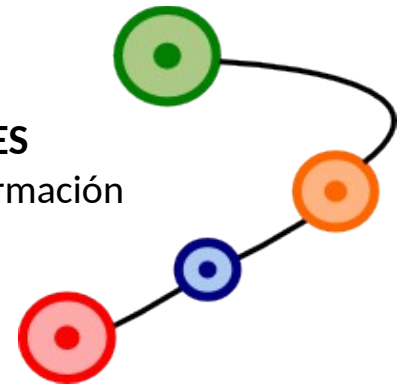




Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Infraestructura de Data Centers y Cloud

Normas de Infraestructura de Data Centers
Virtualización
Software Defined Networks (SDNs)

Equipo docente:

Fernando Lorge (florge@unlu.edu.ar)

Santiago Ricci (sricci@unlu.edu.ar)

Alejandro Iglesias (aaiglesias@unlu.edu.ar)

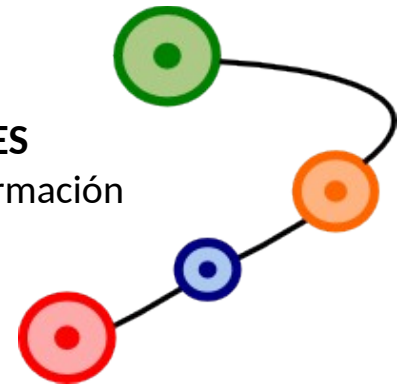
Mauro Meloni (maurom@unlu.edu.ar)

Patricio Torres (ptorres@unlu.edu.ar)



Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Normas de Infraestructura Data Centers

TIA/EIA 942

Uptime Institute Tier Performance Standard

Norma ICREA 2019

Data Center

Lugar físico donde se alojan los equipos y sistemas más críticos de una organización, y desde donde se prestan los servicios a todos los usuarios.



Diseño de Data Centers

- **Data Center:** *a building or portion of a building whose primary function is to house a computer room and its support areas.* (TIA-942)
 - Diseñado para albergar gran cantidad de equipos de cómputo y componentes asociados, tales como sistemas de almacenamiento y telecomunicaciones.
 - TIA-942 cubre los siguientes aspectos:
 - Espacios y distribución.
 - Infraestructura de cableado.
 - Niveles de Confiabilidad.
 - Consideraciones Medioambientales.
- Objetivos**
- Almacenar y proteger los datos de la organización
 - Procesar información
 - Proveer servicios (aplicaciones)
 - Proveer conectividad

Infraestructura de Data Centers

Arquitectura física

- Ubicación
- Seguridad
- Acondicionamiento espacio
- Provisión de energía
- Refrigeración

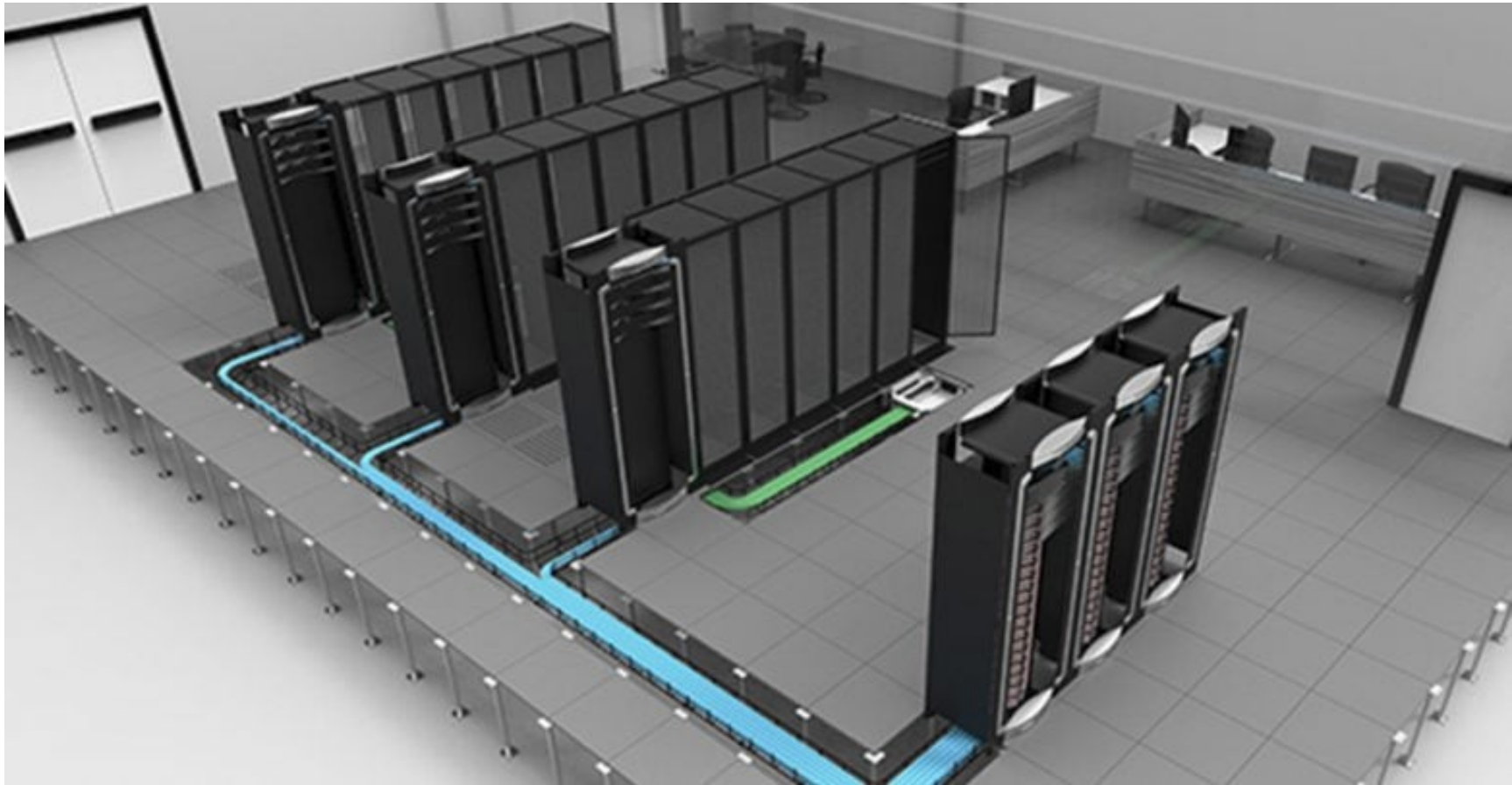
Arquitectura lógica

- Conectividad (switches, routers, ...)
- Sistemas de protección (firewalls)
- Equipos de almacenamiento (storage)
- Equipos de cómputo (servers)
- Sistemas de respaldo (backups)

Requisito: Funcionamiento 7x24 (los 365 días)

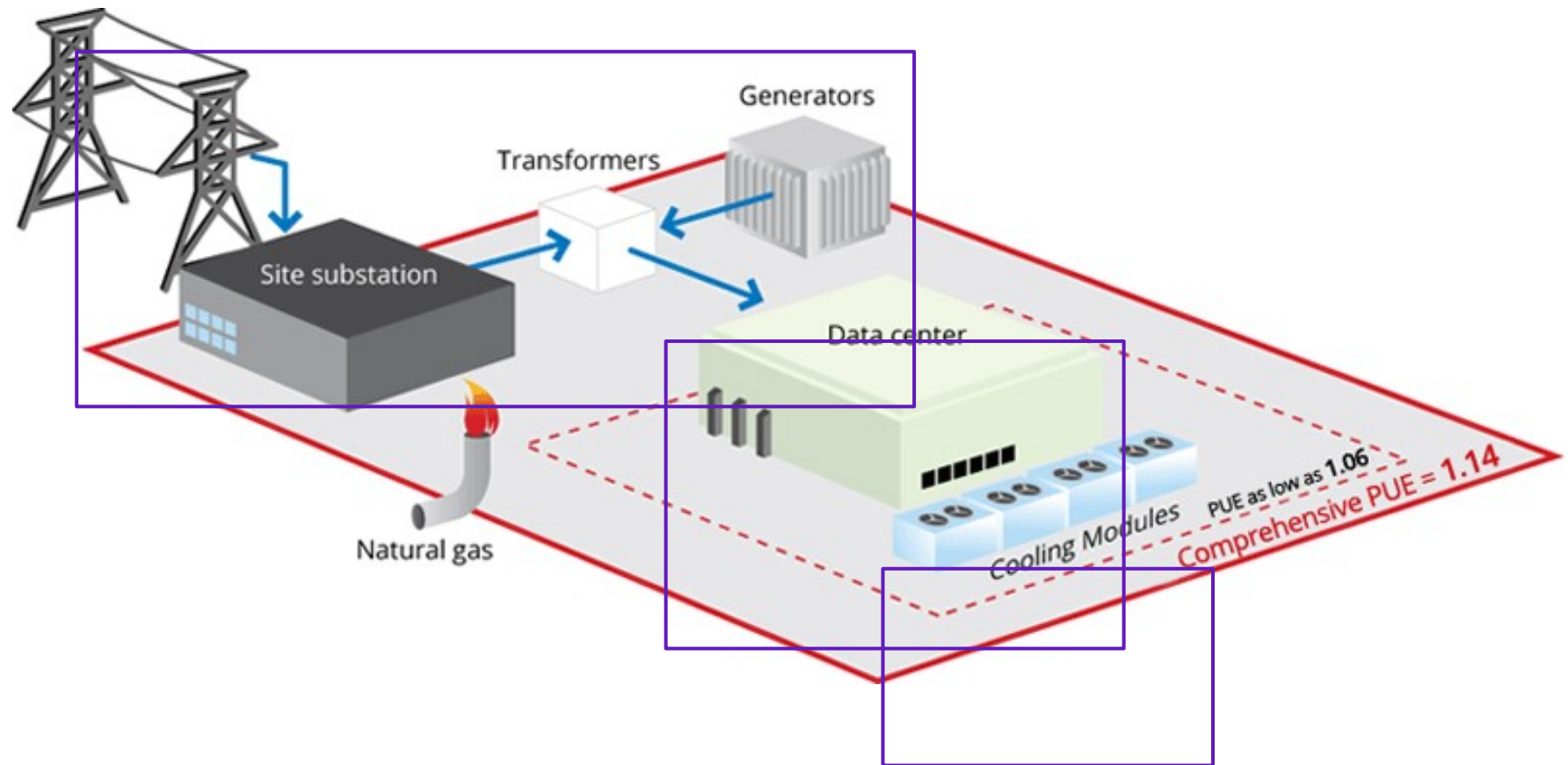
Infraestructura de Data Centers

Ejemplo: Server Room



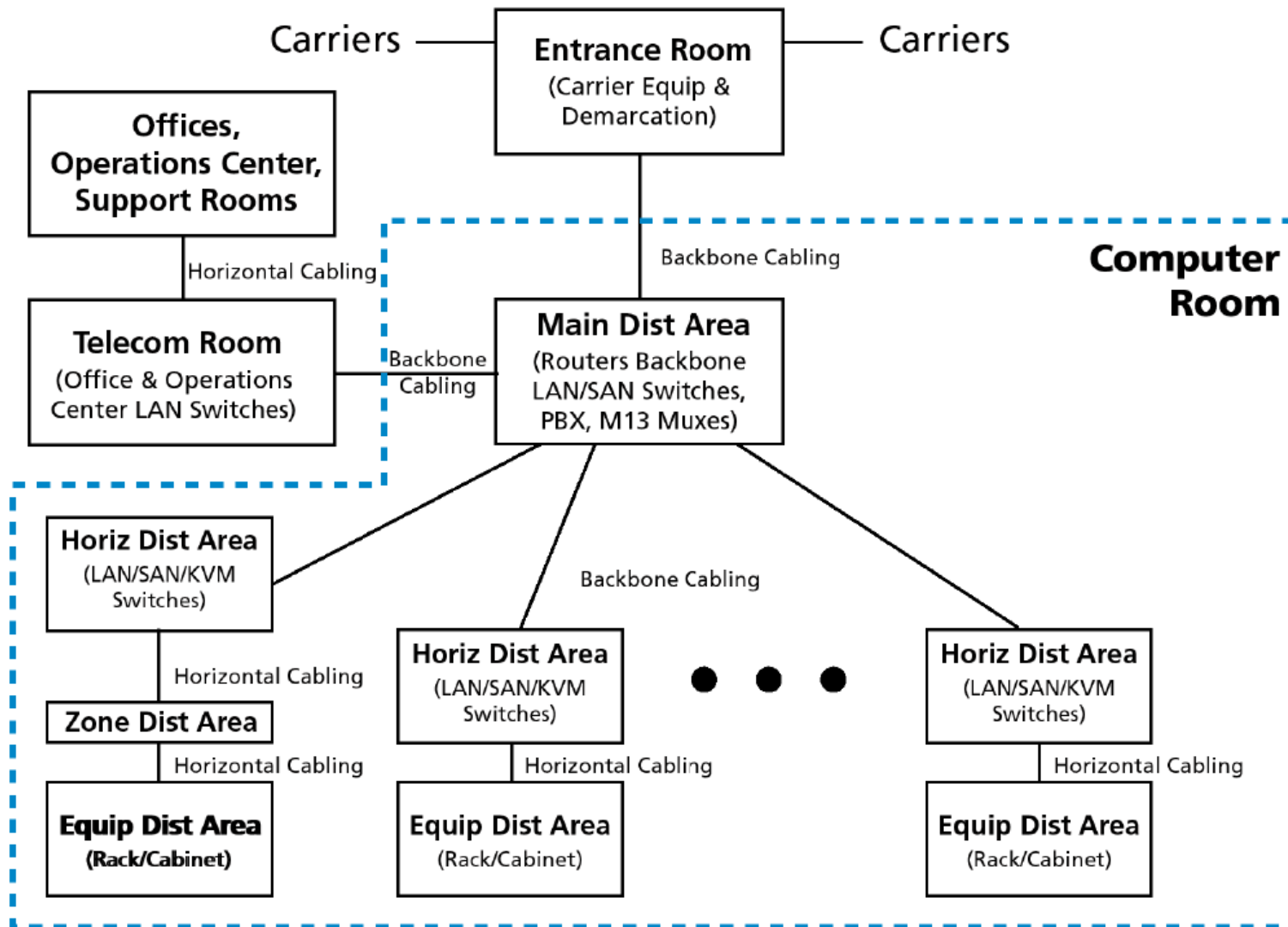
Infraestructura de Data Centers

Ejemplo: Datacenter Campus



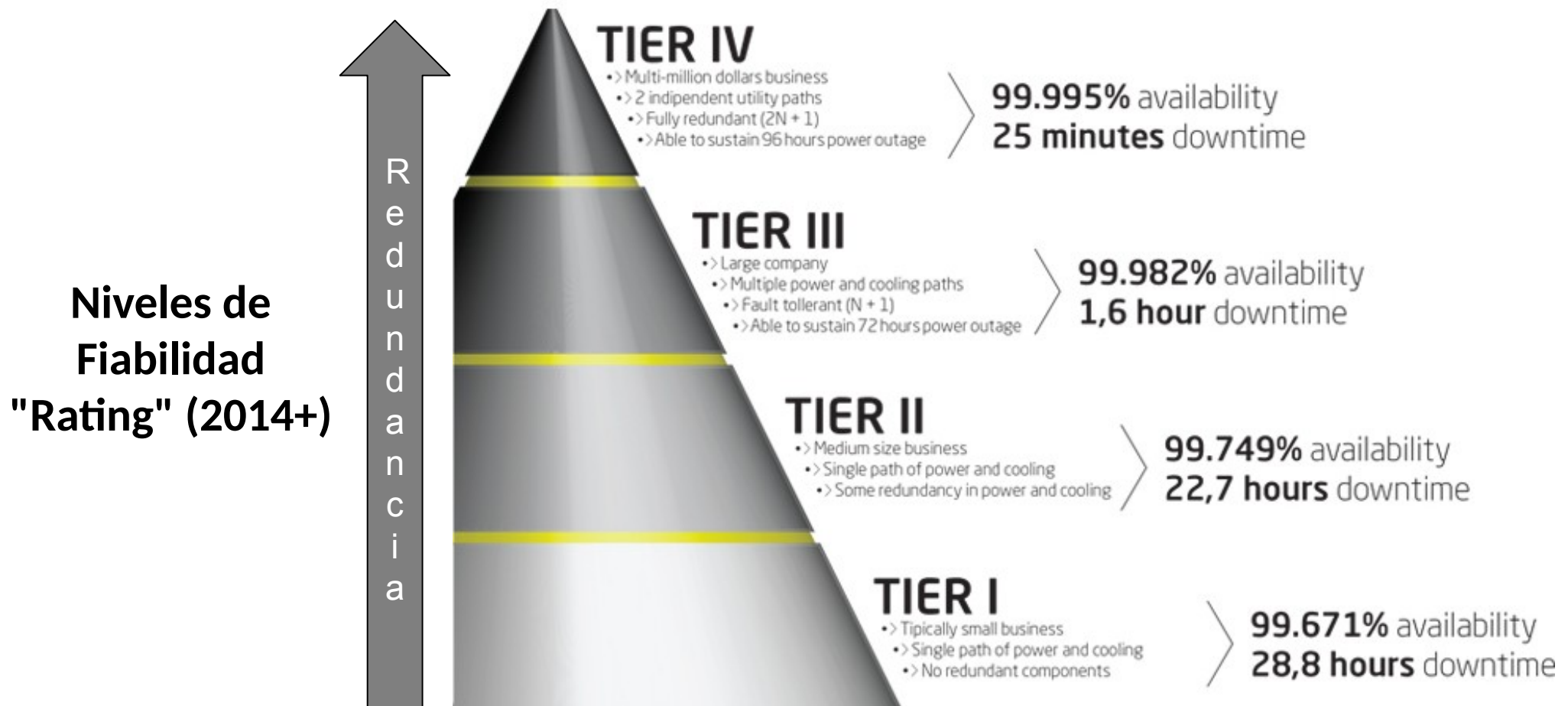
ANSI TIA-942

Telecommunications Infraestructure Standards for Data Centers



Certificaciones - TIA 942

Certificaciones - Estándar TIA 942 (2005+)



Uptime Institute

● Clasificación “Tier Performance Standard”

- No presenta una “checklist” o guía de referencia, sino que se determina el nivel del DataCenter de acuerdo a sus prestaciones. (*Performance confirmation tests and Operational impacts*).
- Certificación en tres categorías progresivas:
 - Diseño (TCDD)
 - Construcción (TCCF)
 - Sustentabilidad operacional (TCOS)



Uptime Institute Tier Performance Standard

- Clasificación en 4 niveles:
 - Tier I: “Basic Site Infrastructure” (DE 99.671%)
 - Tier II: “Redundant Capacity Components Site Infrastructure” (DE 99.741%)
 - Tier III: “Concurrently Maintainable Site Infrastructure” (DE 99.982%)
 - Tier IV: “Fault Tolerant Site Infrastructure” (DE 99.995%)



ICREA-131-2019

International Computer Room Experts Association

Niveles de confiabilidad y seguridad:

Nivel	Descripción	Disponibilidad
I	Sala de Cómputo en ambiente Certificado QADC (Quality Assurance Data Center)	95%
II	Sala de Cómputo en ambiente Certificado de clase mundial WCQA (World Class Quality Assurance Data Center)	99%
III	Sala de Cómputo confiable con ambiente Certificado de clase mundial S-WCQA (Safety World Class Quality Assurance Data Center)	99.9%
IV	Sala de Cómputo de alta seguridad con certificación HS-WCQA (High Security World Class Quality Assurance)	99.99%
V	Sala de cómputos de alta seguridad y alta disponibilidad con certificación de clase mundial HSHA-WCQA (High Security High Available World Class Quality Assurance Data Center)	99.999%
VI	Red de salas de cómputos redundante de alta disponibilidad con certificación de clase mundial RHA-WCQA)	99.9999%

Incluye:

Aspectos generales - Instalaciones eléctricas - Aire acondicionado - Comunicaciones - Entorno (Piso elevado, acabados, obra civil)
- Seguridad (CCTV, control de acceso, detección y supresión de incendios)

Certificaciones en el Mundo

TIA-942
.ORG

*The global registry of data centers
conforming to the ANSI/TIA-942 standard,
certified TIA-942 consultants and auditors*



Data Centers

Consultants & Auditors

Audit Services

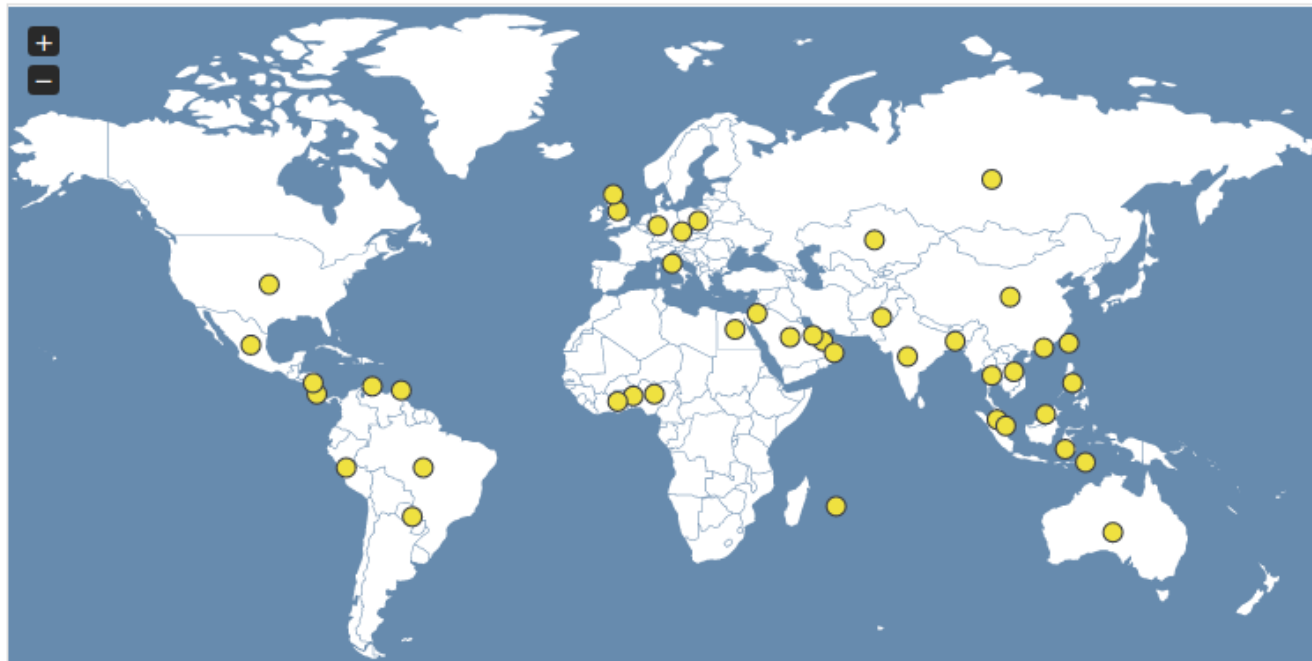
Training for
Consultants & Auditors

Search For Certified Data Center

Search By Form

Search By World Map

The map below indicates data centers with ANSI/TIA-942 compliant certification which are registered with www.TIA-942.org. To view the data center details, click on the country marker on the map or enter the search criteria in the search fields.



Certificaciones en el Mundo

Tiers is the Global Language of Data Center Performance
Tier Certification is the Accountability



Tier Certification Brings Worldwide Credibility

Uptime Institute®

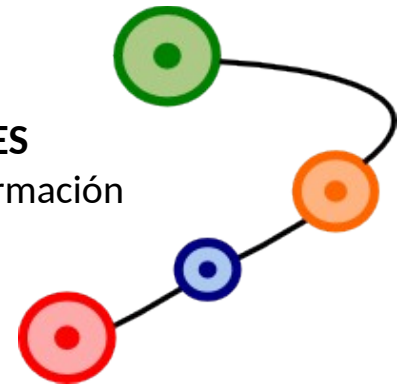
Referencias

- TIA-942 web site [[Link](#)]
- TIA-942 Data Center Standards Overview [[Link](#)]
- Uptime Institute - Tiers Standards [[Link](#)]
- Video Uptime Institute vs TIA-942-B vs ICREA [[Link](#)]
- Norma ICREA 2019 [[Link](#)]
- Videos Data Centers
 - Data Center Iplan Ringo (Parque Patricios, CABA)
<https://youtu.be/QRJPYx9ddQQ>
 - Data Center SkyOnline (San Telmo, CABA)
<https://youtu.be/zyPxAsBhJzo>
 - Data Center Arsat (Benavídez, Buenos Aires)
<https://youtu.be/aFBVmHrv7jc>
 - Data Center Telecom (Pacheco, Buenos Aires)
<https://www.youtube.com/watch?v=mr32PpPOv84>



Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Virtualización

Definición y Clasificación
Características
Técnicas de Virtualización
Definiciones formales

Virtualización

- Abstracción de los recursos de computación
- Abstracción → Aislación → Optimización de Recursos
- Desde el punto de vista de S.O.: memoria, dispositivos, archivos ...
- System Virtualization: Abstracción de una computadora completa, incluyendo memoria, CPU y periféricos.
- Hoy día la virtualización es capaz de proveer un entorno virtual para la ejecución de aplicaciones (o VMs), almacenamiento, memoria, red...



Virtualización

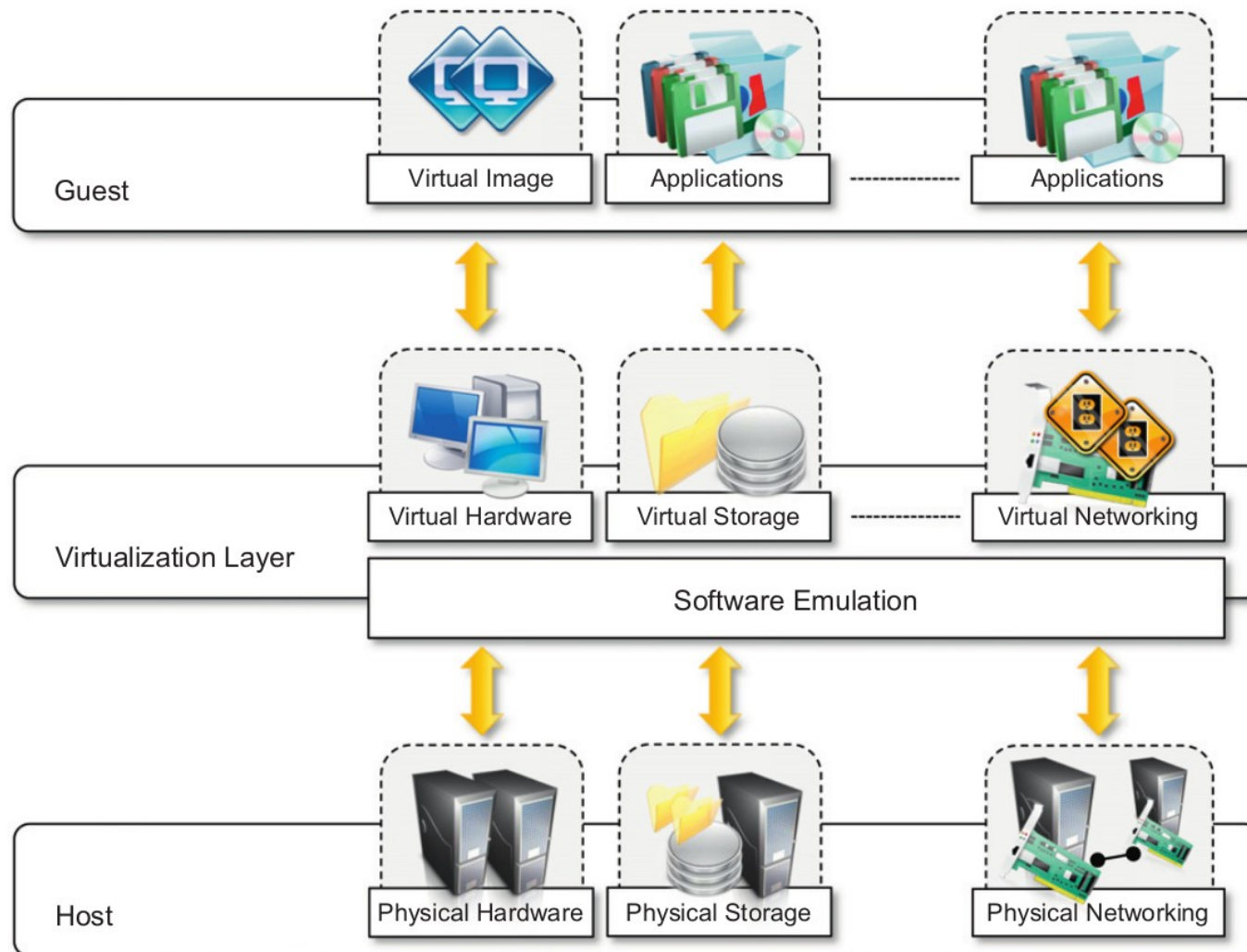
Tipos de Virtualización

- **Máquina** (Computer System Virtualization)
- Almacenamiento (Storage Virtualization) – Por ej., SANs.
- Red (Network Virtualization) – Externa (VLANs), Interna (consecuencia de la virtualización de sistema: NAT, TUN/TAP, etc.)
- Escritorio (Desktop Virtualization) - VNC, RDP, X Server.
- Servidor de Aplicaciones (Application Server Virtualization)



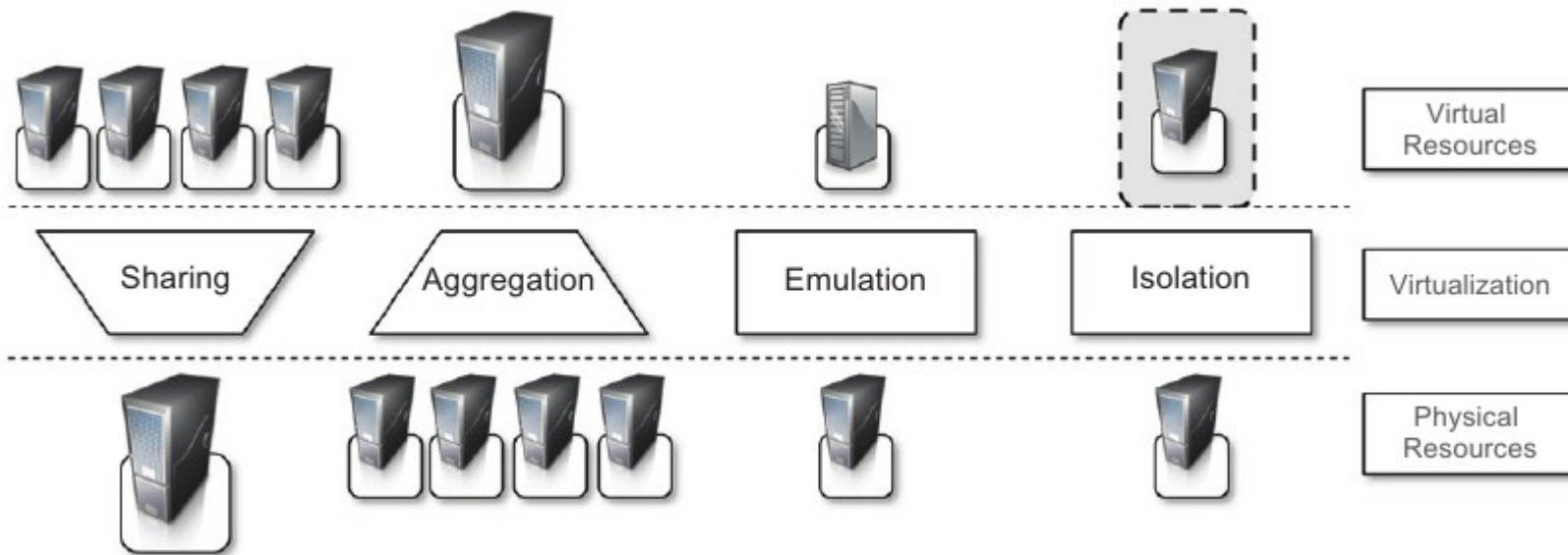
Virtualización

Componentes del Modelo de Referencia



Virtualización

Características implementables



- Compartimiento
- Agregación
- Emulación
- Aislamiento
- Performance Tuning → QoS → SLA



Técnicas de virtualización

Proveen soporte para la ejecución de programas, ya sean:

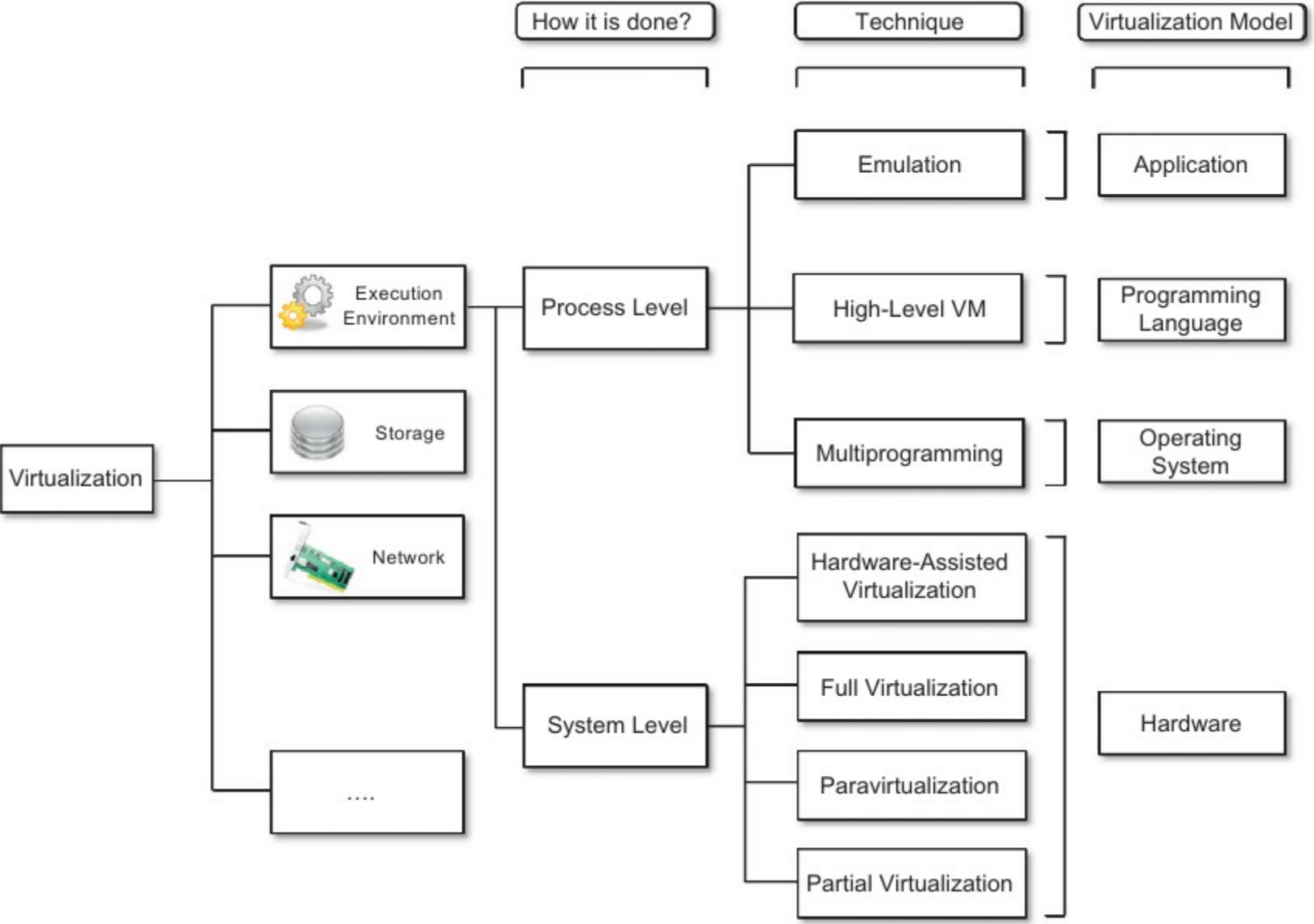
- S.O.
- ABI (Application Binary Interface)
- Programas compilados sobre un modelo de máquina abstracta
- Aplicaciones

Implementados:

- Directamente sobre el hardware
- Sobre el S.O.
- Mediante una aplicación
- Por librerías enlazadas estática o dinámicamente.



Virtualización



Virtualización de máquinas 'computadoras' (system virtualization)

- Introducida por IBM en sus mainframes en la década de 1960.
- Fines de 1990 VMWare en arquitecturas intel x386.
- En 2005 Intel y AMD incorporan a sus líneas de procesadores x86 extensiones específicas para dar soporte a virtualización (Intel VT, AMD-V).

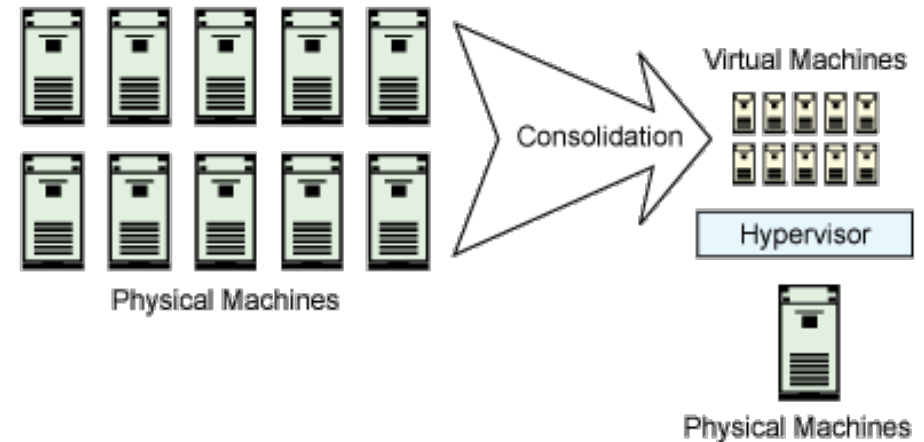


Virtualización

Virtualización

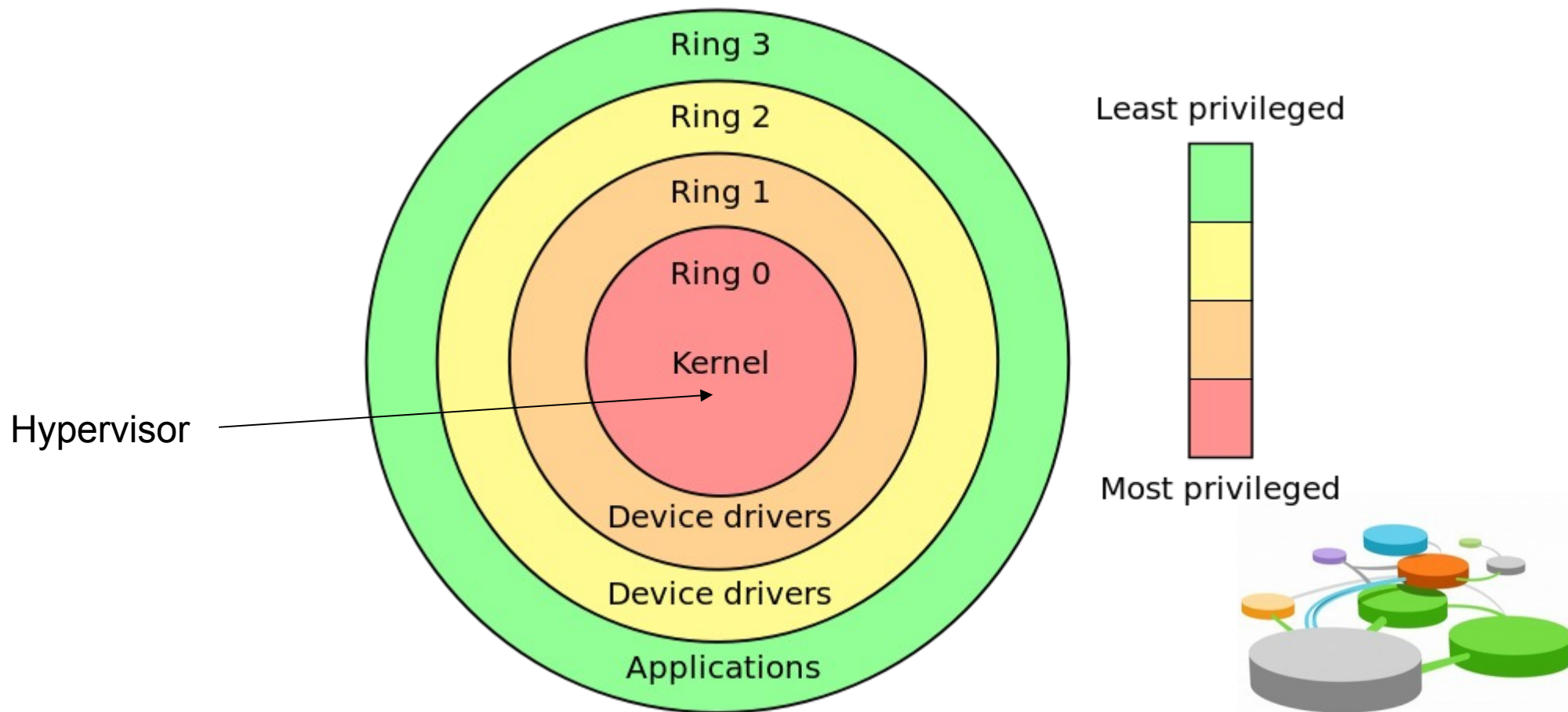
En alza en los últimos años debido a:

- Incremento en la capacidad de cómputo.
- Hardware y software infrautilizado.
- Necesidad de espacio adicional.
- Iniciativas “verdes”
- Costos administrativos



ISA - Soporte de hardware

Las plataformas de hardware generalmente poseen conjuntos de instrucciones dividido en clases que definen quién puede utilizarlas:



ISA - Soporte de hardware

Arquitecturas:

- IBM System/370, System/390 y mainframes ZSeries
- X86: Intel VT, AMD-V 2005
- POWER: ISA v 2.03 2006 (ISA v 2.06 2009 full virt for embedded)
- ARM (Cortex A15 2010)
- SPARC UltraSPARC Architecture 2005



Virtualización

Técnicas de Virtualización de Plataforma o Sistema:

1) Emulación

- Bucle Leer, Interpretar/Traducir, Ejecutar.

2) Full Virtualization

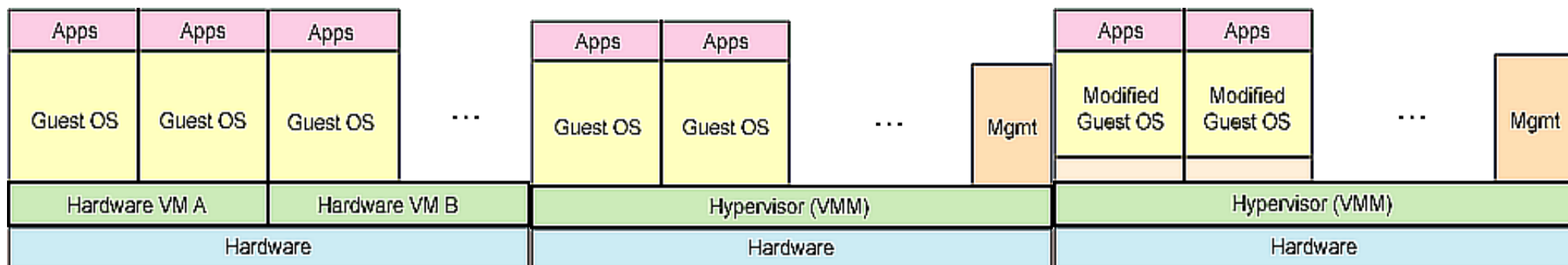
- “Traducción Binaria” realizada por un Hypervisor

3) Paravirtualización

- El Hypervisor permite compartir el hardware, el guest “lo sabe”

4) HVM – Hardware Asisted Virtualization

- El hardware provee soporte en el CPU para ejecutar un hypervisor



1) Emulación

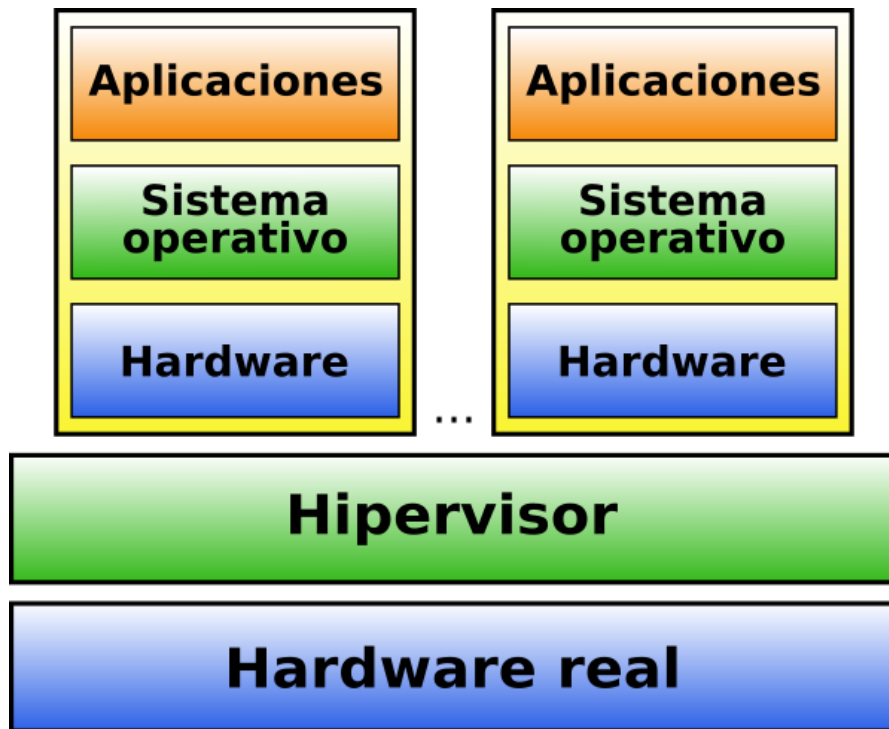
2,4) Full Virtualization, HVM

3) Paravirtualization

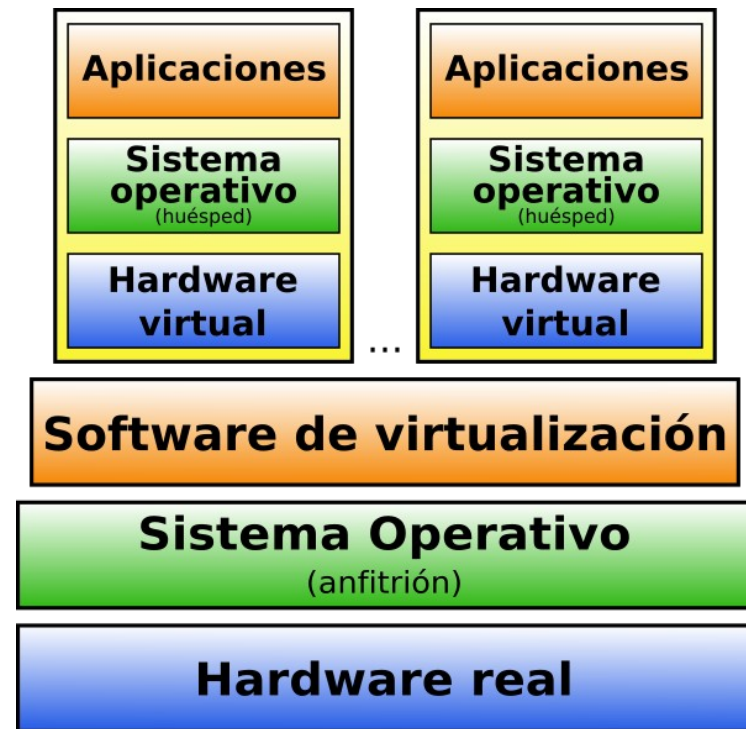
Virtualización

Tipos de Hipervisor

1. Tipo I - Bare-Metal, unhosted, native
2. Tipo II - Hosted



Tipo I - Bare-Metal



Tipo II - Hosted

Hypervisores Type I (native, bare-metal)

- Se ejecutan directamente sobre el hardware.
- Interactúan directamente con la interfaz ISA (instruction Set Architecture) del hardware.
- Deben implementar manejadores de dispositivos y capacidades de administración del hardware.
- Ejemplos: Xen, VMWare ESXi, Citrix, MS Hyper-V Server



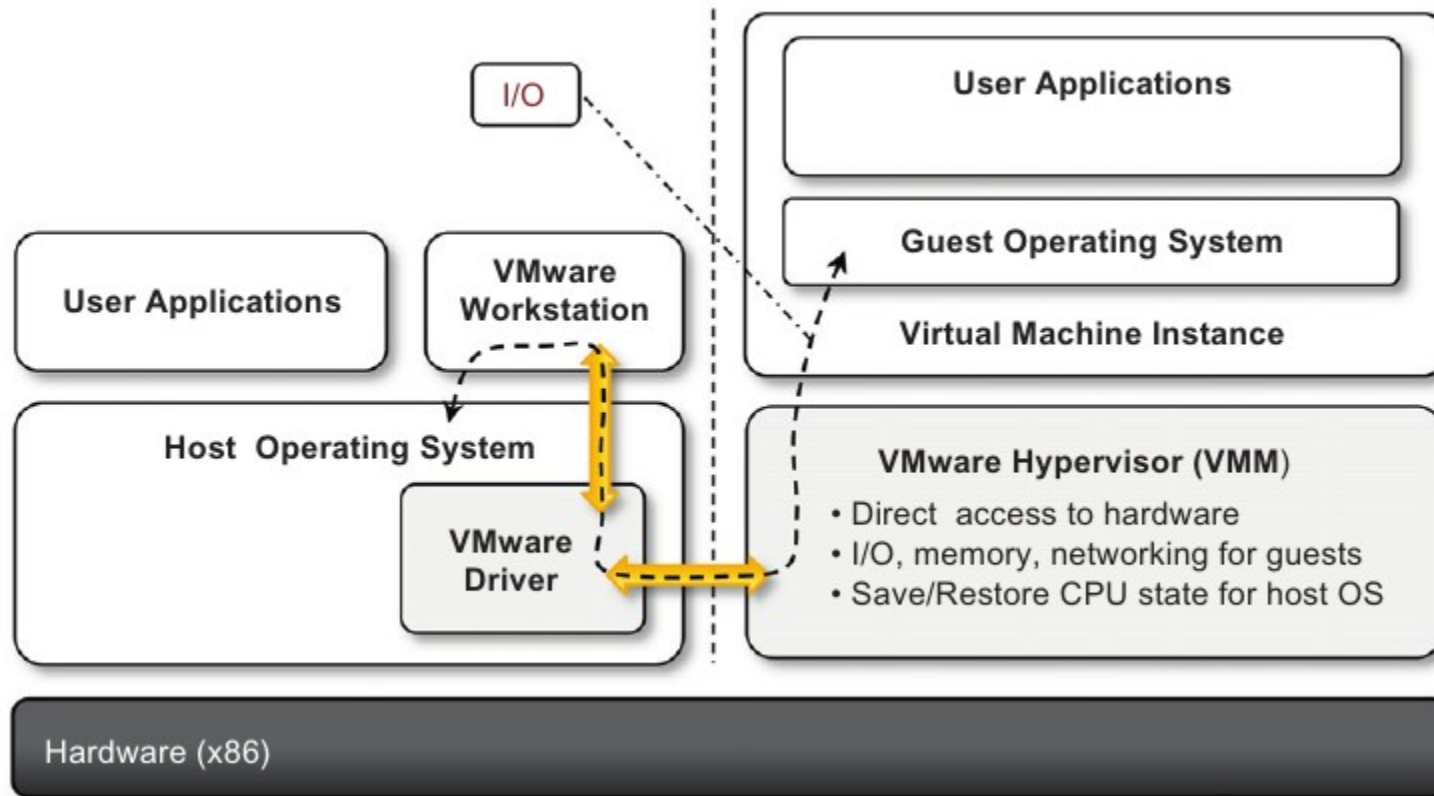
Hypervisores Type II (Hosted)

- Se ejecutan sobre un S.O. convencional.
- Aprovechan los manejadores de dispositivos y capacidades de administración del hardware.
- Interactúan con los S.O. a través de la ABI (Application Binary Interface) y emulan la ISA del hardware virtual para las VMs.
- Ejemplos: VMWare Workstation, Parallels, VirtualBox, VirtualPC, KVM*



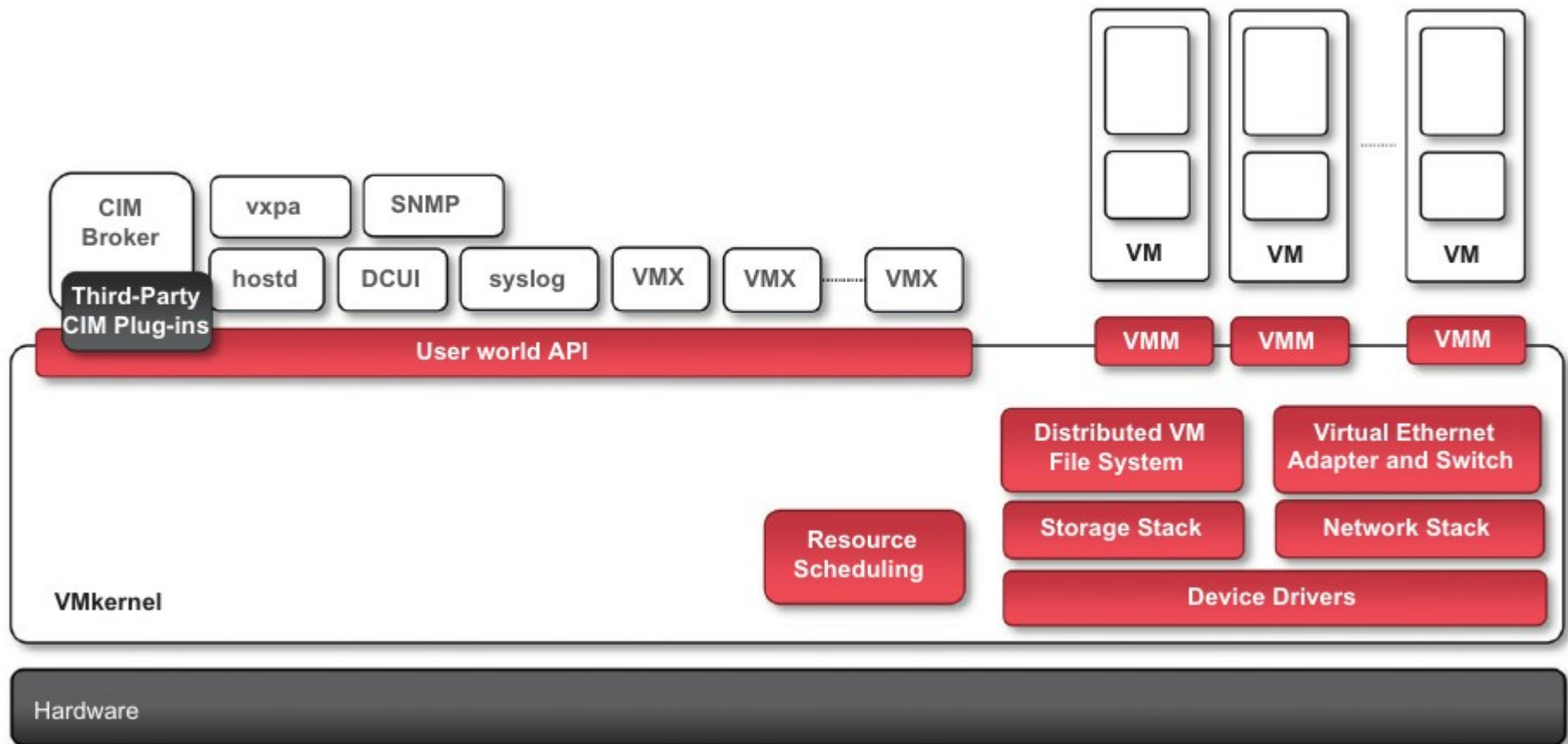
Virtualización

Arquitectura VMWare Workstation



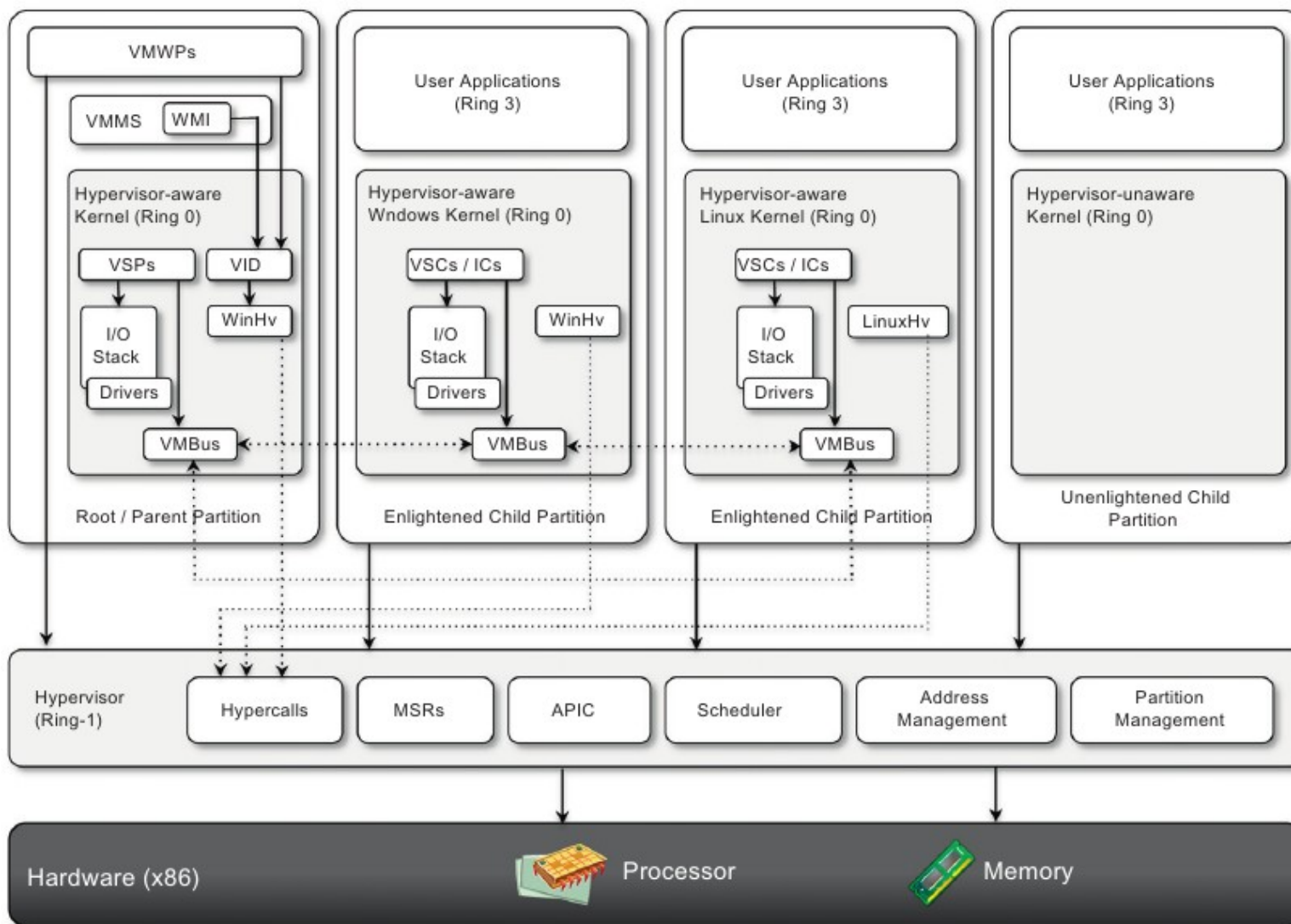
Virtualización

Arquitectura VMWare ESXi Server



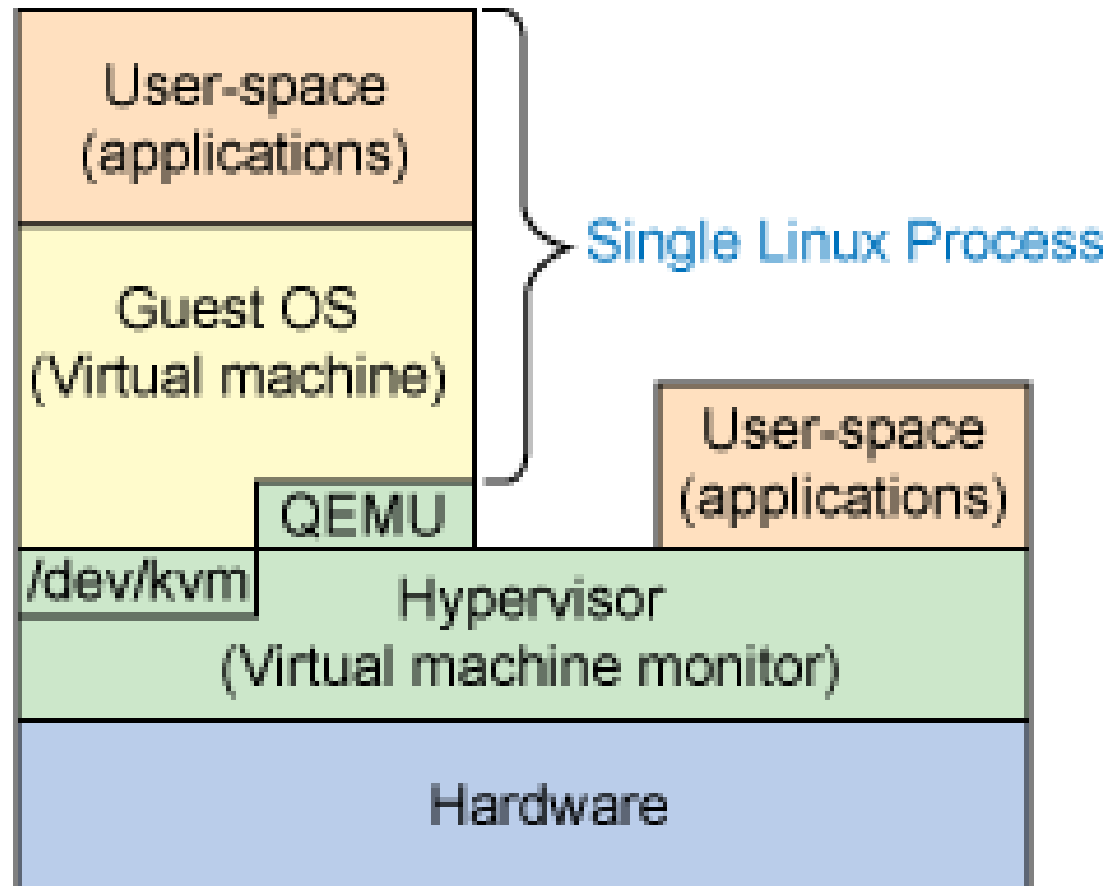
Virtualización

Arquitectura Microsoft Hyper-V



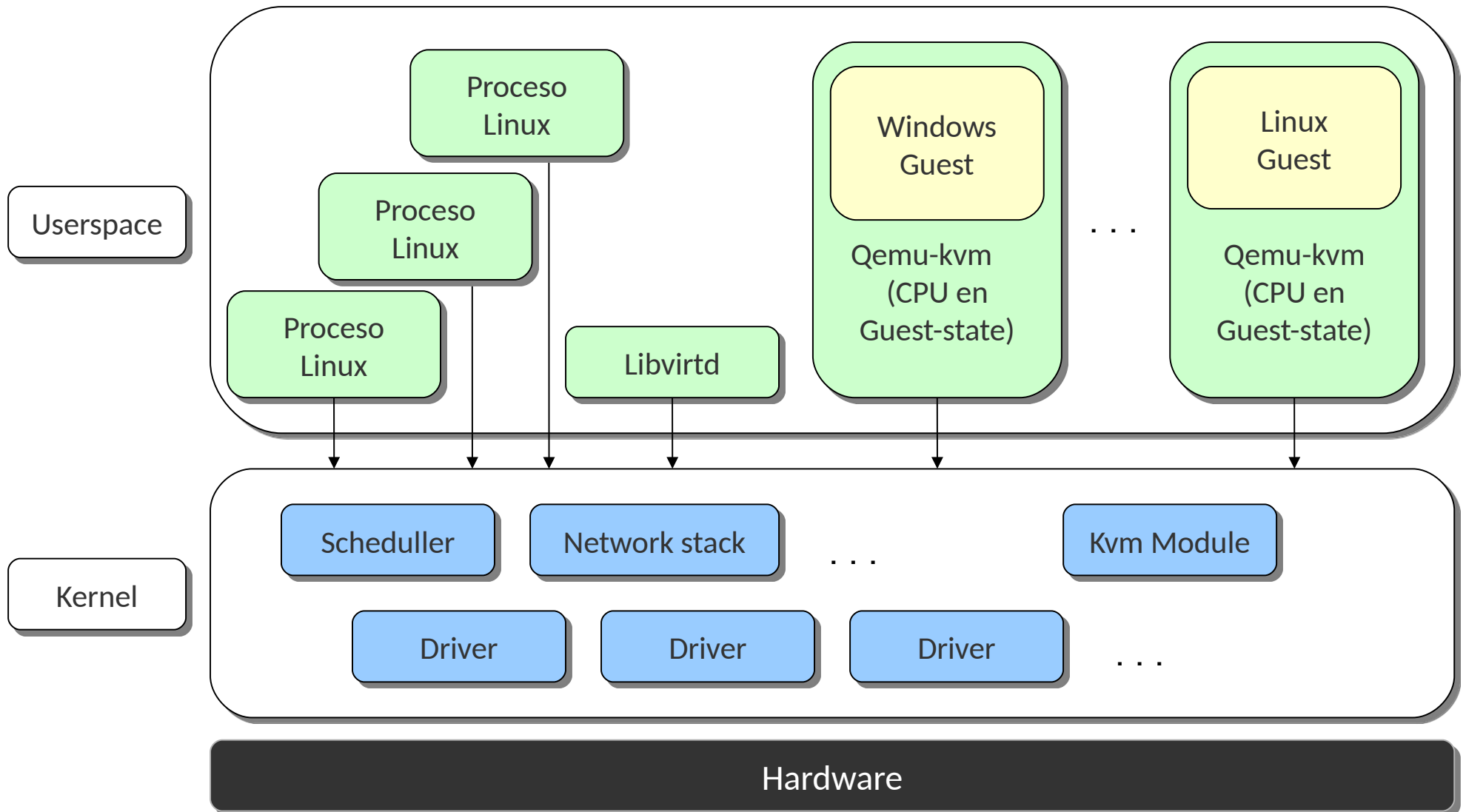
Virtualización

Arquitectura KVM



Virtualización

Arquitectura KVM



Virtualización

AWS EC2 Virtualization Types

	Bare-metal performance
	Near-metal performance
	Optimized performance
	Poor performance

Importance

Most → Least

CPU, Memory

Network I/O

Local Storage I/O

Remote Storage I/O

Interrupts, Timers

Motherboard, Boot

	#	Tech	Type	With						
Old ↓ New	1	VM	Fully Emulated		VS	VS	VS	VS	VS	VS
	2	VM	Xen PV 3.0	PV drivers	P	P	P	P	VS	VS
	3	VM	Xen HVM 3.0	PV drivers	VH	P	P	P	VS	VS
	4	VM	Xen HVM 4.0.1	PVHVM drivers	VH	P	P	P	P	VS
	5	VM	Xen AWS 2013	PVHVM + SR-IOV(net)	VH	VH	P	P	P	VS
	6	VM	Xen AWS 2017	PVHVM + SR-IOV(net, stor.)	VH	VH	VH	P	P	VS
	7	VM	AWS Nitro 2017		VH	VH	VH	VH	VH	VS
	8	HW	AWS Bare Metal 2017		H	H	H	H	H	H
			Bare Metal		H	H	H	H	H	H

VM: Virtual Machine. HW: Hardware.

VS: Virt. in software. VH: Virt. in hardware. P: Paravirt. Not all combinations shown.

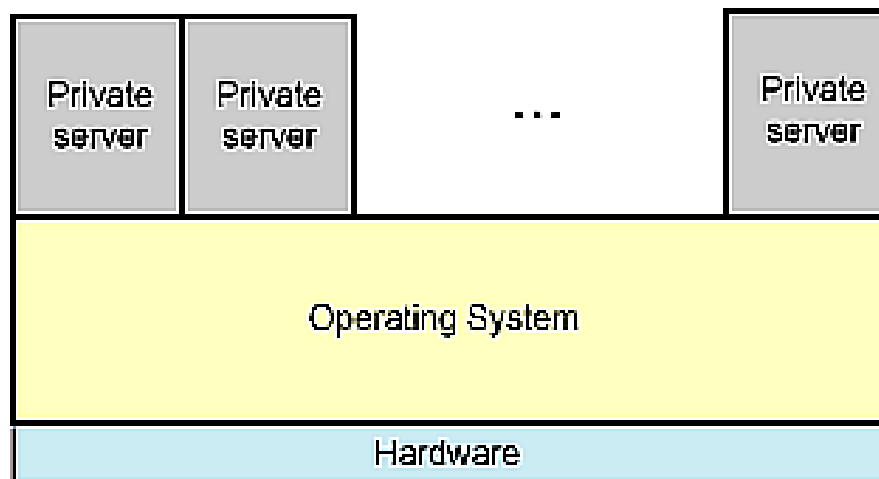
SR-IOV(net): ixgbe/ena driver. SR-IOV(storage): nvme driver.

Virtualización

Virtualización a nivel de S.O.

(Containers, Zones, Jails, Virtual Private Servers...)

- **Guests** comparten kernel con el anfitrión.
- **Orientados a aislar servicios, no plataformas completas**
- **Basada en aislación de “user-spaces” - Separación de espacios de nombres.**
 - FreeBSD Jails
 - HP-UX Containers
 - Solaris Zones
 - Linux Vserver
 - OpenVZ
 - Virtuozzo
 - LXC
 - Docker
 - VMWare ThinApp

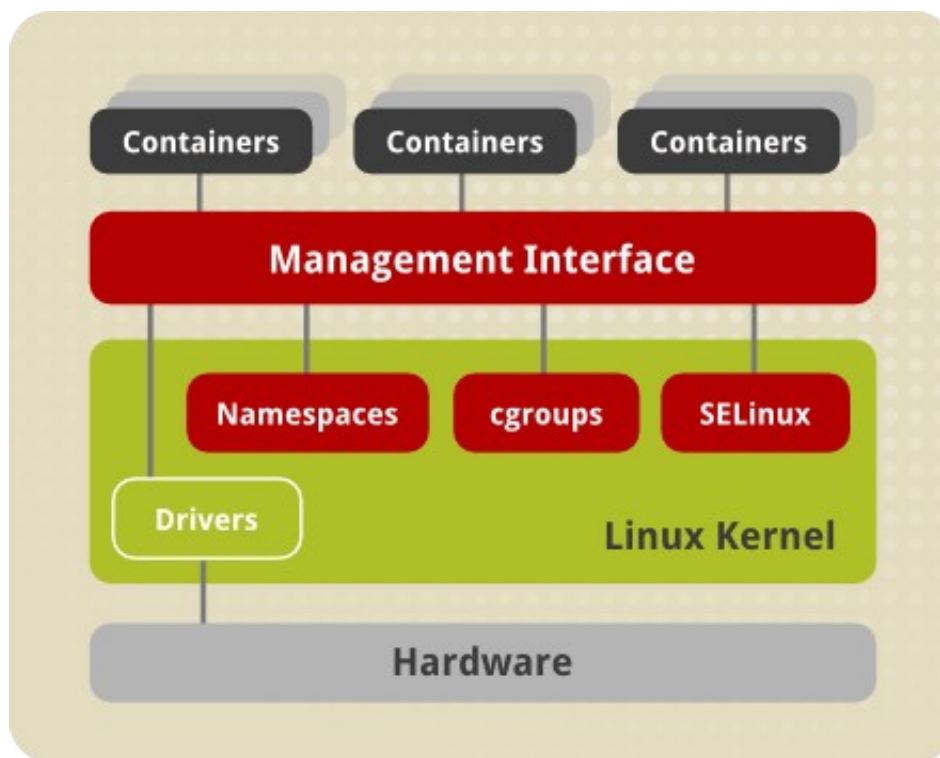


Virtualización

Virtualización a nivel de S.O. Linux Containers

- Namespaces:

- Mount
- UTS
- IPC
- Process
- Network
- User



Propiedades de las VM

"Formal Requirements for Virtualizable Third Generation Architectures"

Popek, G. J.; Goldberg, R. P. (1974)

- **The efficiency property.** All innocuous instructions are executed by the hardware directly, with no intervention at all on the part of the control program.
- **The resource control property.** It must be impossible for that arbitrary program to affect the system resources, i.e. memory, available to it; the allocator of the control program is to be invoked upon any attempt.
- **The equivalence property.** Any program K executing with a control program resident, with two possible exceptions, performs in a manner indistinguishable from the case when the control program did not exist and K had whatever freedom of access to privileged instructions that the programmer had intended.



Propiedades de las VM

"Formal Requirements for Virtualizable Third Generation Architectures"

Popek, G. J.; Goldberg, R. P. (1974)

● Instrucciones privilegiadas (Privileged instructions)

- Generan un trap si CPU en modo usuario, no en supervisor.

● Instrucciones sensibles (Sensitive instructions)

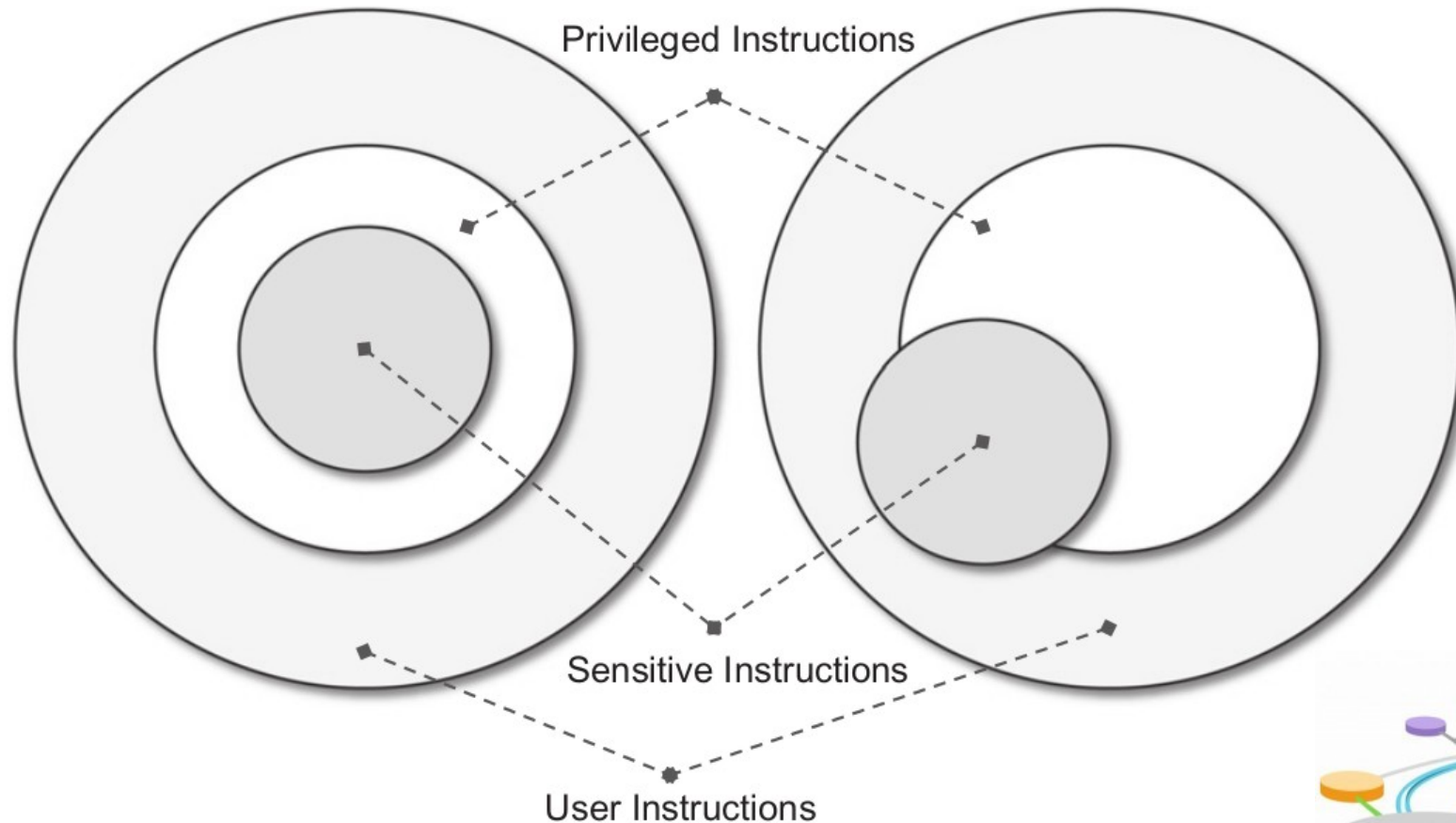
- Control sensitive: Cambian la configuración/recursos del sistema (por ej., memoria)
- Behavior sensitive: Dependen específicamente del hw para el que fue hecho



Propiedades de las VM

"Formal Requirements for Virtualizable Third Generation Architectures"

Popek, G. J.; Goldberg, R. P. (1974)



¿Cuál CPU es virtualizable?



Propiedades de las VM

"Formal Requirements for Virtualizable Third Generation Architectures"

Popek, G. J.; Goldberg, R. P. (1974)

- **THEOREM 1:** For any conventional third generation computer, a virtual machine monitor may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.
- **THEOREM 2:** A conventional third generation computer is recursively virtualizable if it is: (a) virtualizable, and (b) a VMM without any timing dependencies can be constructed for it.
- **THEOREM 3:** A hybrid virtual machine monitor may be constructed for any conventional third generation machine in which the set of user sensitive instructions are a subset of the set of privileged instructions.

Referencias

Mastering Cloud Computing, Foundations and Applications Programming. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi. Chapter 3: Virtualization. Morgan Kaufmann. 2013.

Real World Multicore Embedded Systems. Edited by: Bryon Moyer. Chapter 7: System Virtualization in Multicore Systems. David Kleidermacher. Newnes. 2013.

Cloud Computing, Theory and Practice. Dan C. Marinescu. Chapter 5 – Cloud Resource Virtualization. Morgan Kaufmann. 2013.

Formal requirements for virtualizable third generation architectures. Popek, G. J.; Goldberg, R. P. . Communications of the ACM. 1974

Referencias

Virtual Linux

<https://web.archive.org/web/20120214062442/http://www.ibm.com/developerworks/linux/library/l-linuxvirt/>

Anatomy of a Linux hypervisor

<https://www.ibm.com/developerworks/linux/library/l-hypervisor/>

Virtio: An I/O virtualization framework for Linux

<https://www.ibm.com/developerworks/linux/library/l-virtio/>

Linux virtualization and PCI passthrough

<https://www.ibm.com/developerworks/linux/library/l-pci-passthrough/index.html>

Virtual networking in Linux

<https://www.ibm.com/developerworks/linux/library/l-virtual-networking/>

Anatomy of Security-Enhanced Linux (SELinux)

<https://www.ibm.com/developerworks/linux/library/l-selinux/>

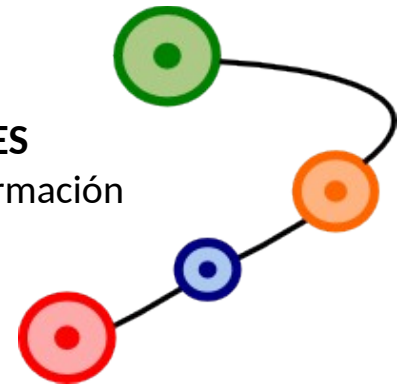
Understanding Full Virtualization, Paravirtualization, and Hardware Assist - VMware White Paper

https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware_paravirtualization.pdf



Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Software-Defined Networks

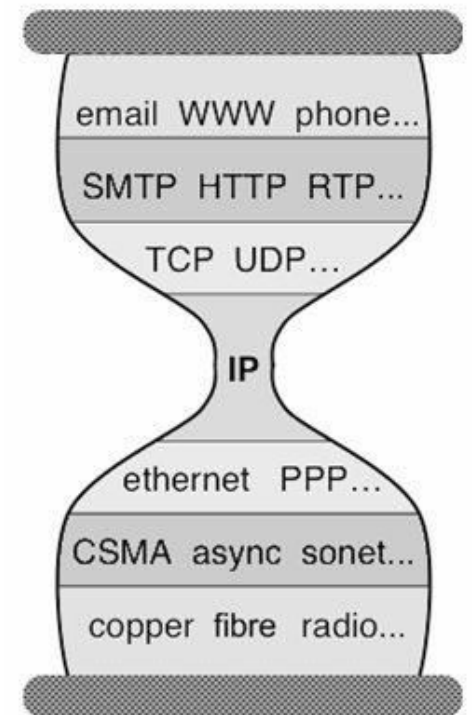
SDN

Redes Definidas por Software
-
Openflow

Software-Defined Networking

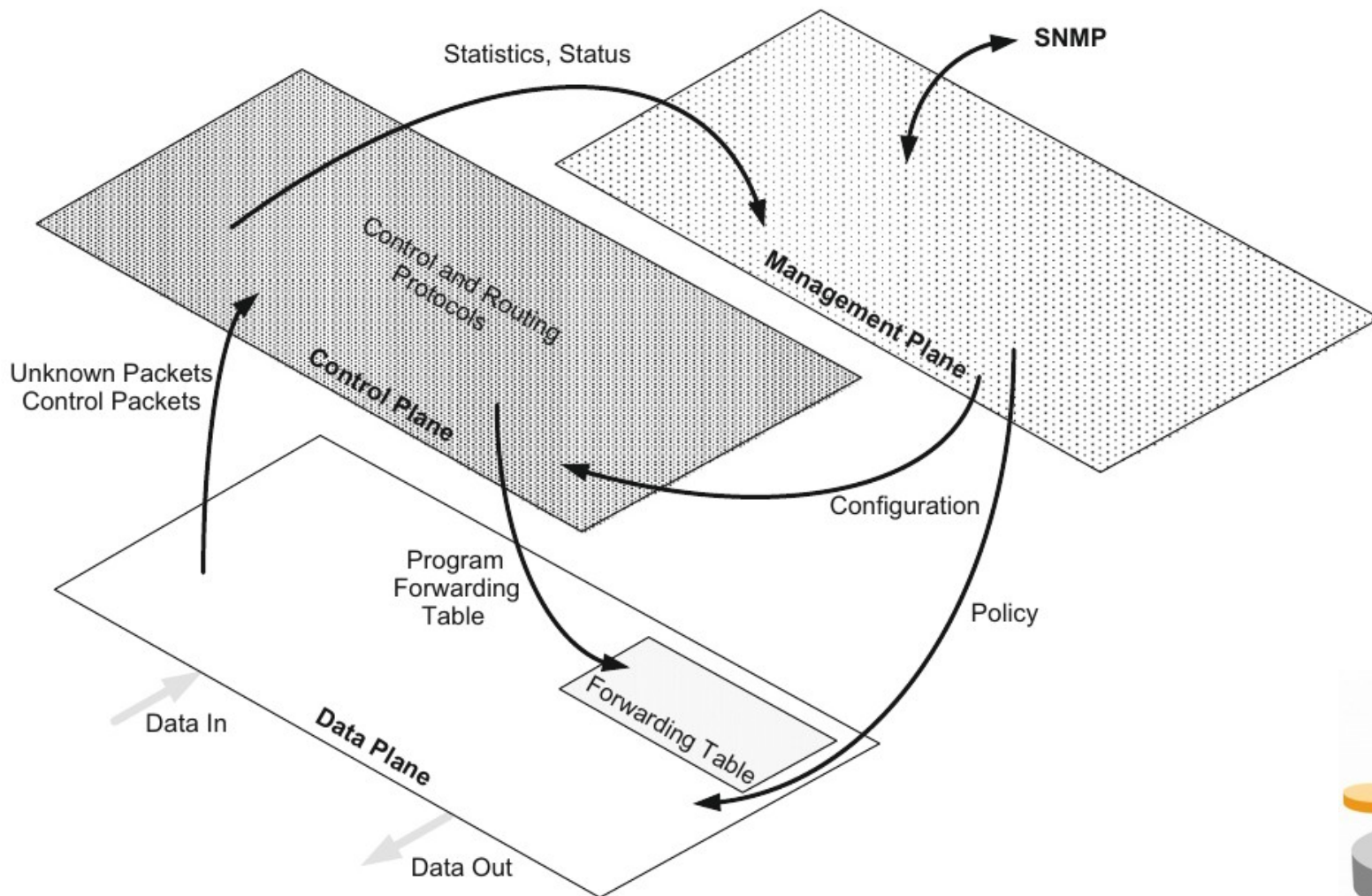
Modelo de red tradicional: El éxito de Internet

- Protocolos diseñados en capas, independientes:
 - Aplicaciones... creadas sobre...
 - Transporte fiable o no-fiable
 - Envío de paquetes “Best-effort”
 - Envío de tramas locales “Best-effort”
 - Transferencia física de bits
- Solución pensada en base a redes no-fiables para patrones de tráfico “conocidos”.
- Dispositivos de red autónomos.



Software-Defined Networking

Diferentes Planos: Control, Data y Management Plane



Software-Defined Networking

Planos tradicionales

Data Plane

- Manejo y reenvío de paquetes: switcheo, ruteo, filtrado, modificación...
- Tablas de reenvío: FIB, LFIB [1]
- Para performance se suele utilizar CAMs o TCAMs [2].
- Filters, Meters, Markers

[1] <https://mellowd.co.uk/ccie/?p=788>

[2] https://en.wikipedia.org/wiki/Content-addressable_memory



Software-Defined Networking

Planos tradicionales

Control Plane

- Configura el Data Plane indicándole como tratar los paquetes
- Acorde al hardware/software particular
- Algoritmos distribuidos (ruteo)
- Descubrimiento de topología, selección de rutas, failover...



Software-Defined Networking

Planos tradicionales

Management Plane

- Configuración del Control Plane (y posiblemente del Data Plane directamente)
 - Command Line
 - GUI
 - SNMP
 - Netconf ...



Software-Defined Networking

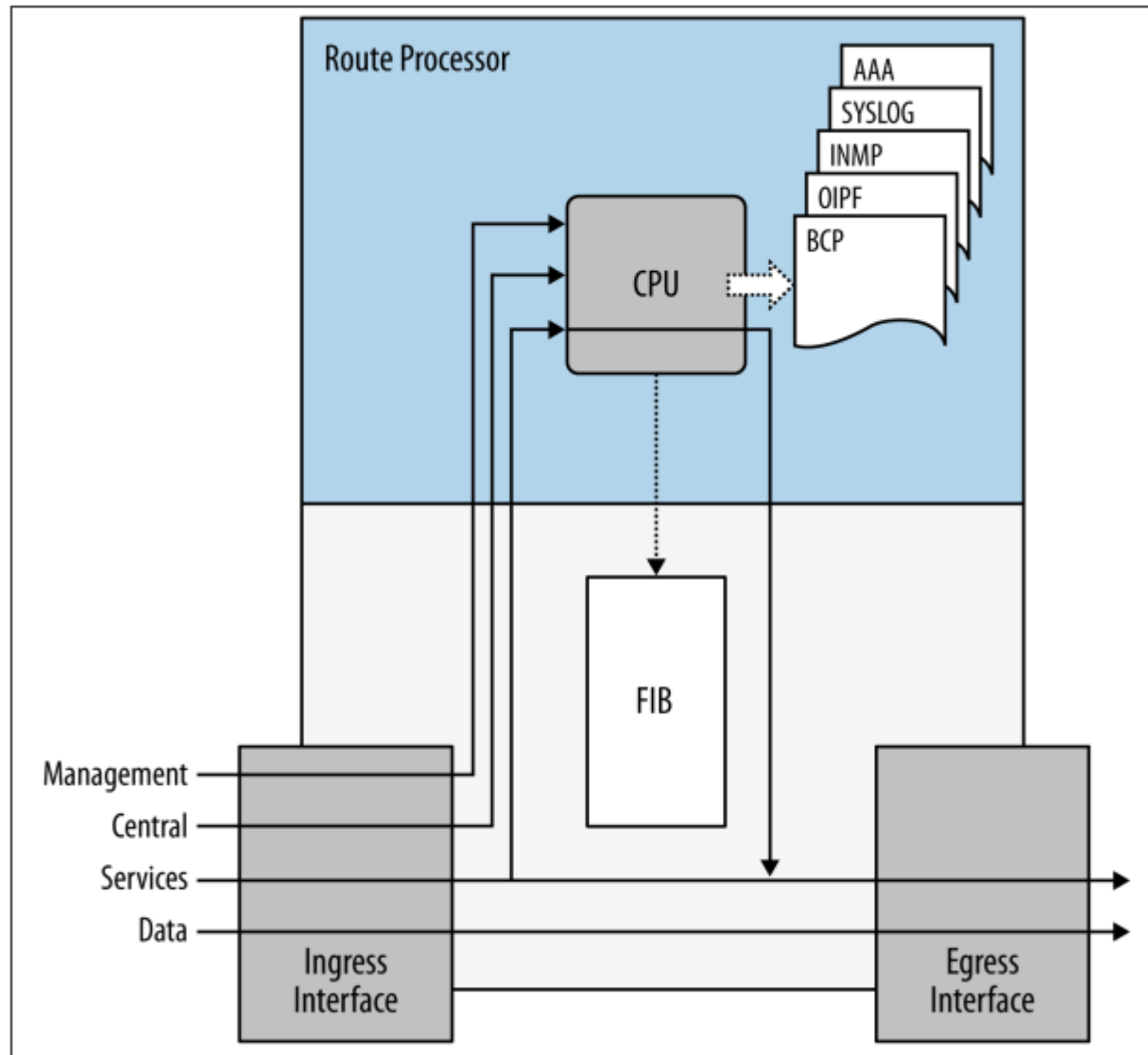


Figure 2-6. Control and data plane example implementation



Software-Defined Networking

Software-defined networks (SDN)

- Separar el Control Plane de los dispositivos y centralizarlo en un controlador...
- Administrar los servicios de red abstrayéndose del bajo nivel...
- Programar mediante una API la funcionalidad de la red...
- ¿Se usa? ¿Dónde?...



Software-Defined Networking

Software-Defined Networks (SDN):

“A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.”

(UIT-T Y.3300)

“A programmable networks approach that supports the separation of control and forwarding planes via standardized interfaces.”

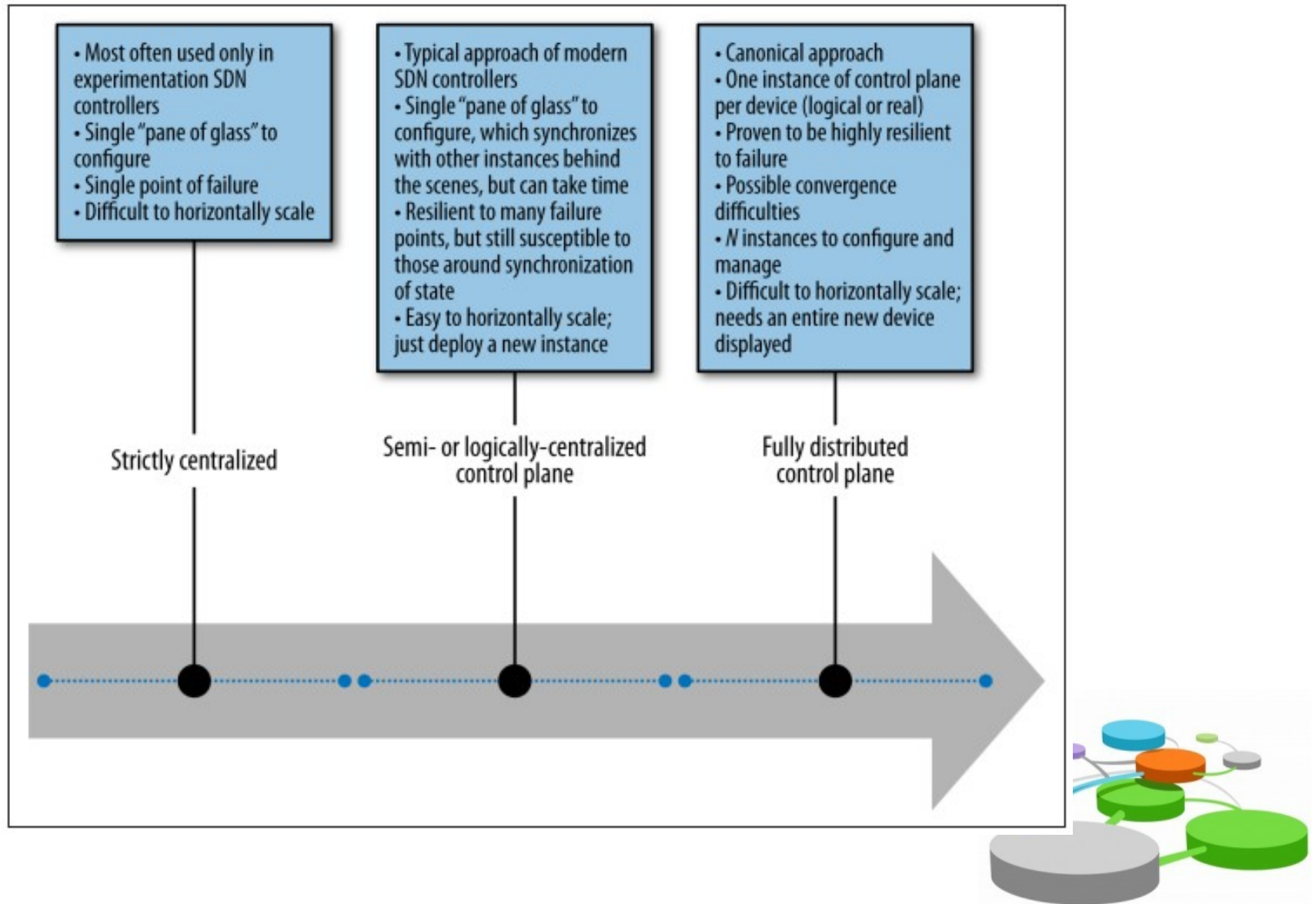
(IETF RFC 7426)

“The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.”

(ONF - Open Networking Foundation)



Software-Defined Networking



Software-Defined Networking

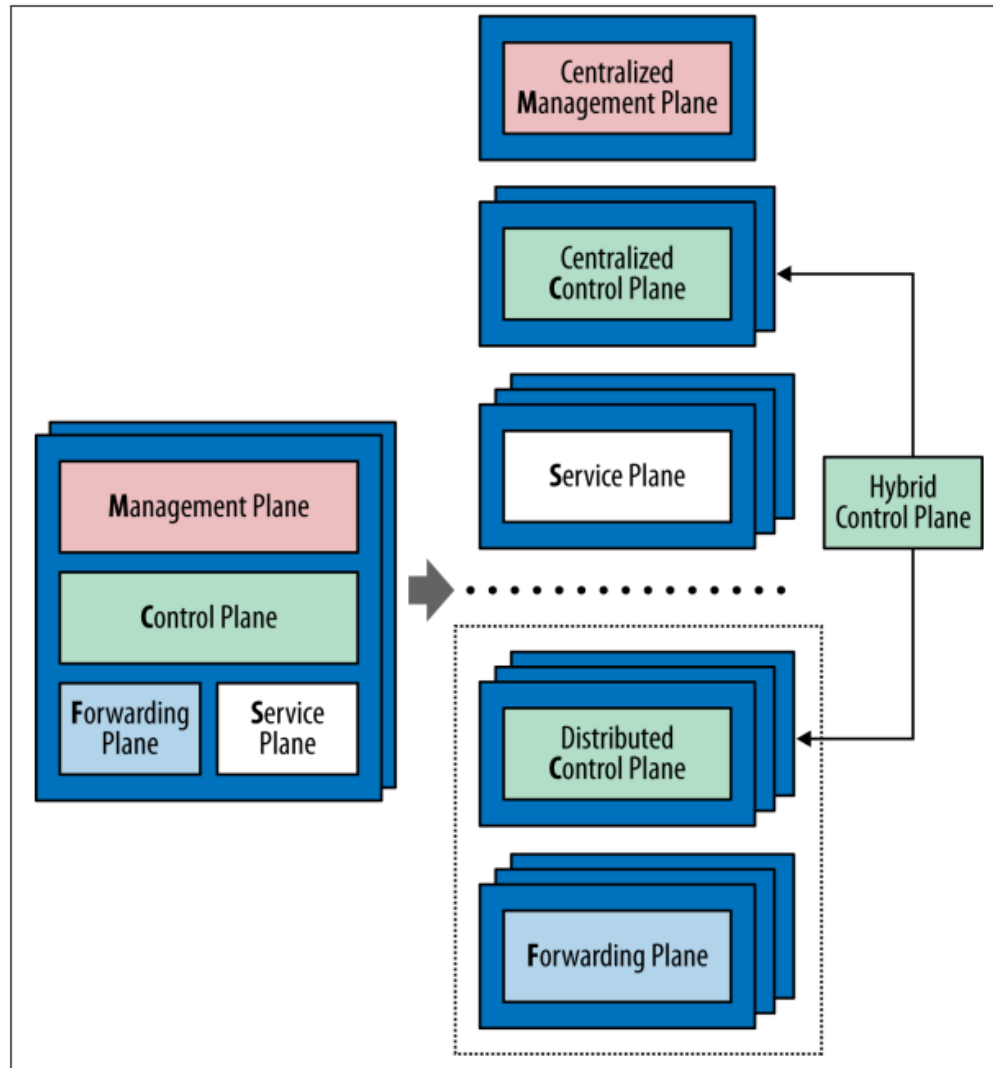


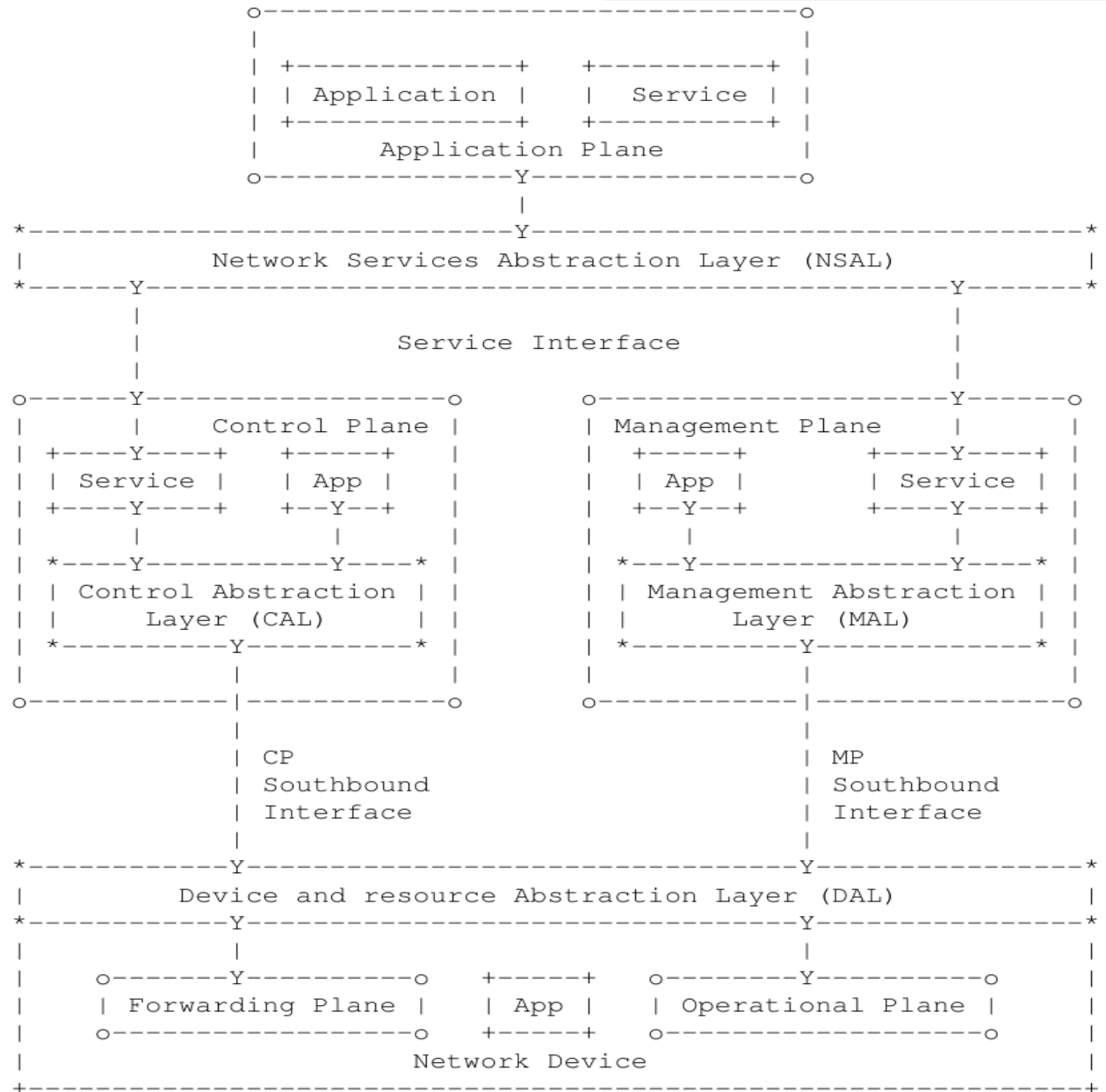
Figure 2-7. Separating the integrated management, control, service, and forwarding planes so that they can scale independently



Software-Defined Networking

Arquitectura SDN

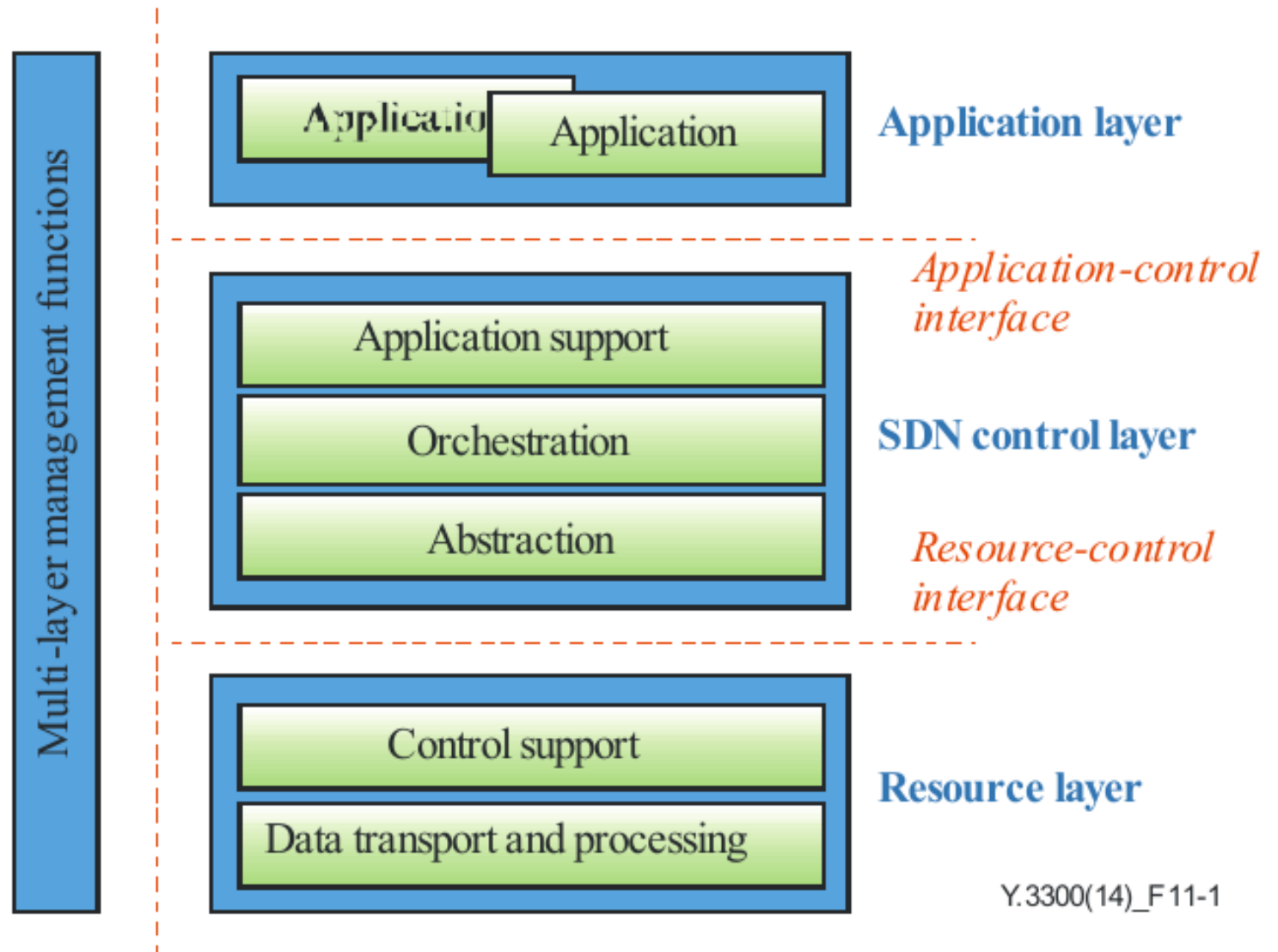
IETF - RFC 7426



Software-Defined Networking

Arquitectura
SDN

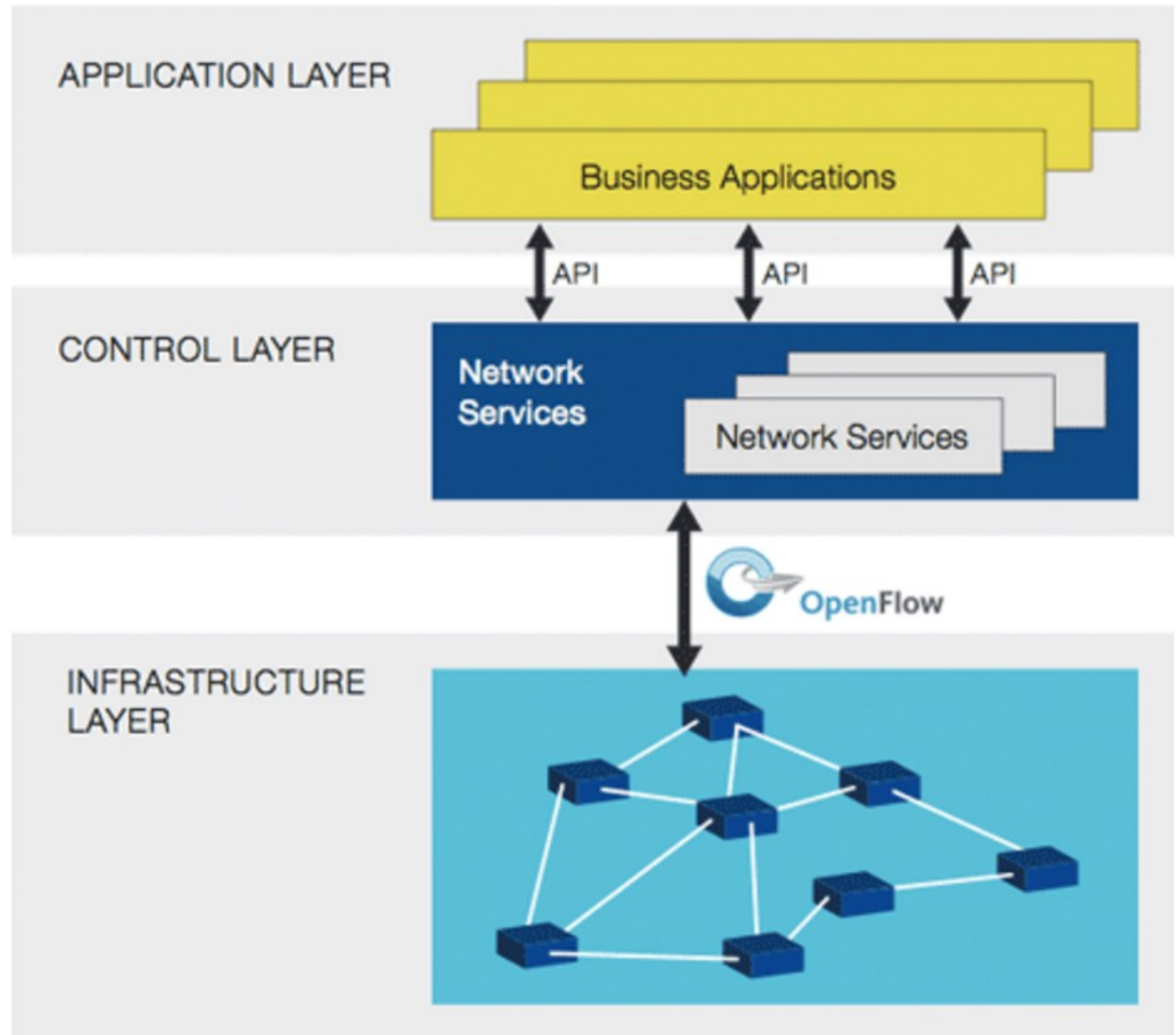
UIT-T Y.3300



Software-Defined Networking

Arquitectura
SDN

ONF



Software-Defined Networking

Actualmente diversos controladores/plataformas

- Open source:
 - OpenDaylight, Tungsten Fabric (antes OpenContrail), Floodlight, Onos, Ryu, Beacon, Nox/Pox
 - ONAP: merge de open source ECOMP (liberado por AT&T en 2017) y Open Orchestrator Project (OPEN-O)
- Proprietarios:
 - Brocade, Cisco, Big Switch Networks, Dell, HPE, Cumulus Network, Juniper, IBM, NEC, VMWare, Nuage , Pica8...
- Internos:
 - Google (Al menos 5 controladores: FirePath, Andromeda, B4, TE, BwE, uso de openflow), Microsoft ...



Software-Defined Networking

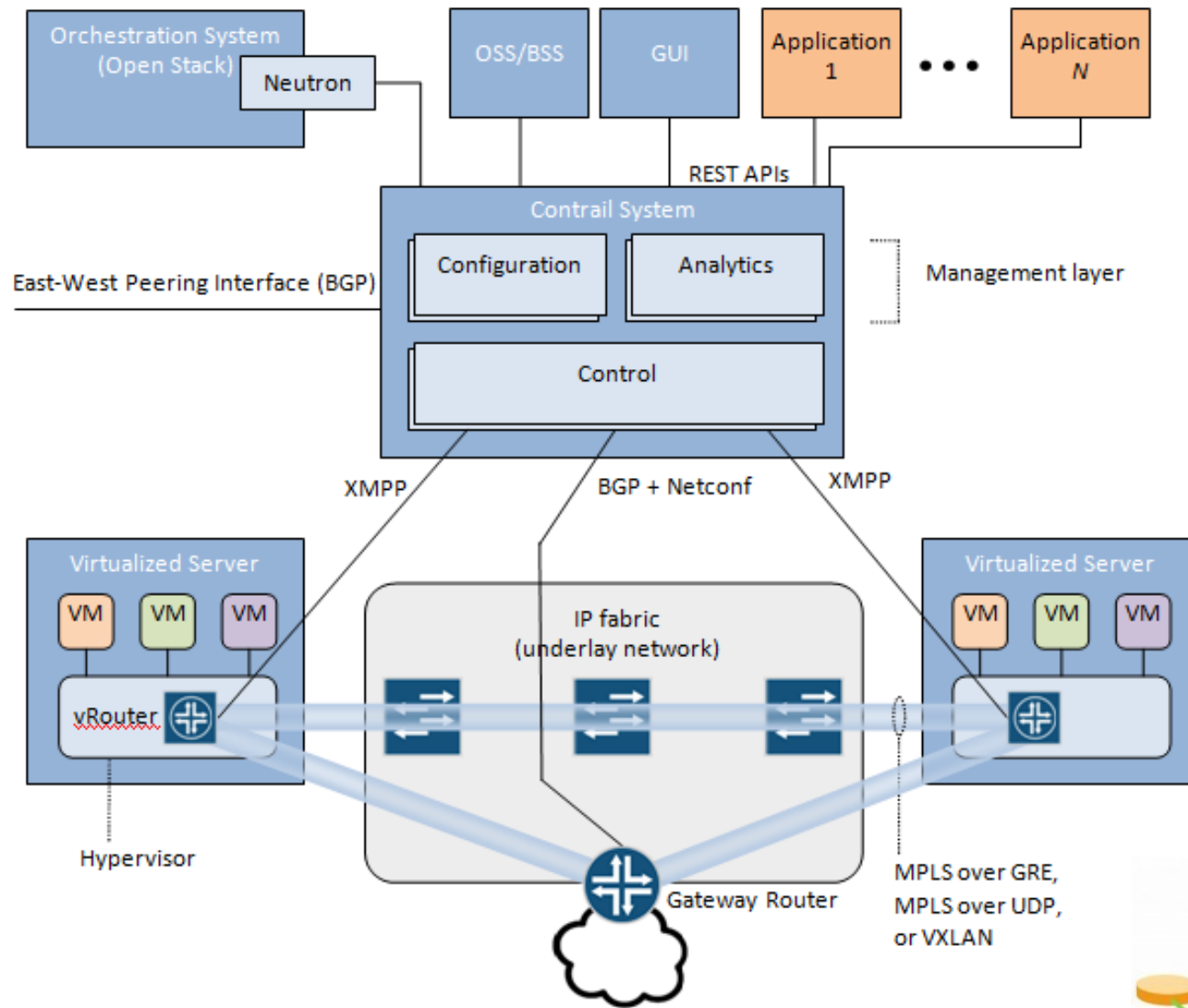


Figure 1: OpenContrail System Overview



Software-Defined Networking

SDN en Carriers Globales

(AT&T, Level3, Colt, Orange BS, SK Telecom, Telefónica ...)

- **Encuesta tiempo de despliegue de IHS (2018):**

- 78 % de encuestados dice haber implementado o implementarán SDN en 2018
- 9% en 2019.
- 4% en 2020 o luego.
- 9% En algún momento harán despliegue...

- **En AT&T:**

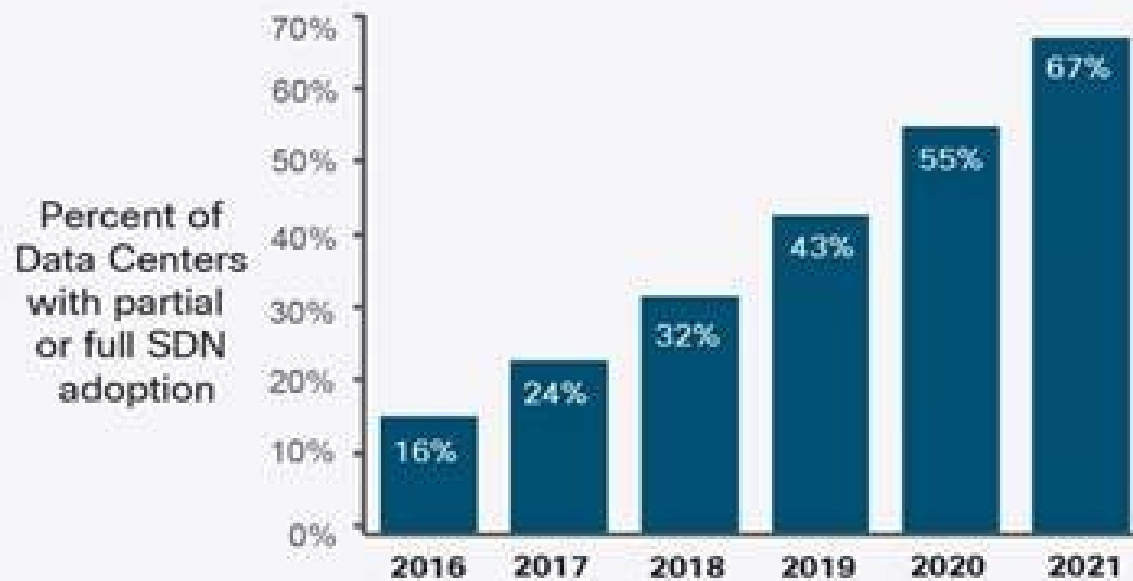
- Virtualización en 2015 5.7%, planeaban 30% para 2016 y 75% para 2020.
- Septiembre 2019 [1]: 75% del tráfico sobre túneles MPLS, que conectan
- los elementos de “core” de su red están bajo el control SDN.

[1] https://about.att.com/innovationblog/2019/09/delivering_sdn_promise.html



Software-Defined Networking

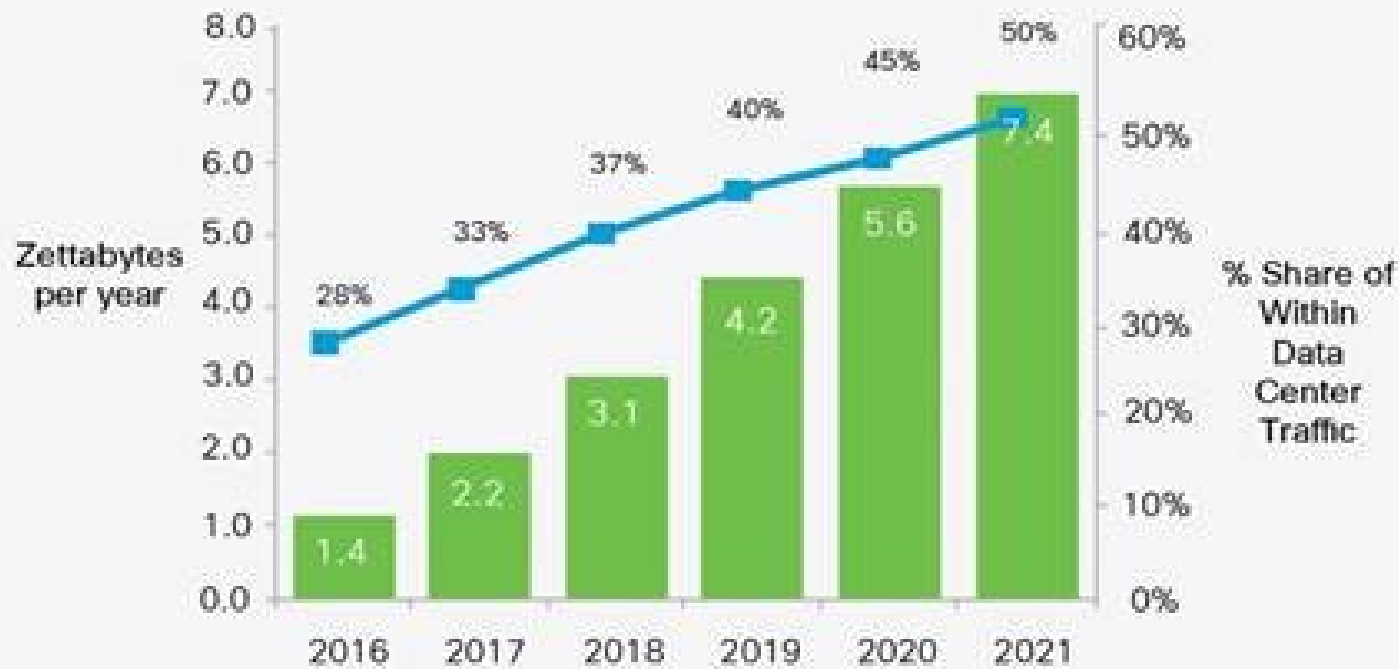
Tráfico SDN en Datacenters



Source: Cisco Global Cloud Index, 2016-2021.

Software-Defined Networking

Tráfico SDN en Datacenters



Source: Cisco Global Cloud Index, 2016-2021.

Software-Defined Networking

Propuesta OpenFlow:

- 2008: UC Berkeley, Washington, Princeton “OpenFlow: Enabling Innovation in Campus Networks”
- Aprovechar las funciones comunes ya implementadas en flow-tables.

OpenFlow Switch Specification (Open Network Foundation)

Version actual* 1.5.1 – Marzo 2015:

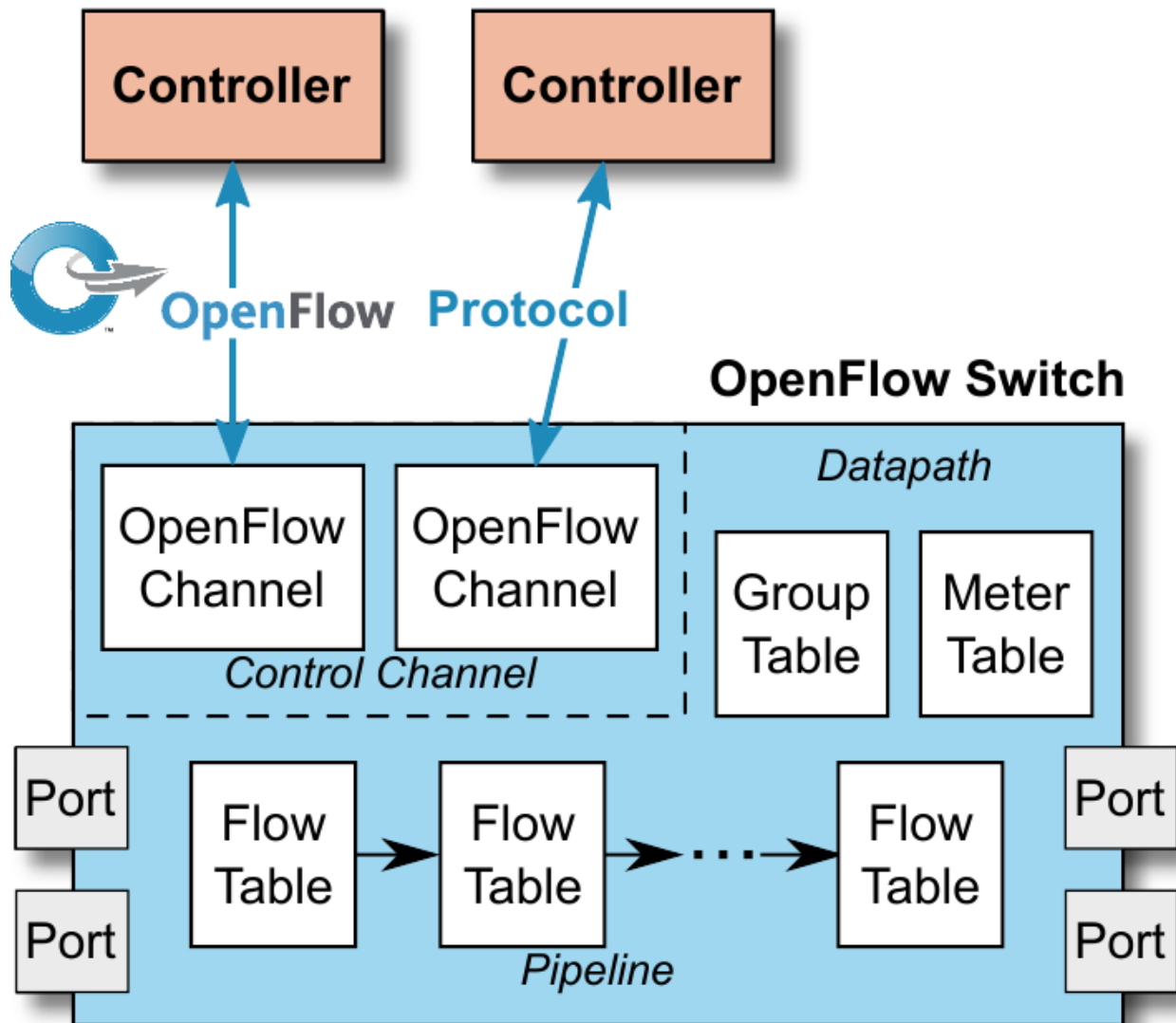
- Openflow Switch:
 - Flow-tables
 - Canal seguro de comunicación con el controlador
 - El protocolo OpenFlow (interfaz estándar para definir las flow-tables)
- Switchs dedicados o híbridos.

* Versión 1.6 (2016-09) Members only



Software-Defined Networking

OpenFlow Switch - Componentes principales



Software-Defined Networking

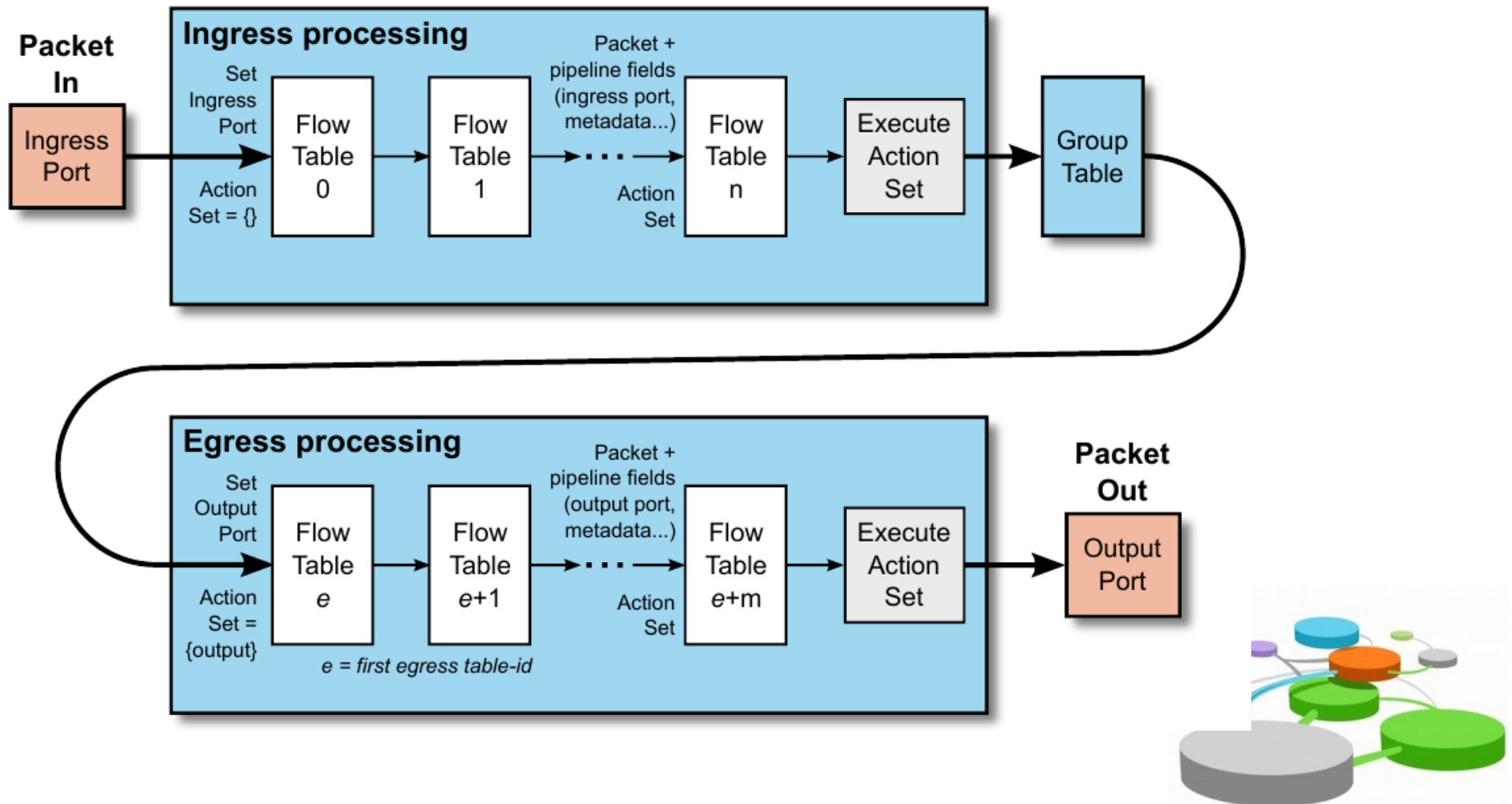
OpenFlow Switch Specification

- El controlador agrega, modifica o borra entradas de las flow tables utilizando el protocolo OpenFlow switch protocol.
- Las actualizaciones pueden ser proactivas o reactivas.
- Cada entrada en la Flow Table contiene:
 - Campos de matcheo
 - Acciones (conjunto de instrucciones)
 - Contadores



Software-Defined Networking

OpenFlow Pipeline Processing



Software-Defined Networking

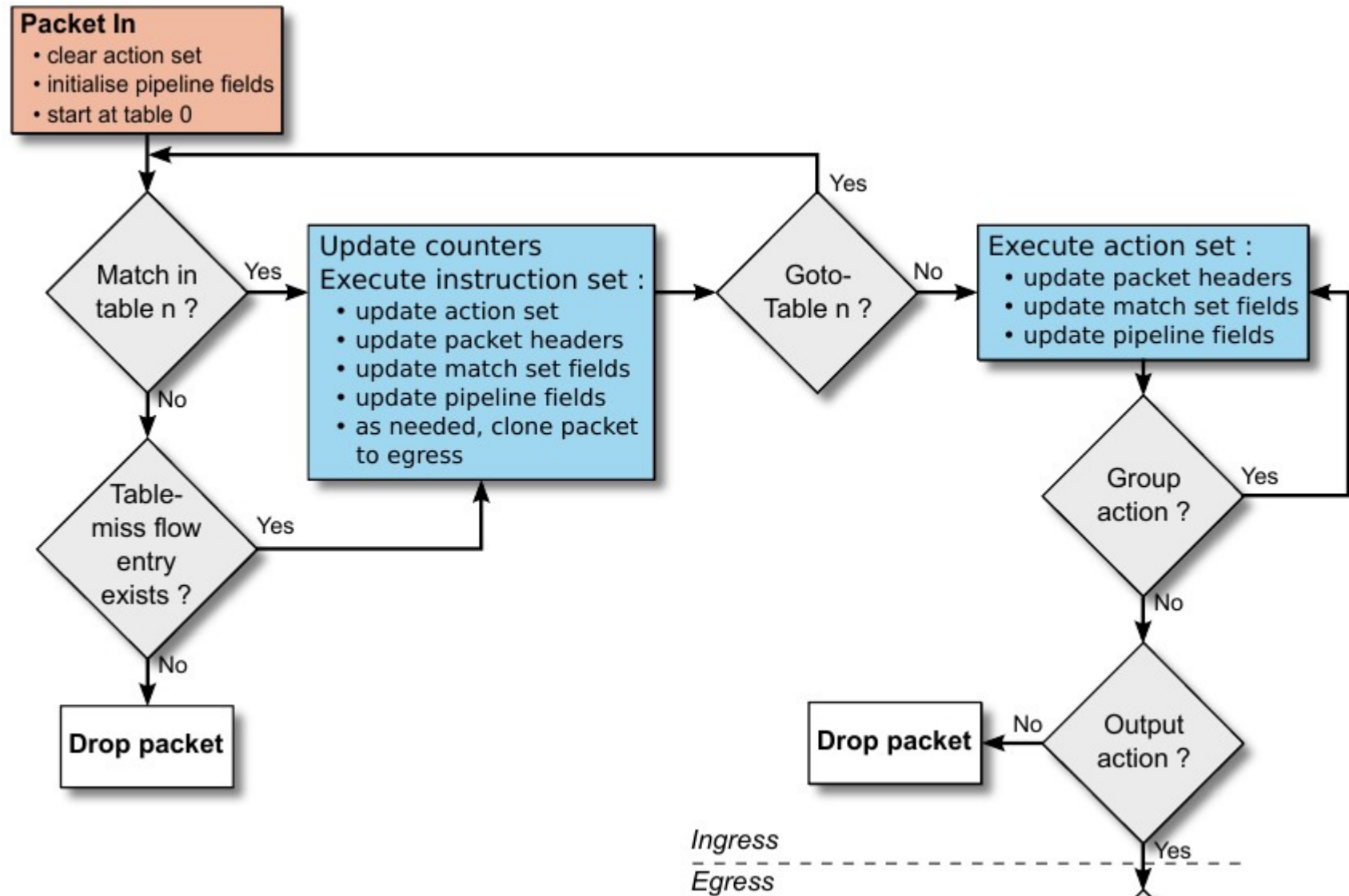
OpenFlow Switch Specification Flow Table Entry

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags
--------------	----------	----------	--------------	----------	--------	-------

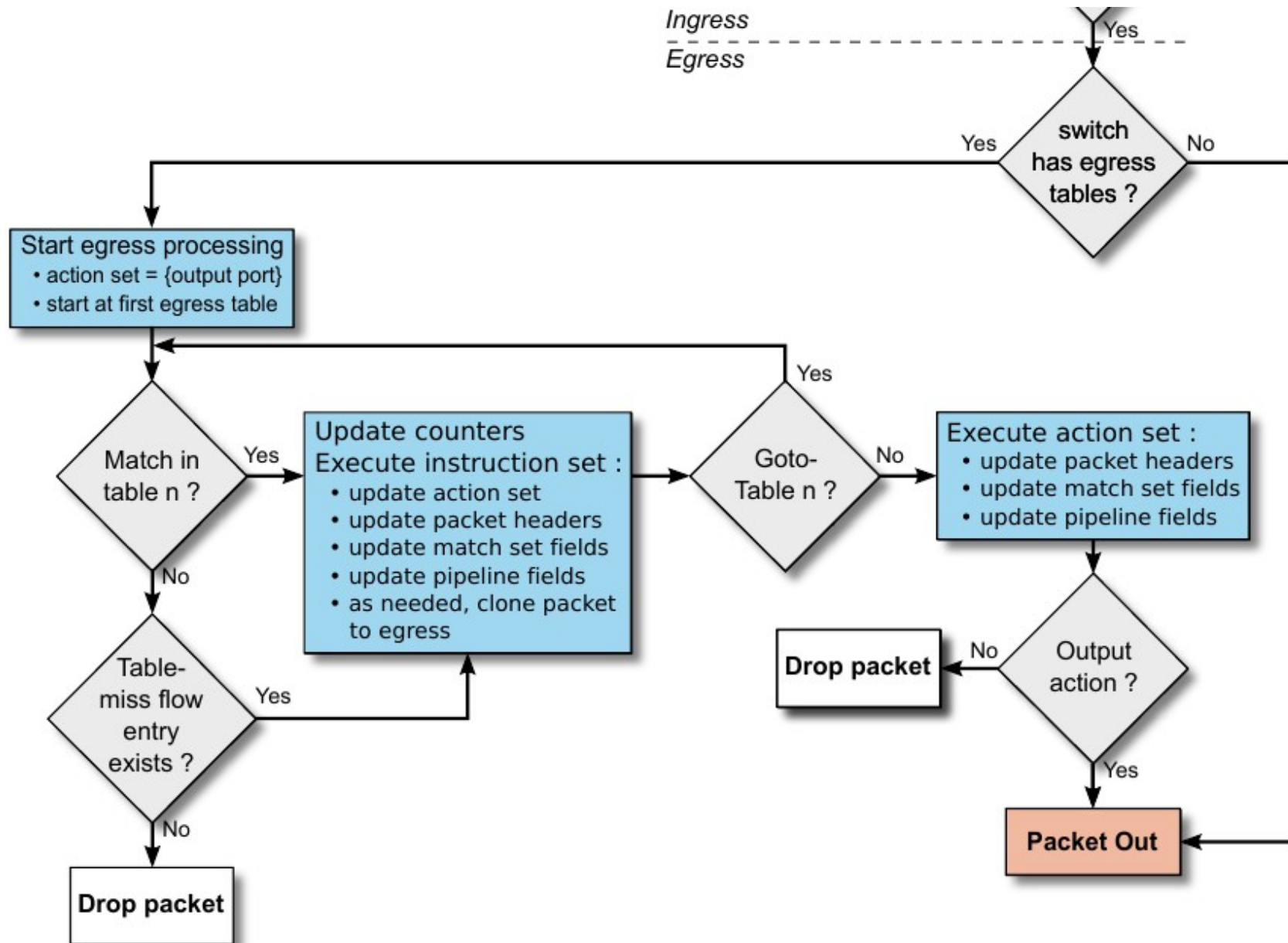
- **match fields:** Campos de “matcheo”: Puerto de ingreso y encabezados del paquete, y opcionalmente otros datos del procesamiento tal como metadatos agregados por tablas anteriores.
- **priority:** precedencia de la entrada.
- **counters:** actualizados cuando matchean paquetes.
- **instructions:** modifican el conjunto de acciones (action set) o el procesamiento en tubería
- **timeouts:** Tiempo máximo de la entrada o tiempo máximo sin actividad.
- **cookie:** Valor colocado por el controlador. Puede ser utilizado para filtrar entradas afectadas por estadísticas, requerimientos de modificación o borrado. No utilizado en el procesamiento de paquetes.
- **flags:** Determinan como son administradas las entradas.



Software-Defined Networking



Software-Defined Networking



Software-Defined Networking

Ports

- Físicos
- Lógicos (tuneles, loopback, grupos de agregación de enlaces)
- Reservados:
 - All
 - Controller
 - Table
 - IN_PORT
 - ANY
 - UNSET
 - Local (Opcional)
 - Normal (Opcional)
 - Flood (Opcional)



Software-Defined Networking

Instrucciones

- **Requeridas:**
 - **Write-Actions:** Agrega acción/es al conjunto de acciones actual.
 - **Goto-Table:** Indica la próxima tabla a procesar.
 - **Clear-Actions:** Vaciar el Action Set.
- **Opcionales:**
 - **Apply-Actions:** Aplica las acciones especificadas inmediatamente sin afectar el action set.
 - **Stat-Trigger:** Generar un evento al controlador si alguna estadística del flujo atraviesa un umbral.
 - **Write-Metadata:** Modifica el campo de metadatos.



Software-Defined Networking

Acciones

- Requeridas:
 - Output *port_no.*: Forwardea el paquete por el puerto Open Flow especificado donde comienza el egress processing.
 - Group *group_id.*: Procesa el paquete por el grupo especificado.
 - Drop.
- Opcionales:
 - Set-Queue *queue_id.*: Utilizado para determinar en cuál cola del puerto se encolará el paquete.
 - Meter *meter_id.*: Dirigir paquete al “meter” especificado.



Software-Defined Networking

Acciones (2)

- Opcionales:
 - Push-Tag/Pop-Tag *ethertype*: Push o Pop de encabezados VLAN, MPLS o PBB.
 - Set-Field *field_type value*: Modifica valor de campo de encabezado del paquete.
 - Copy-Field *src_field_type dst_field_type*: Copiar valores entre headers o campos del pipeline.
 - Change-TTL *tvl*: Modifica IPv4 TTL, IPv6 Hop Limit o MPLS TTL.



Software-Defined Networking

OpenFlow Switch Specification Match Fields v1.3.4

Switch input port.
Switch physical input port.
Metadata passed between tables.
Ethernet destination address.
Ethernet source address.
Ethernet frame type.
VLAN id.
VLAN priority.
IP DSCP (6 bits in ToS field).
IP ECN (2 bits in ToS field).
IP protocol.
IPv4 source address.
IPv4 destination address.
TCP source port.
TCP destination port.
UDP source port.
UDP destination port.
SCTP source port.
SCTP destination port.
ICMP type.
ICMP code.
ARP opcode.
ARP source IPv4 address.
ARP target IPv4 address.
ARP source hardware address.
ARP target hardware address.
IPv6 source address.
IPv6 destination address.
IPv6 Flow Label.
ICMPv6 type.
ICMPv6 code.
Target address for ND.
Source link-layer for ND.
Target link-layer for ND.
MPLS label.
MPLS TC.
MPLS BoS bit.
PBB I-SID.
Logical Port Metadata.
IPv6 Extension Header pseudo-field



Software-Defined Networking

Mensajes (TCP/TLS port 6653)

Tres tipos de mensajes

- **Controller-to-Switch:** Iniciados por el controller
 - Administrar o inspeccionar el estado del switch
 - Puede o no requerir respuesta del switch
 - Features, Configuration, Modify-state, Read-state, Packet-out, Barrier, Role-request, Asynchronous-configuration
- **Asynchronous**
 - Enviados por el switch sin que lo solicite el controlador
 - Informan de paquetes que arriban, cambios en el estado del switch, errores.
 - Packet-in, Flow-removed, Port-status, Error
- **Symmetric**
 - Enviados por switch o controlador sin ser solicitados
 - Hello, Echo, Experimenter



Software-Defined Networking

Otras características

- **Múltiples controladores:**
 - Role equal:
 - Pueden enviar mensajes controller-to-switch
 - Reciben todos los mensajes asíncronos.
 - Role Slave:
 - read-only access sobre el switch.
 - Por defecto no reciben todos los mensajes asíncronos, salvo los port-status
 - Role Master
 - Similar a equal, solamente uno en ese estado.
- **Conexiones auxiliares**
 - Opcionalmente creadas por el switch para performance aprovechando sus capacidades-paralelismo (TLS, TCP, DTLS o UDP)

Referencias

OpenFlow: enabling innovation in campus networks. Nick McKeown, Tom Anderson, HariBalakrishnan, Guru Parulkar, Larry Peterson, Jennifer RexfordScott Shenker, Jonathan Turner. ACM SIGCOMM Computer Communication Review, Volume 38, Number 2, April 2008.

Software-Defined Networking: The New Norm for Networks. ONF White paper. 2012

<http://www.opennetsummit.org/archives/apr12/hoelzle-tue-openflow.pdf>

SDN: Software Defined Networks An Authoritative Review of Network Programmability Technologies. O'Reilly Media. 2013

<https://www.opennetworking.org/>

RFC 7426 “Software-Defined Networking (SDN): Layers and Architecture Terminology”

UIT-T Rec Y.3300 “Framework of software-defined networking”