways or trays allow placement of additional cables for interconnecting devices between racks on an ad-hoc basis. Check support for cable bend radius, weight allowance, sagging points for cables, and flexibility in installing the pathways. In addition, ensure that pathways allow cable drop points where needed. These trays should be easy to install and to customize.

Figure 9. Overhead cable pathways used for routing cables between distribution areas or for ad-hc cabling between distant racks.

Building a Common Framework for the Racks

The goal of this step is to stage a layout that can be mirrored *across all racks in the data center* for consistency, management, and convenience. Starting with an empty 4-post rack or two, build out and establish an internal standard for placing patch panels, horizontal cable managers, vertical cable managers, power strips, KVM switch, serial console switch, and any other devices that are planned for placement into racks or a group of racks. The idea is to fully cable up the common components while monitoring the cooling, power, equipment access, and growth for the main components in the racks (such as servers and network switches).

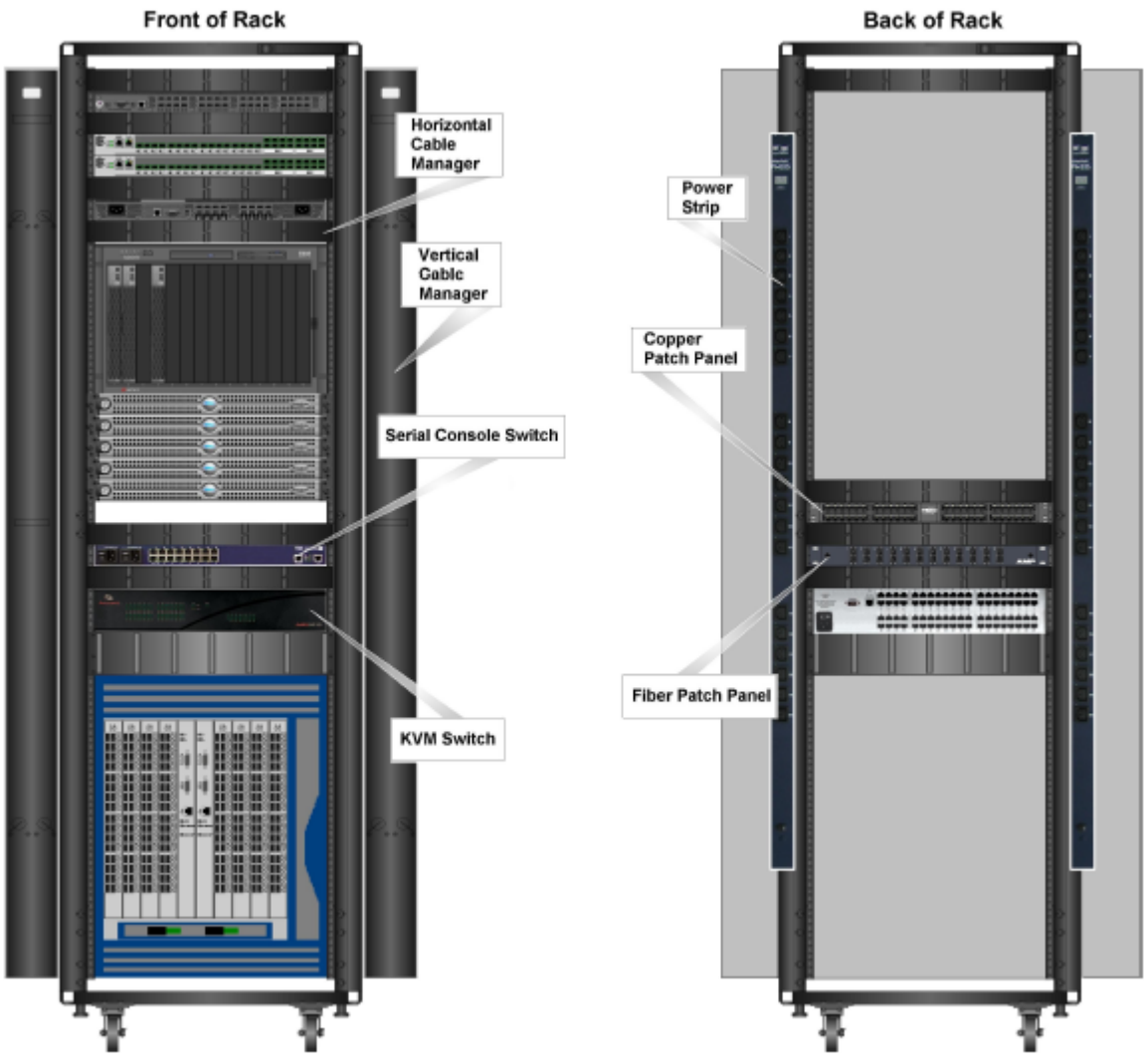


Figure 10. Front and back view of a rack showing placements of common cabling components.

A good layout discourages cabling in between racks due to lack of available data ports or power supply ports. Allow more power outlets and network ports than you need. This will save you money in the long run as rack density increases, calling for more power and network connectivity. Using correct length cables, route patch cables up or down through horizontal patch panels, avoiding overlapping other ports. Some cable slack may be needed to enable easy removal of racked equipment.

Once you are satisfied that the rack is populated and cabled efficiently, label, document and establish this as an internal standard for your data center. Once you have created the ideal layout of a rack, you will be able to get an idea of cable density, power consumption, weight, and the heat generated per rack—for the entire data center. The actual figures will vary from rack to rack, but this will establish baseline metrics.

Vertical cable managers should be mounted between racks. The outermost rack may not need a vertical cable manager if you decide to route cables using the between-rack vertical cable managers only. Also, ensure that the front of the vertical cable manager is flush with the front of the horizontal cable manager to provide better routing and management of the cables.

Placement of horizontal cable managers is important too. Use one horizontal cable manager to route cables between two adjacent 1U switches that have a single row of ports. For switches and equipment that have two rows of ports, route the cables from the top row of the equipment to a horizontal cable manager placed above this equipment; route the cables from the bottom row of the equipment to a horizontal cable manager placed below the equipment. Bladed systems, especially ones with high port counts, usually come with recommended cable routing guidelines—ensure that are addressed in your layout.

Comparison of Cable Types

The following table compares fiber optic and copper cables.

Fiber						
Medium	**Standard**	**Maximum Rate**	**Maximum Distance**	**Maximum Bandwidth**	**Common Connectors**	**Common Applications**
MMF (62.5/125 microns)	OM1	1 Gbit/sec	300 m[2]	200 MHz[2]	LC, SC, ST, MPO	FDDI, Ethernet
MMF (50/125 microns)	OM2	1 Gbit/sec	500 m[2]	500 MHz[2]	LC, SC, ST, MPO	SANs, High Speed Ethernet
MMF Laser Optimized (50/125 microns)	OM3	10 Gbit/sec	300 m[2]	2000 MHz[2]	LC, SC, ST, MPO	SANs, High Speed Ethernet
MMF Laser Optimized (50/125 microns)	OM3E	10 Gbit/sec	550 m[2]	4700 MHz[2]	LC, SC, ST, MPO	SANs, High Speed Ethernet
SMF (9/125 microns)	OS1	10 Gbit/sec	40 km	Infinite	LC, SC, ST, FC, FJ, MPO	SANs, WANs, Telco

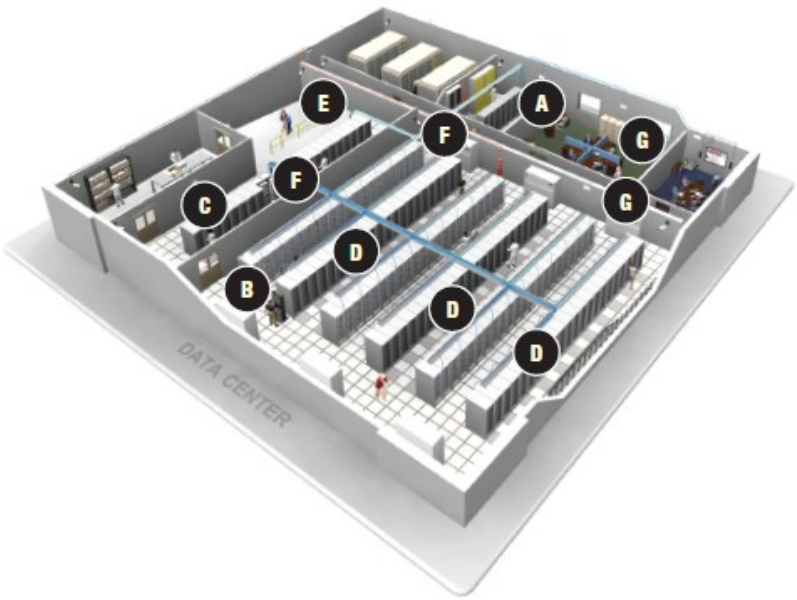
Copper [1]						
Medium	**Standard**	**Maximum Rate**	**Maximum Distance**	**Maximum Bandwidth**	**Common Connectors**	**Common Applications**
UTP	Cat 1	1Mb/sec		1 MHz		Analog voice ISDN Doorbell wiring
UTP	Cat 2	4 Mb/sec		4 MHz		IBM Token Ring
UTP, S/STP, S/UTP	Cat 3	10 Mb/sec	100 m	16 MHz		Voice/Data on 10BASE-T Ethernet
UTP, S/STP, S/UTP	Cat 4	16 Mb/sec	100 m	20 MHz		Token Ring
UTP, S/STP, S/UTP	Cat 5	100 Mb/sec	100m	100 MHz	RJ-45	100 Mbps Networks 155 Mbps ATM
UTP, S/STP, S/UTP	Cat 5e/ Class D	1 Gbit/sec	100 m	100 MHz	RJ-45	100 Mbps Networks 155 Mbps ATM
UTP, S/STP, S/UTP	Cat 6/ Class E	10 Gbit/sec	55 m	250 MHz	RJ-45	Broadband Most popular for new installs
UTP, S/STP, S/UTP	Cat 6a/ Class Ea	10 Gbit/sec	100 m	500 MHz	RJ-45	10GBASE-T
S/STP	Cat 7/ Class F	10 Gbit/sec		600 MHz	Vary by Manufacturer	Full-motion video Government/ Industrial environments
S/STP	Cat 7a/ Class Fa	10 Gbit/sec		1,000 MHz	Vary by Manufacturer	Full-motion video Government/ Industrial environments

[1] For copper cabling, only Cat 5e or greater is relevant in today's data centers

[2] Rating for 850 nm laser sources

The following table compares UTP and STP cables.

	Advantages	Disadvantages
UTP	Easier to implement	Requires more space due to a larger cable diameter
	Better performance for PoE plus	External noise suppression
	Better suited for dynamic environments	Category 6a is not a ratified standard yet
	Slightly cheaper	
STP (S/UTP, S/STP, STP)	Smaller diameter	Cost of labor is higher
	Better internal and eternal noise suppression	Low acceptance in North America
	Better suited for static or noisy (EMI/RFI) or secure environments	Installation has to be precise, more time consuming
	Some exceed performance	Strict grounding requirements
	AXT minimized by shielding rather than by space	
	Better resistance to EMI and RFI	



A Entrance/Telecommunications Room (TR)

The entrance room of the Data Center is the location for access provider equipment, demarcation points and interface with other campus locations. The TIA-942 standard recommends locating the entrance room outside of the computer room for security purposes. Specific Belden products found in the entrance room include:

- Fiber*Express* Ultra HD Cassettes and Patch Cords
- Fiber*Express* Brilliance® Fiber Connectors
- 10GX and GigaBIX IDC Systems
- 735 and 734 Series Coax Cables
- High-Density 2- and 4-Post Racking Systems
- Enclosures
- Power Distribution Units (PDUs)
- Surveillance (CCTV) Camera Cables
- Door Access (Access Control) Cables

Connected to the Data Center MDA through backbone cabling, TRs are spaces for housing equipment, cable terminations and cross connects that serve office areas on specific floors. In addition to voice, data, and wireless systems, TRs can house equipment for life safety/fire systems, security, and building automation systems. Belden offers a variety of products to support all of these systems within the TR:

- 10GX Pre-Term Coupler Patch Panels and Cable Assemblies
- 10GX Patch Panels and Modular Cords
- 3600 Pre-Term Coupler Patch Panels and Cable Assemblies
- CAT6+ Patch Panels and Modular Cords
- Fiber*Express* Ultra HD Pre-Term System
- Fiber*Express* Ultra HD Panels and Patch Cords
- Fiber*Express* Brilliance Fiber Connectors
- Equipment Cable Harnesses
- High-Density 2- and 4-Post Racking Systems
- Enclosures
- Power Distribution Units (PDUs)

B Zone Distribution Area (ZDA)

The optional ZDA acts as a consolidation point within the horizontal cabling run between the HDA and Equipment Distribution Area. The ZDA allows frequent reconfiguration and provides additional flexibility. Belden products typically deployed in the ZDA include the following:

- 10GX Ultra High-Density Patch Panels and Cords
- CAT6+ Ultra High-Density Patch Panels and Cords
- 10GX and GigaBIX IDC Systems

C Main Distribution Area (MDA) & Horizontal Distribution Area (HDA)

The MDA houses the main cross connect and the core routers and switches. The HDA houses cross-connects and active equipment (switches) for connecting to the equipment distribution area and storage area network (SAN). Per the TIA-942 standard, both the MDA and HDA require separate racks for fiber, UTP and coax cable. Several Belden products provide maximum performance, density and management in the MDA and HDA, including:

- 10GX Pre-Term Coupler Patch Panels and Cable Assemblies
- 10GX Patch Panels and Modular Cords
- 3600 Pre-Term Coupler Patch Panels and Cable Assemblies
- CAT6+ Patch Panels and Modular Cords
- Fiber*Express* Ultra HD Pre-Term System
- Fiber*Express* Ultra HD Panels and Patch Cords
- Fiber*Express* Brilliance Fiber Connectors
- Equipment Cable Harnesses
- High-Density 2- and 4-Post Racking Systems
- Switch Enclosures
- Side-to-side Airflow Managers
- Enclosure Power Distribution Units with Monitoring
- Climate Monitoring Solutions
- GarrettCom Active Devices
- Rack-Level Access Control

D Equipment Distribution Area (EDA)

The EDA is where equipment enclosures and racks house the servers and where the horizontal cabling from the HDA is terminated at patch panels. In the EDA, racks and cabinets should be arranged in a hot aisle/cold aisle configuration along with airflow systems that maintain proper separation of supply (cold) and exhaust (hot) air. Belden offers a variety of products for the EDA, including our high-end freestanding enclosures that help provide optimal airflow and ease of management:

- 10GX Pre-Term Coupler Patch Panels and Cable Assemblies
- 10GX Patch Panels and Modular Cords
- 3600 Pre-Term Coupler Patch Panels and Cable Assemblies
- CAT6+ Patch Panels and Modular Cords
- Fiber*Express* Ultra HD Pre-Term System
- Fiber*Express* Ultra HD Panels and Patch Cords
- Fiber*Express* Ultra HD Cassettes and Patch Cords
- Fiber*Express* Brilliance Fiber Connectors
- Server Enclosures
- Airflow Management Systems
- Enclosure Power Distribution Units with Monitoring
- Climate Monitoring Solutions
- GarrettCom Active Devices
- Rack-Level Access Control

E Storage Area Network (SAN)

The SAN houses all data storage devices such as disk arrays, tape libraries and high-capacity optical disk libraries for applications like video surveillance. The use of a separate SAN eliminates the need to store data directly on servers, which provides better network capacity. Access to stored data must be fast, requiring high-speed connections from the HDA. Belden products for the SAN include:

- Fiber*Express* Ultra HD Pre-Term System
- Fiber*Express* MPO Cables
- Fiber*Express* Patch Cords
- High-Density 2- and 4-Post Racking Systems
- Enclosures
- Power Distribution Units (PDUs)
- Airflow Management Systems
- Surveillance (CCTV) Camera Cabling
- Door Access (Access Control) Cabling
- Rack-Level Access Control

F Backbone Cabling & Horizontal Cabling

The backbone cabling within the Data Center provides the critical connections between the entrance room, MDA and HDA. The backbone cabling in many of today's Data Centers supports 10 Gigabit transmission speeds for current and future applications. The horizontal cabling within the Data Center provides the connection between the HDA and EDA and SAN, including the optional ZDA. Belden products for the backbone and horizontal cabling include:

- 850 nm Laser-Optimized 50/125µm Multimode OM3 and OM4 Fiber*Express* Cables
- 10 Gigabit 4-Pair UTP Belden 10GX Cables
- Belden IBDN Category 6 Cables
- Fiber*Express* MPO Cables

G Support Offices and Open Areas

Whether it's cubicles, conference rooms, hallways or cafeterias, there are many areas throughout a facility or campus where people conduct day-to-day activities. From wired and wireless voice and data systems to fire alarm and security surveillance, Belden offers a variety of products that deliver these systems to where it matters most.

- KeyConnect Workstation Outlets and MediaFlex Outlets
- Belden IBDN Cables
- Surveillance (CCTV) Camera Cabling
- Door Access (Access Control) Cabling
- 2-Hour EVAC Systems Cabling
- Wall-Mount Racks and Enclosures