

**Технология виртуализации  
сетевых функций NFV и  
программно-конфигурируемые  
сети SDN**

**Профессор В.Ю. Деарт**

# Содержание

1. Общие принципы виртуализации
2. Виртуализация сетевых функций NVF
3. Пример виртуализации LTE радиочасти
4. Пример виртуализации EPC и IMS
5. Что такое облако?
6. Что такое OpenStack?
7. Концепция SDN
8. Введение в OpenFlow

# Общие принципы виртуализации

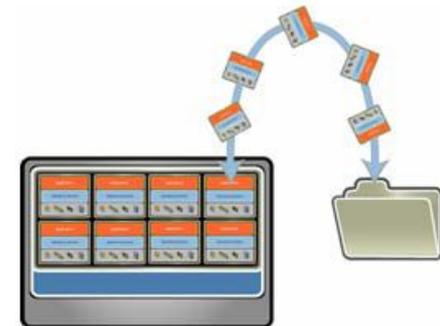
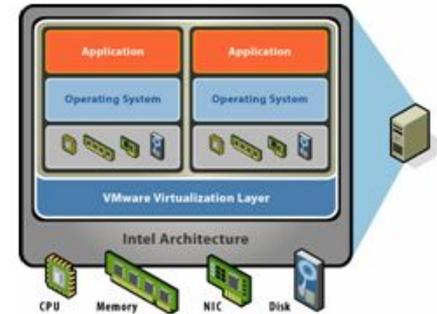
**Virtualization** abstracts the underlying physical structure of various technologies. Virtualization, in computing, is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system, a storage device or network resources

## Server virtualization

- Creates multiple isolated environments
- Allows multiple OS's and workloads to run on the same physical hardware
- Solves the problem of tight coupling between OS's and hardware

# Virtual Machines

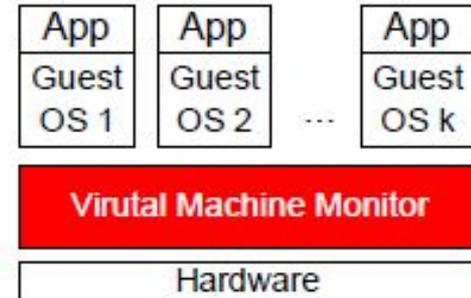
- **Virtual machines provide:**
- Hardware independence—Guest VM sees the same hardware regardless of the host hardware
- Isolation —VM’s operating system is isolated from the host operating system
- Encapsulation—Entire VM encapsulated into a single file



# Two classifications of Hypervisors

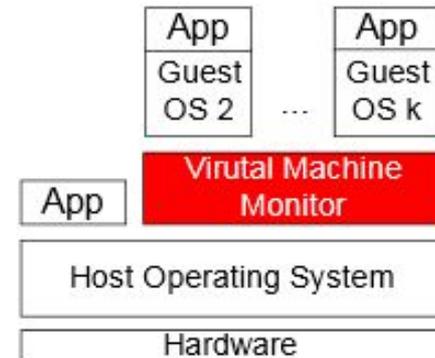
## Native (Hardware-Level): software runs

- directly on top of a given hardware platform as a control program for operating systems



## Hosted (OS-Level): software runs

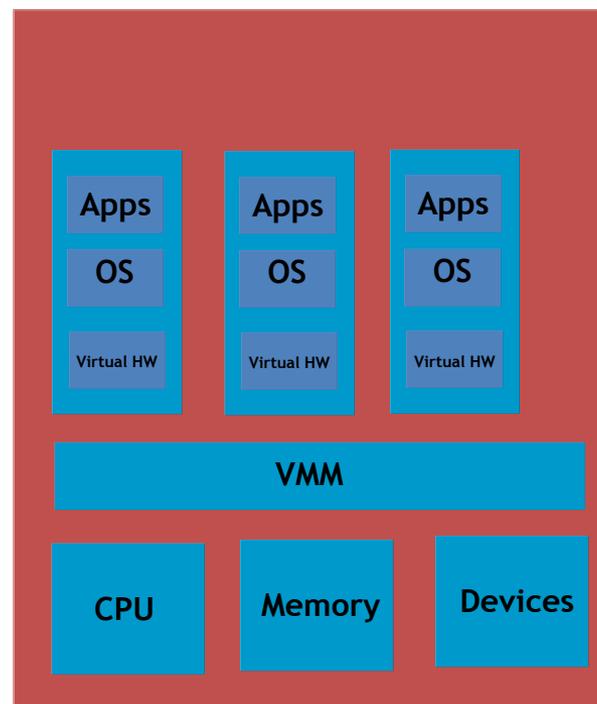
- within an operating system environment as a control program for other operating systems



# Virtualization and Hypervisors

## Hypervisors: Type-1

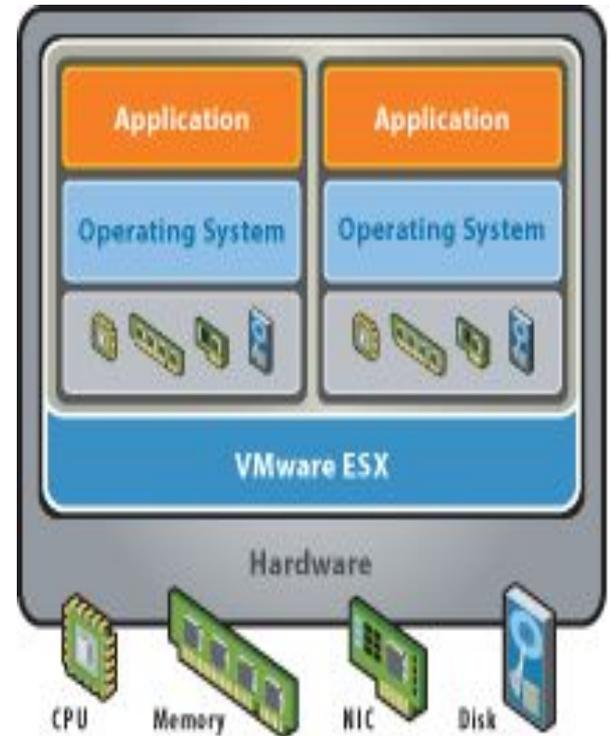
- Properties
  - Thin hypervisor layer with direct hardware control running on top of bare metal.
  - Hypervisor manages CPU and memory resources.
  - Guest OSs rely either on hypervisor or its facilities for I/O.
  - Strong partitioning and isolation by controlling the hypervisor.
- Applications
  - Server virtualization in datacenters.



# Virtualization and Hypervisors

## Hypervisors: Type-1 VMware ESXi/ESX

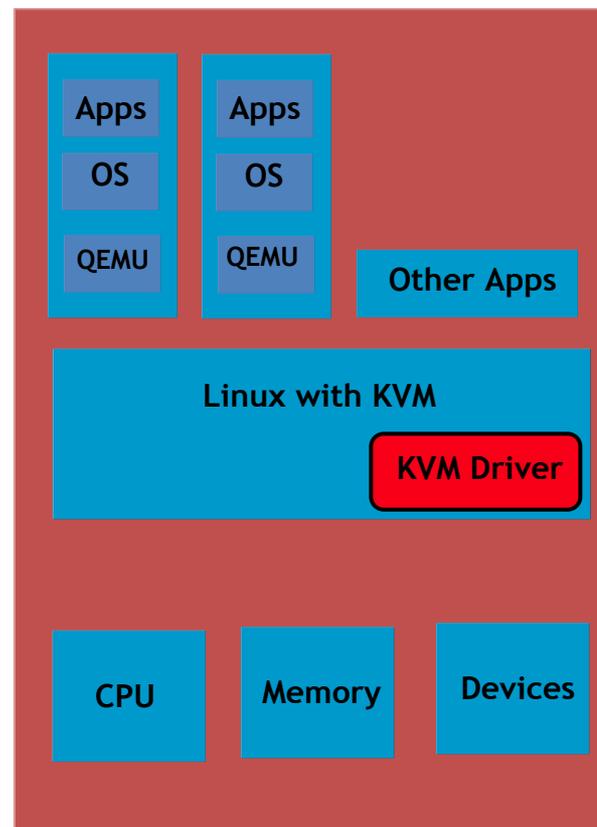
- Type-1 bare metal hypervisor.
  - ESX is the original version that includes “Service Console” for management.
  - ESXi is a compact version with minimal management.
    - Management through Web interface and vSphere.
- Supports any guest OS.
  - Windows, Linux, etc
- Key technology: Binary translation.
  - Enables direct execution of Guest OS without modifications.
- I/O managed by hypervisor.



# Virtualization and Hypervisors

## Hypervisors: Type-1 KVM

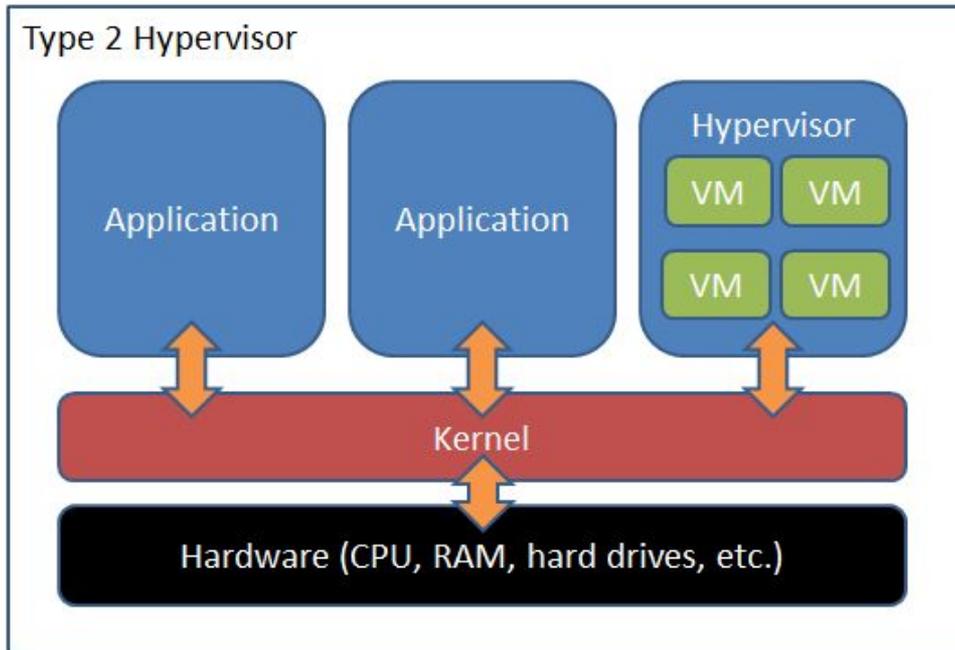
- Requires hardware virtualization.
- Standard with several Linux distributions.
  - QEMU + KVM Driver -> Virtualization.
- QEMU provides hardware abstraction and allows any unmodified OS to run on top of Linux.



# Virtualization and Hypervisor



## Hypervisors: Type-2



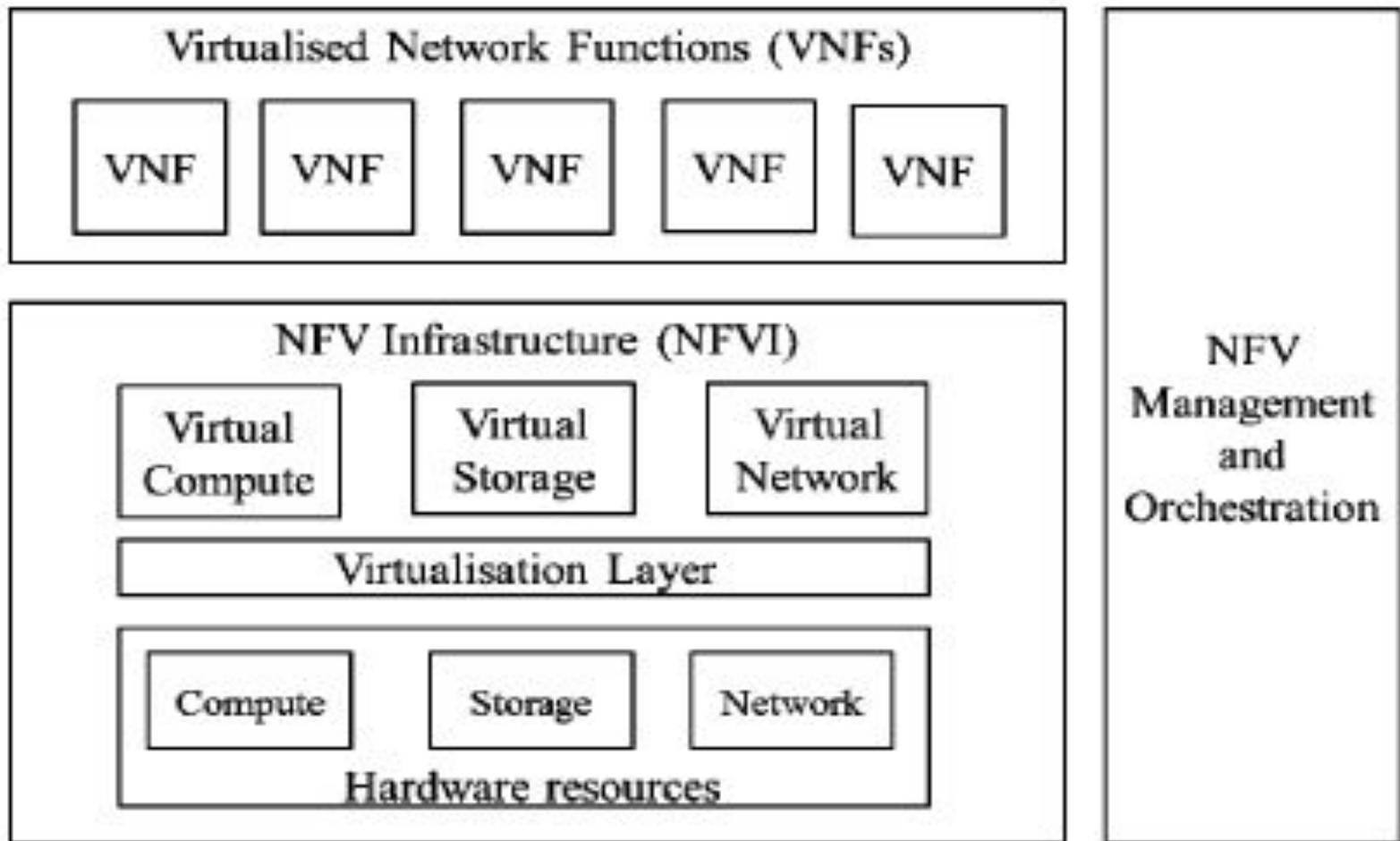
# Преимущества виртуализации

- Simplified administration
- Hardware independence/portability
- Increased hardware utilization
- Server consolidation
- Decreased provisioning times
- Improved security

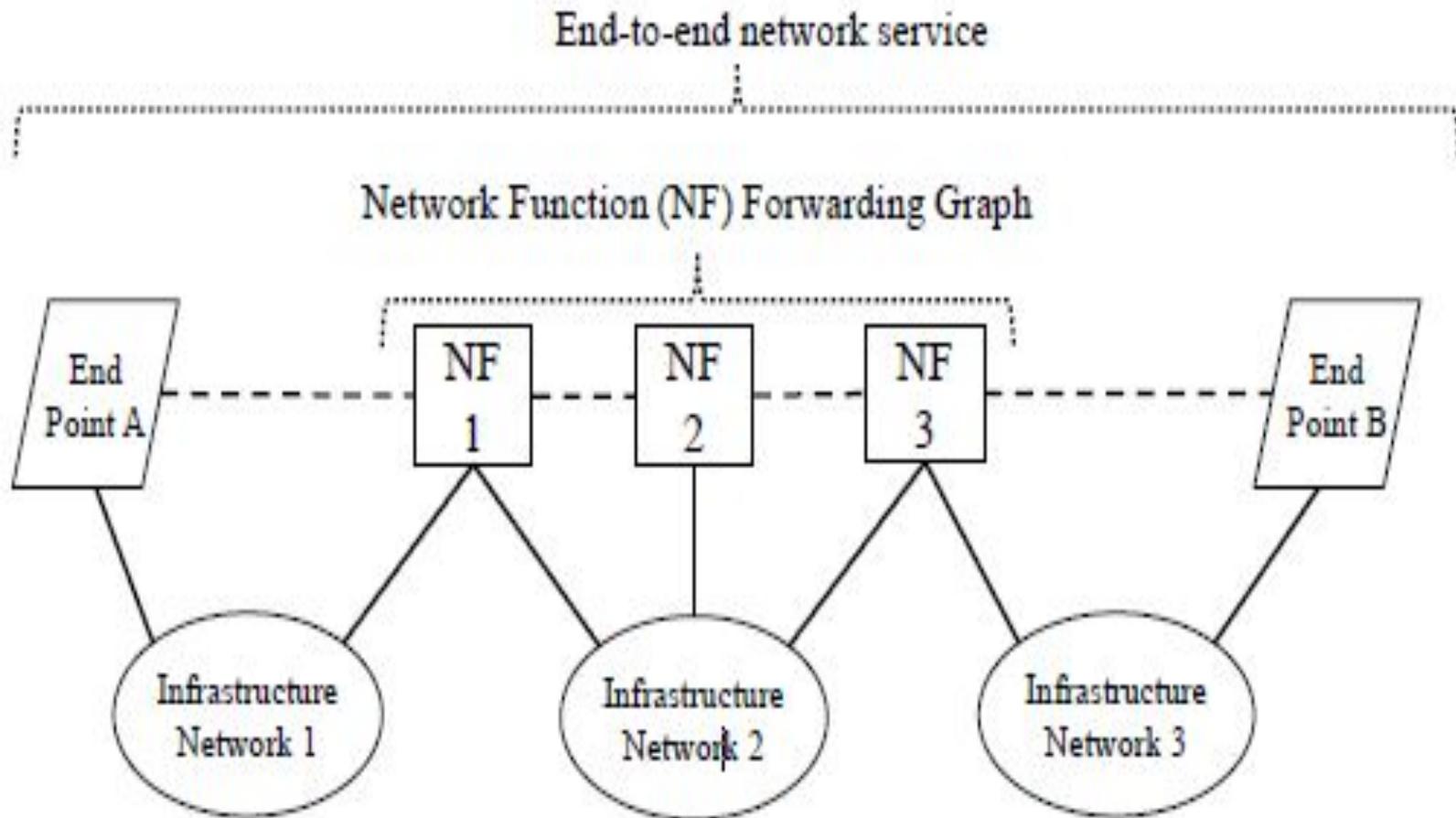
# Виртуализации сетевых элементов

- Network Function Virtualization NFV – это замена сетевых элементов IMS, Firewall, DNS на виртуальные машины VM, которые выполняют функции сетевых элементов.
- Виртуальные машины могут размещаться в стандартных серверах HP, IBM или в облаках (Clouds).

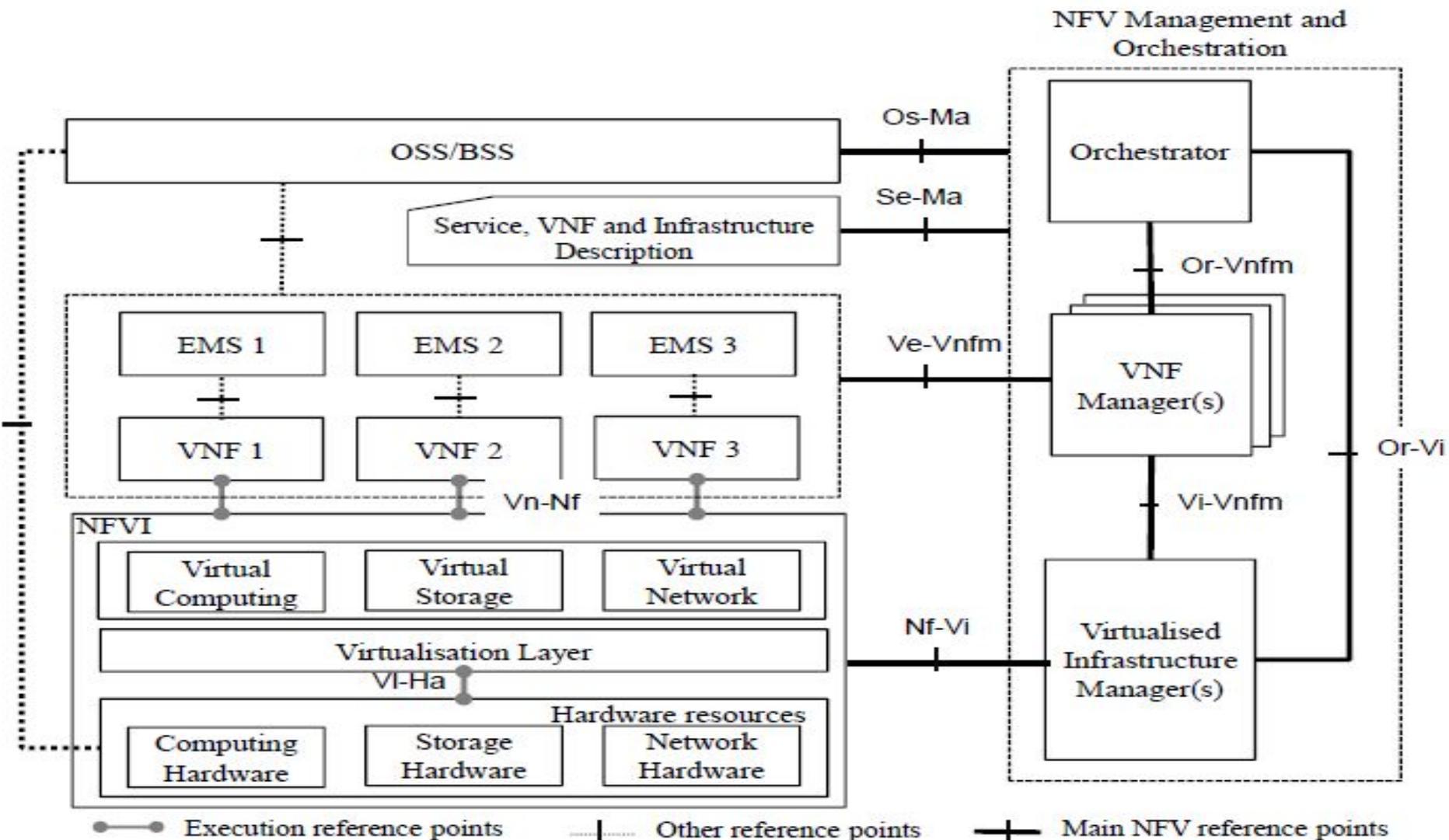
# Архитектура NFV



# Граф пересылки между VNF



# Архитектура и интерфейсы NFV



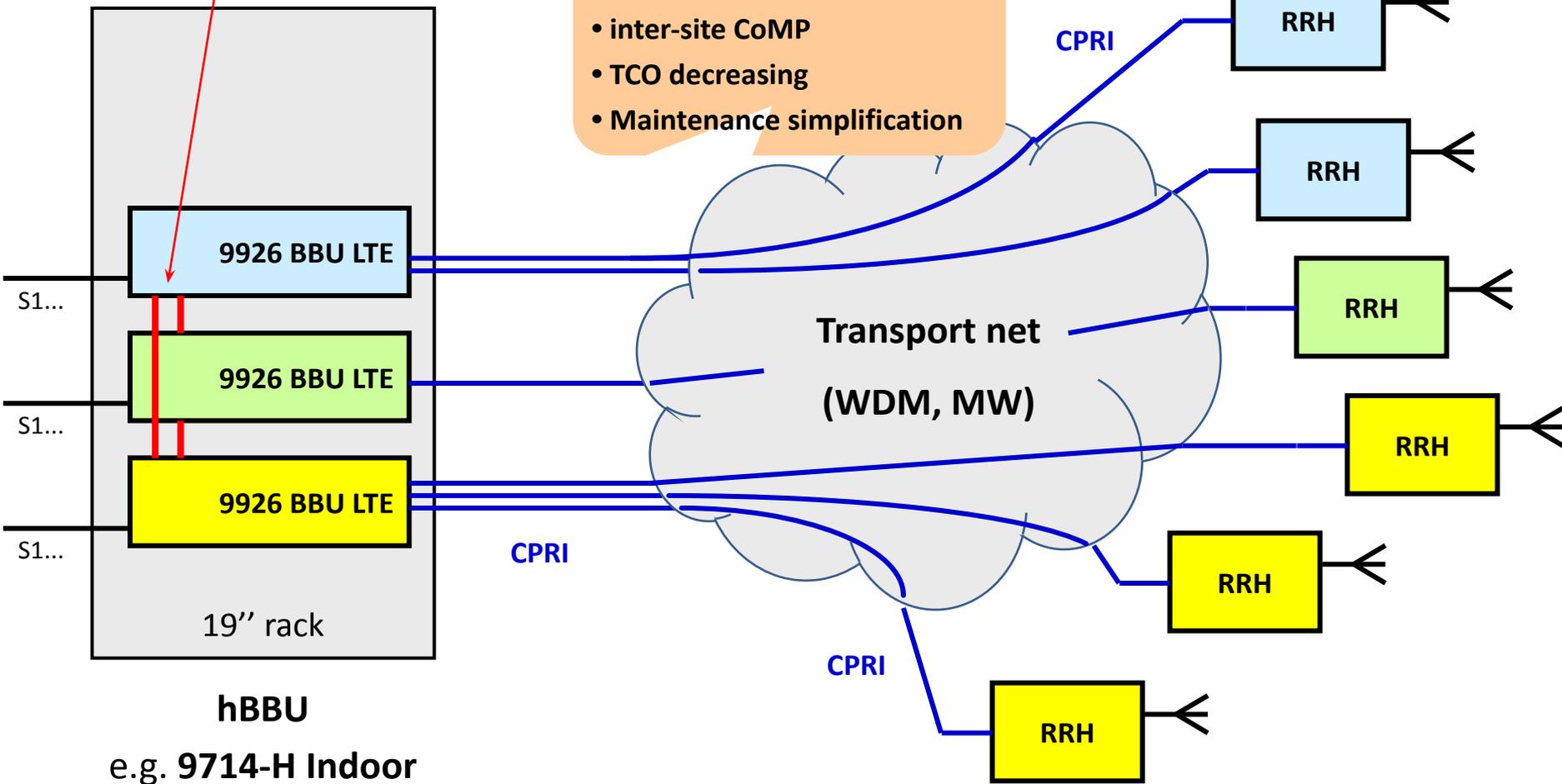
# Пример построения сети LTE (1)

## Аппаратное решение

**sRIO interface for inter-eNB CoMP**

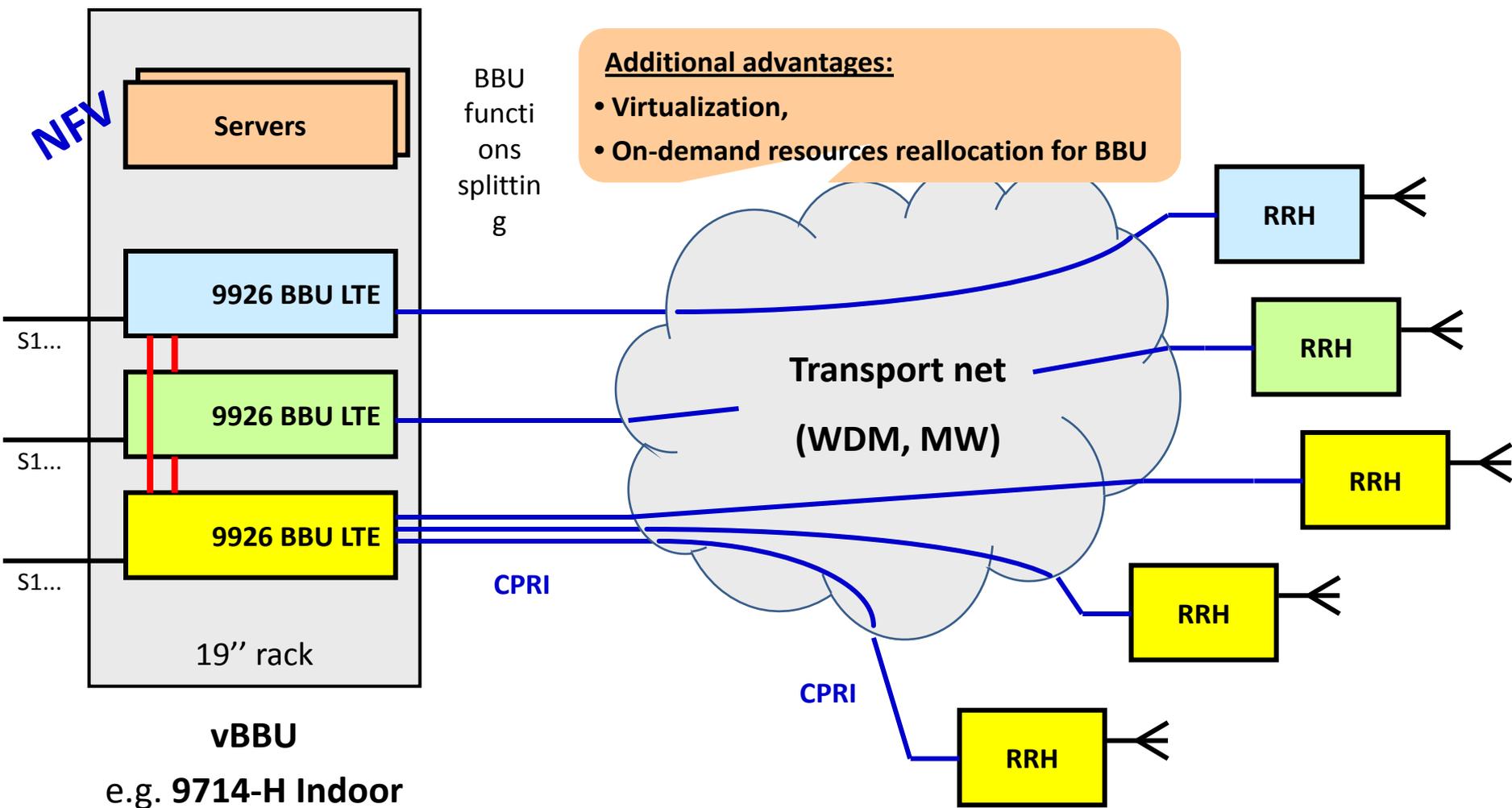
### Advantages:

- inter-site CoMP
- TCO decreasing
- Maintenance simplification



# Пример построения сети LTE (2)

## Частичная виртуализация

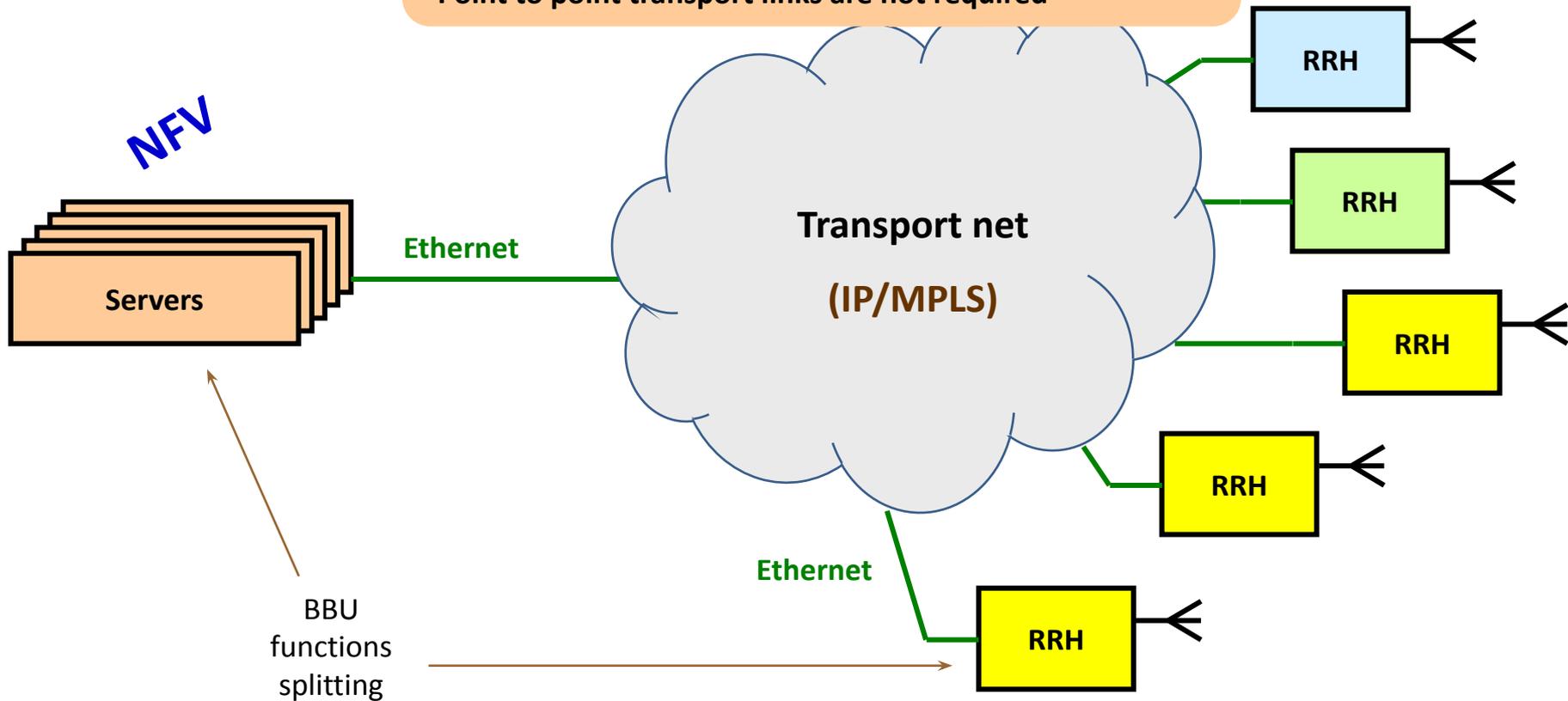


# Пример построения сети LTE (3)

## Полная виртуализация.

### Additional advantages:

- Full Virtualization,
- Point to point transport links are not required

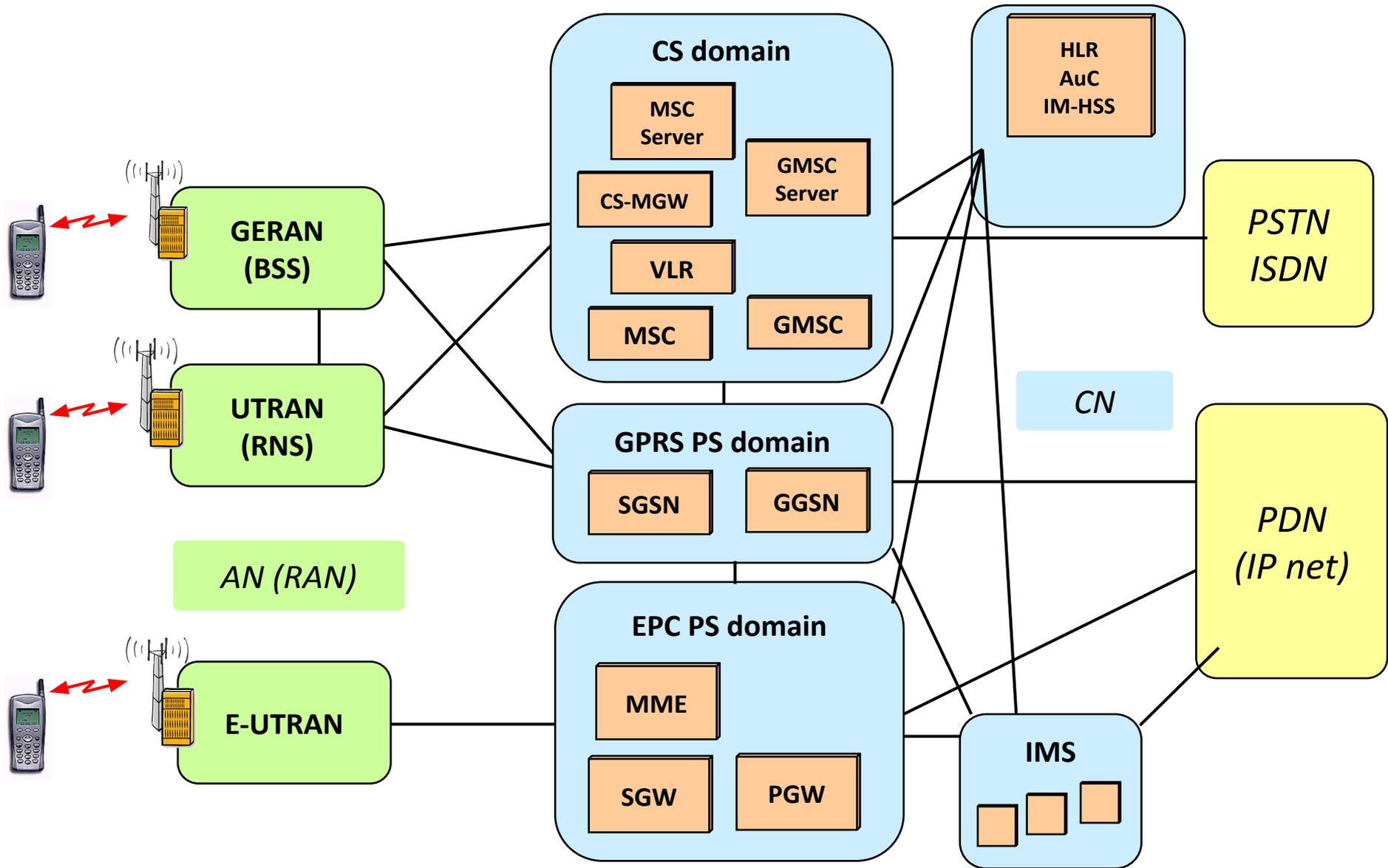


# Виртуализация ядра мобильной сети и IMS

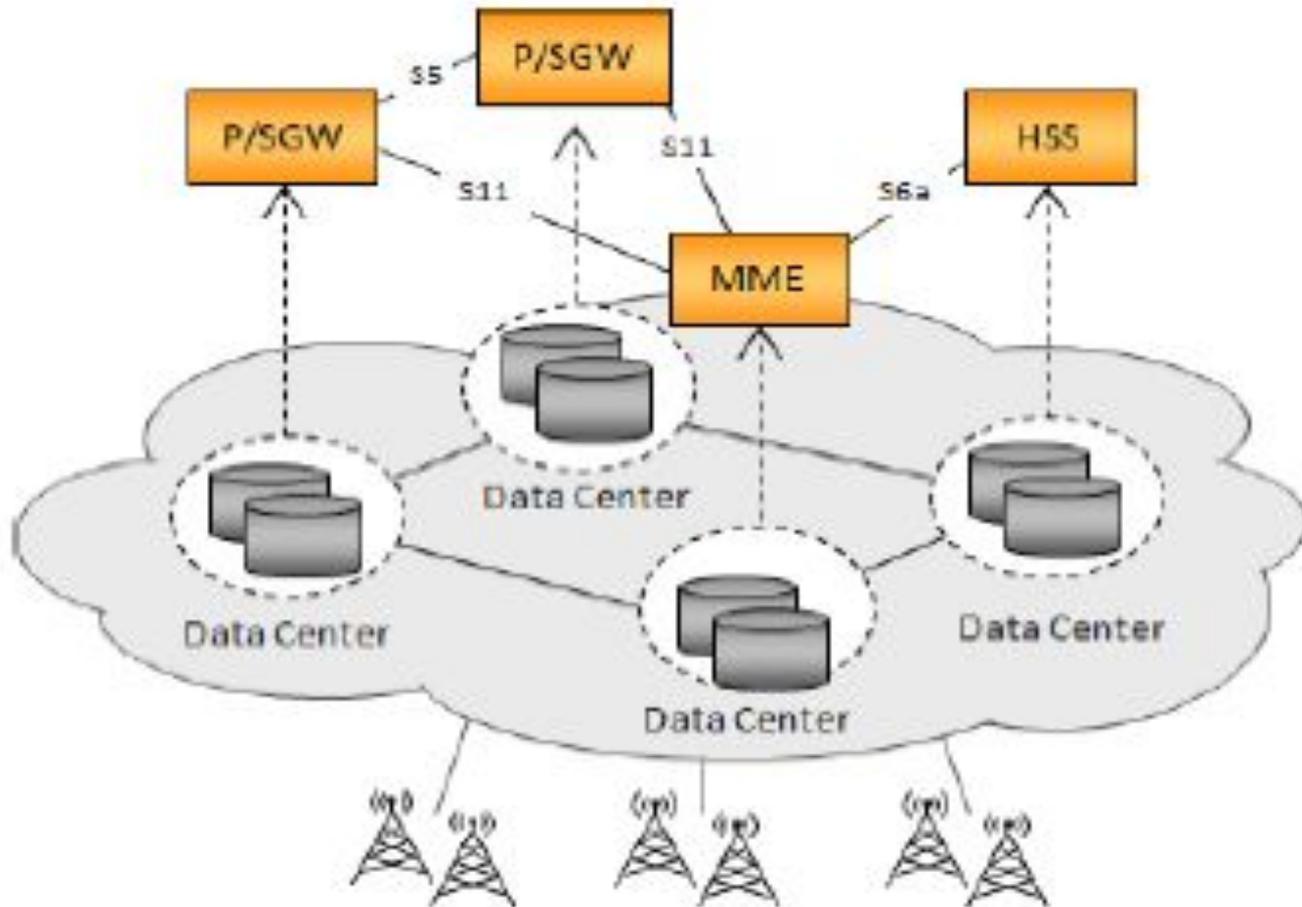
## 1. Цель виртуализации

- 1.1 Сократить расходы на владение сетью
- 1.2 Обеспечить более высокую надежность за счет динамического изменения виртуализированной структуры
- 1.3 Обеспечить быструю подстройку сетевой инфраструктуры под изменения трафика
- 1.4 Обеспечить быстрое изменение топологии сети для обеспечения ее максимальной пропускной способности

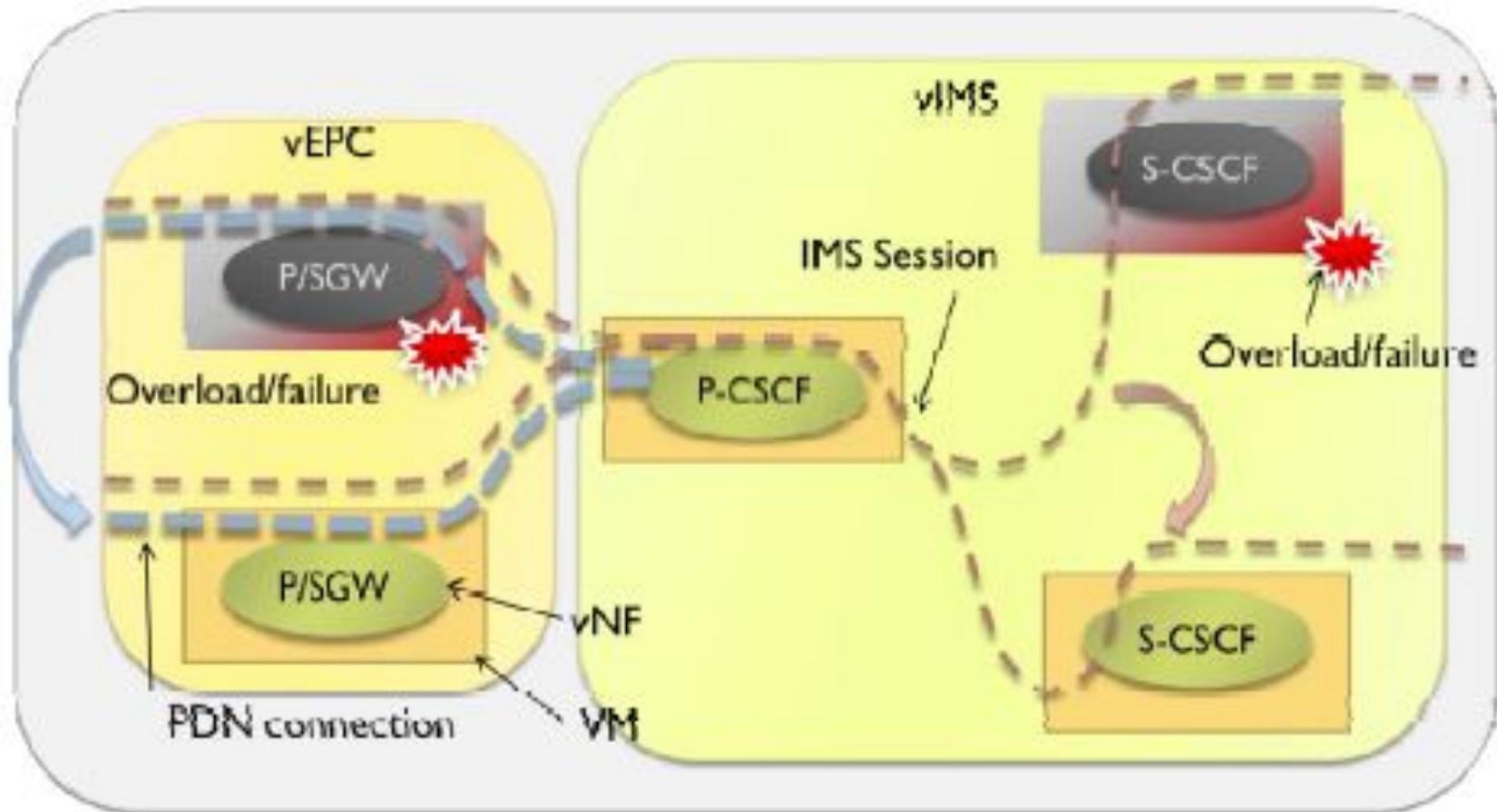
# Структура 3GPP mobile network.



# Виртуализированное решение для EPC



# Виртуализация IMS



# Что такое Облако?

**Облачные вычисления** это модель для организации удобного **сетевого доступа по запросу** к общему разделяемому пулу конфигурируемых **вычислительных ресурсов** (например, сетей, серверов, хранилищ, приложений и сервисов), которые могут быть **быстро настроены и предоставлены** с минимальными усилиями и взаимодействием со стороны провайдера услуги.

## Традиционный



Покупка дома

VS

## Облако



Аренда дома

# Сравнительные характеристики облаков

- Объединение ресурсов в общие пулы
- Скоростной сетевой доступ
- Самообслуживание по требованию
- Мгновенная эластичность
- Прозрачность измерения и выставления счета



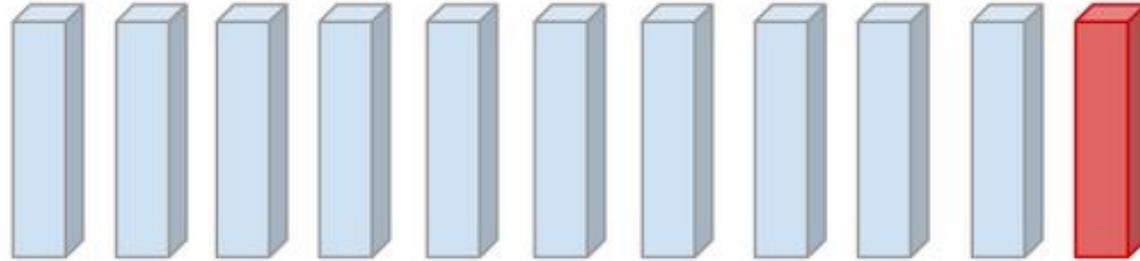
Виртуализация



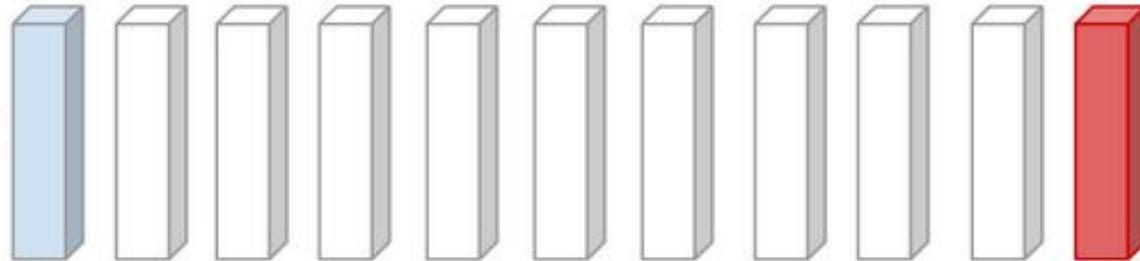
Облачное ПО,  
например  
OpenStack

# Эластичность

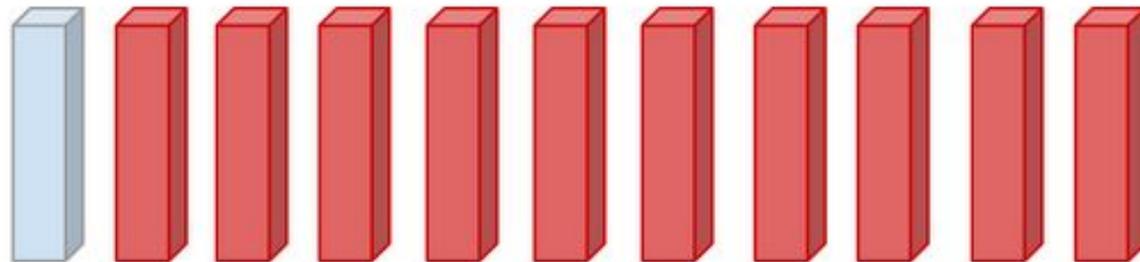
Дневное время.  
Все работают  
в компании А.



Рабочее время  
закончилось.  
Дело сделано,  
можно и по  
домам.



Остались самые  
упорные. ~~Мафия~~  
~~открывает глаза~~  
Компания В  
начинает работу.



Сервера выделенные компании А



Сервера выделенные компании В

# Модель облачных сервисов

## Клиенты Облака

Веб-браузер, мобильные приложения, тонкий клиент, эмулятор терминала,...



## SaaS

CRM, Email, виртуальный рабочий стол, игры,...



## PaaS

база данных, веб сервер, сервер приложений, средства разработки и тестирования, ...

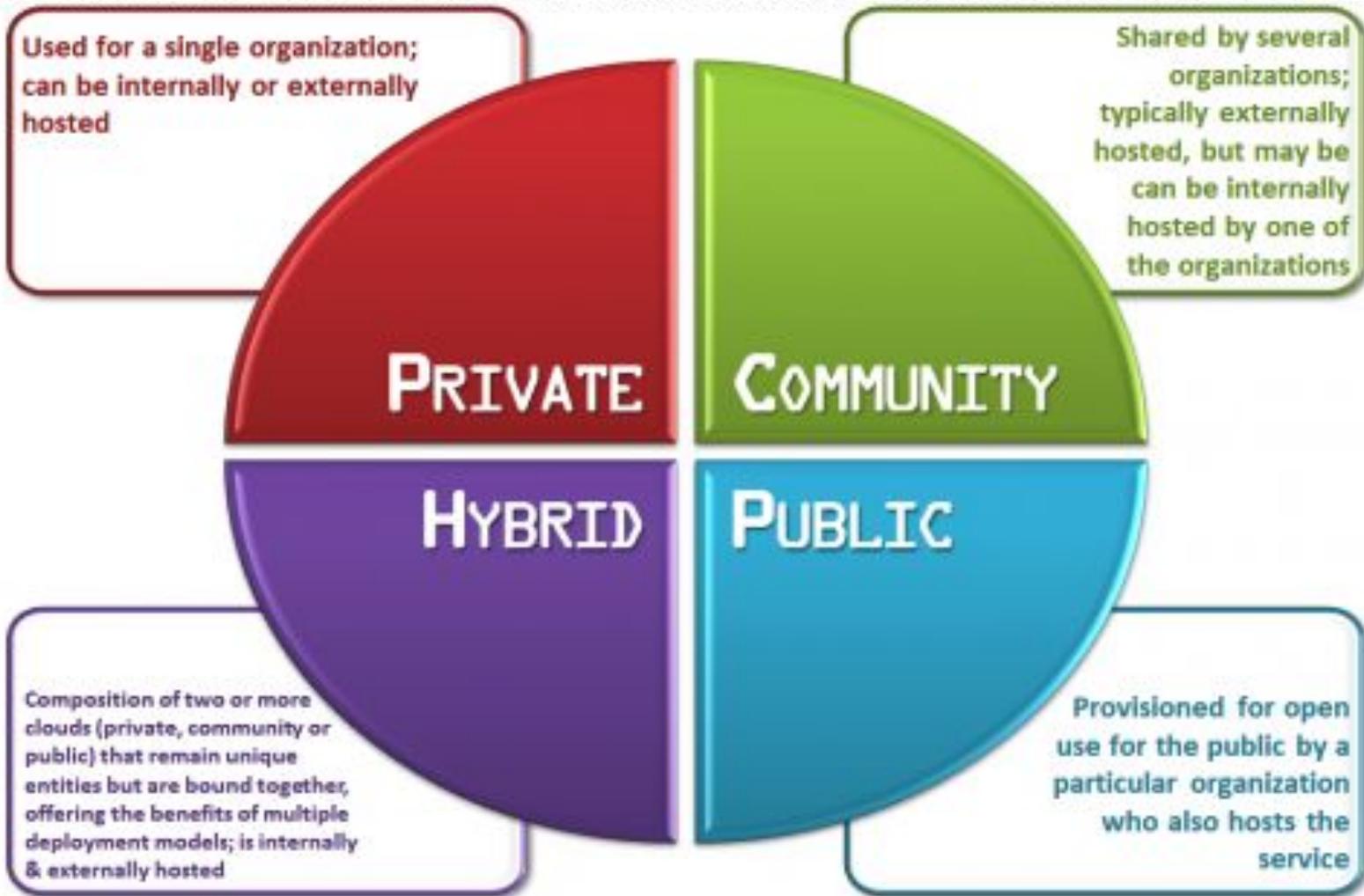


## IaaS

виртуальные машины, сервера, хранилища, балансировщики нагрузки, сети...



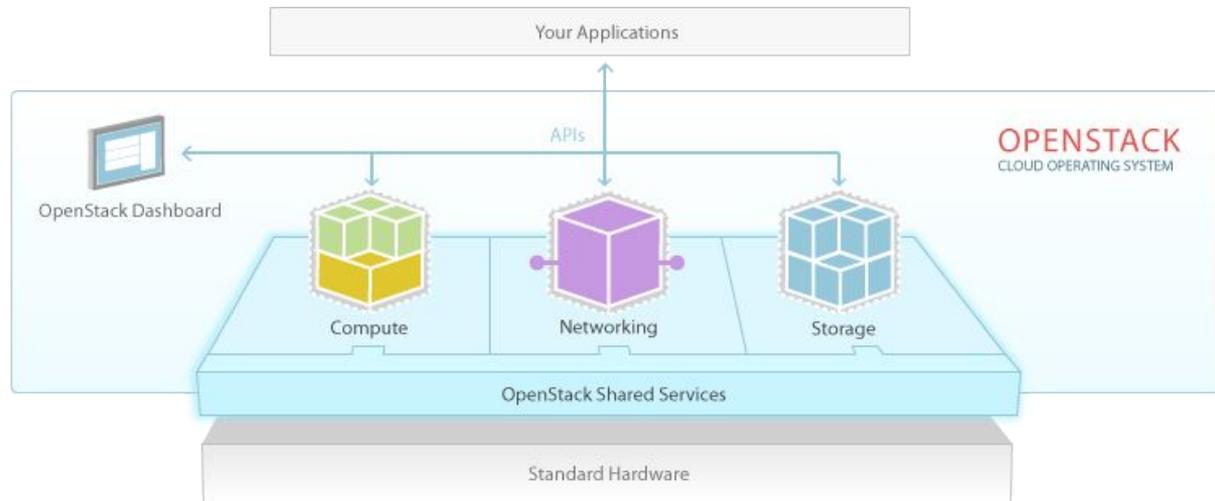
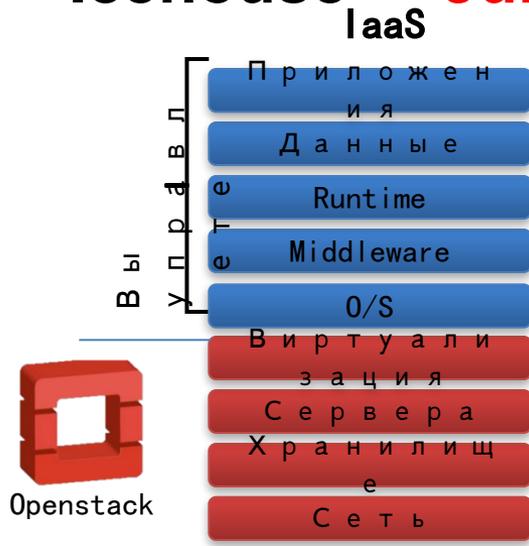
# Типы облаков



# Что такое OpenStack ?

**OpenStack:** комплекс проектов свободного программного обеспечения, который может быть использован для создания инфраструктурных облачных сервисов.

**Версии:** Austin -> Bexar -> Cactus -> Diablo ->  
Essex -> Folsom -> Grizzly -> Havana ->  
Icehouse -> **Juno** -> Kilo -> Liberty



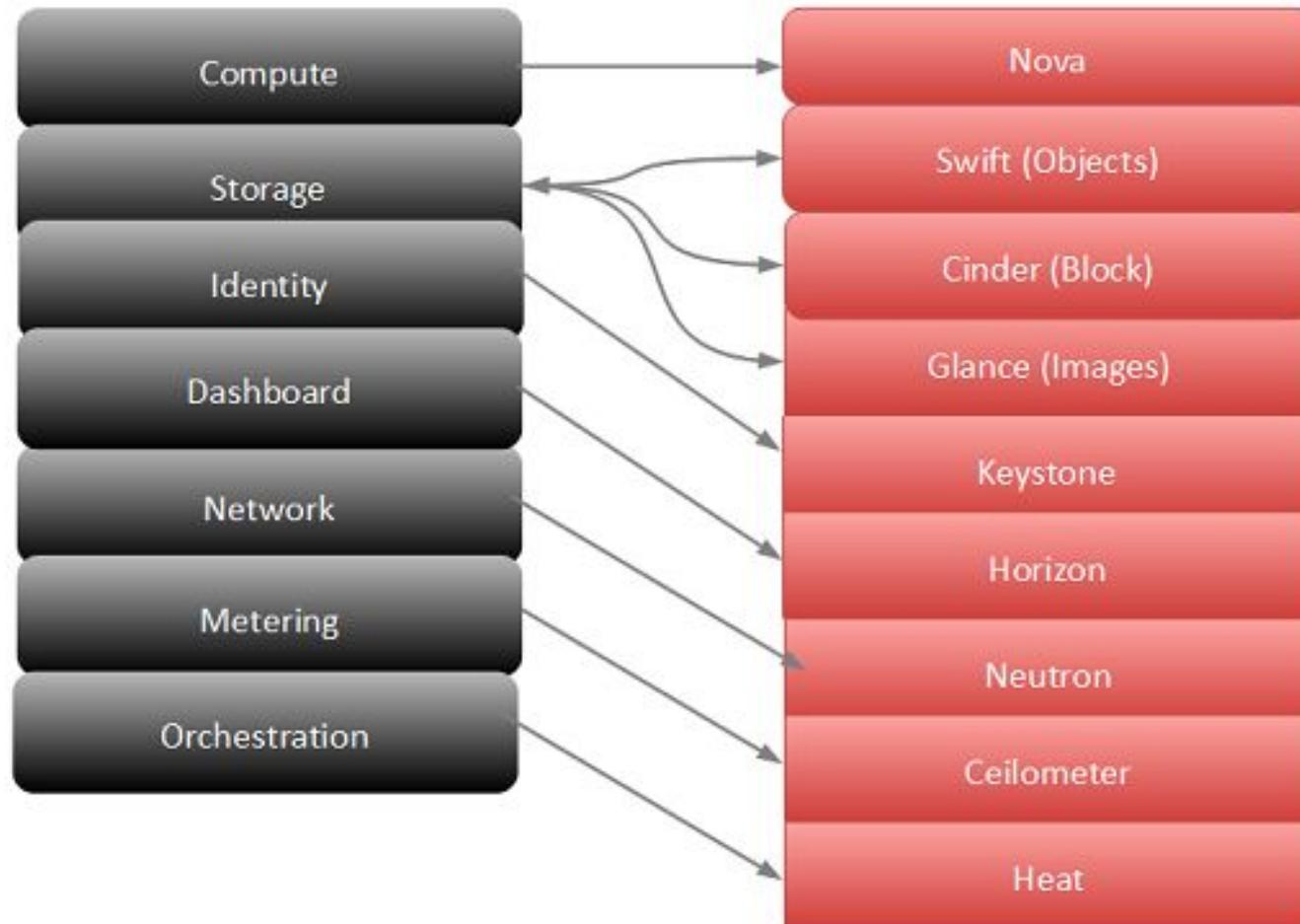
# Что такое OpenStack ?



# Components of OpenStack

IaaS

OpenStack проекты



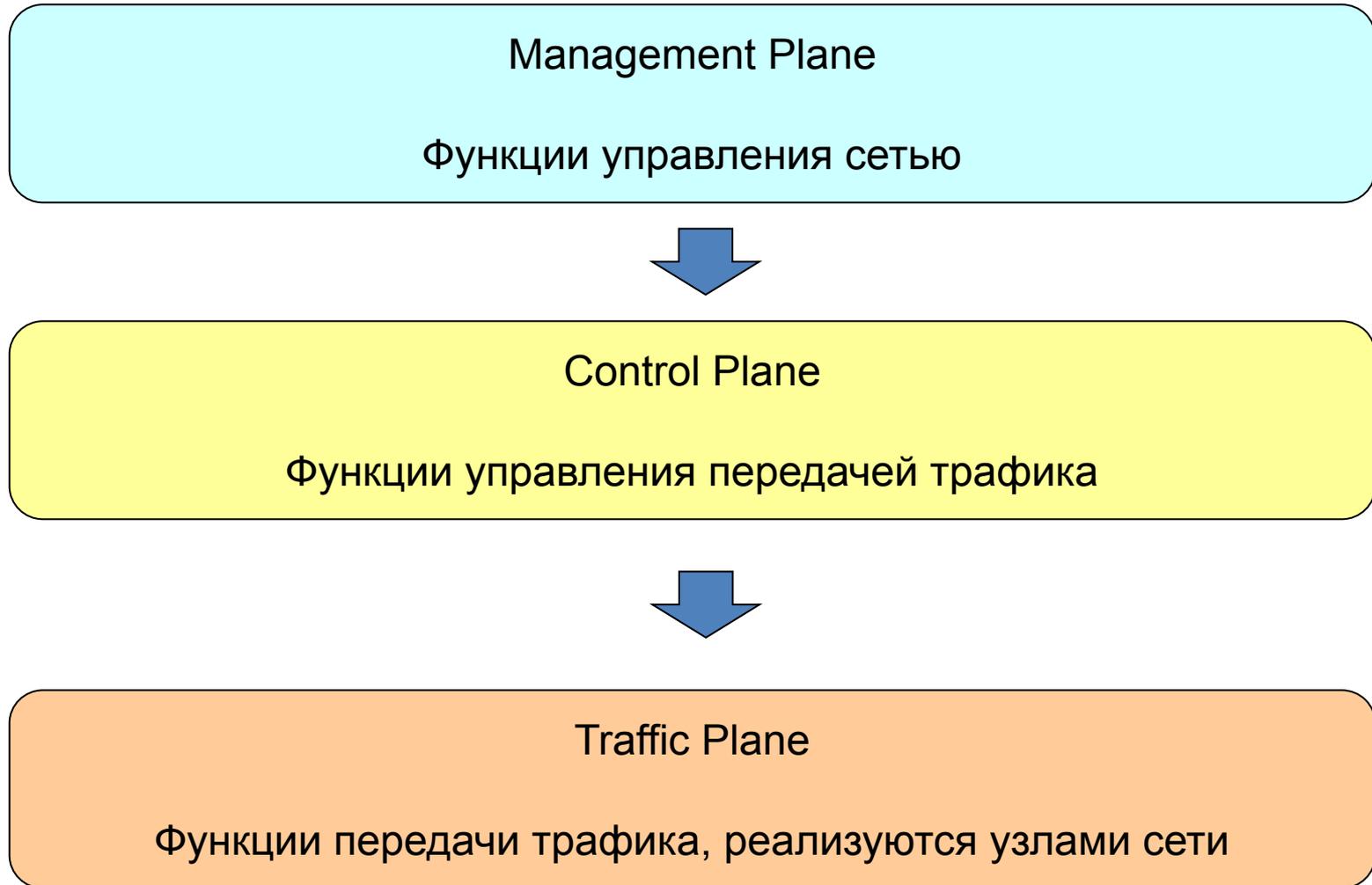


---

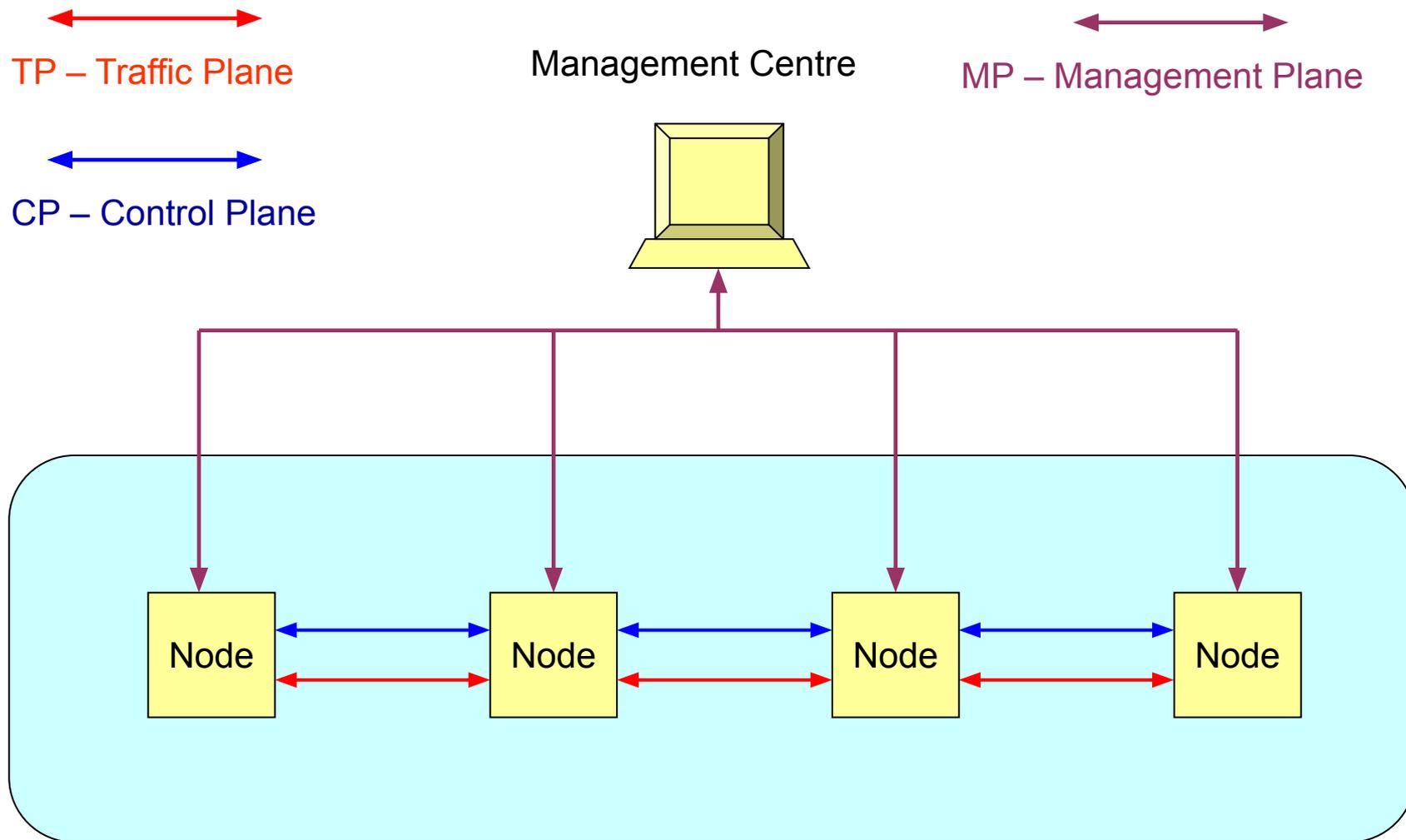
# Концепция SDN

# Traffic Plane, Control Plane, Management Plane

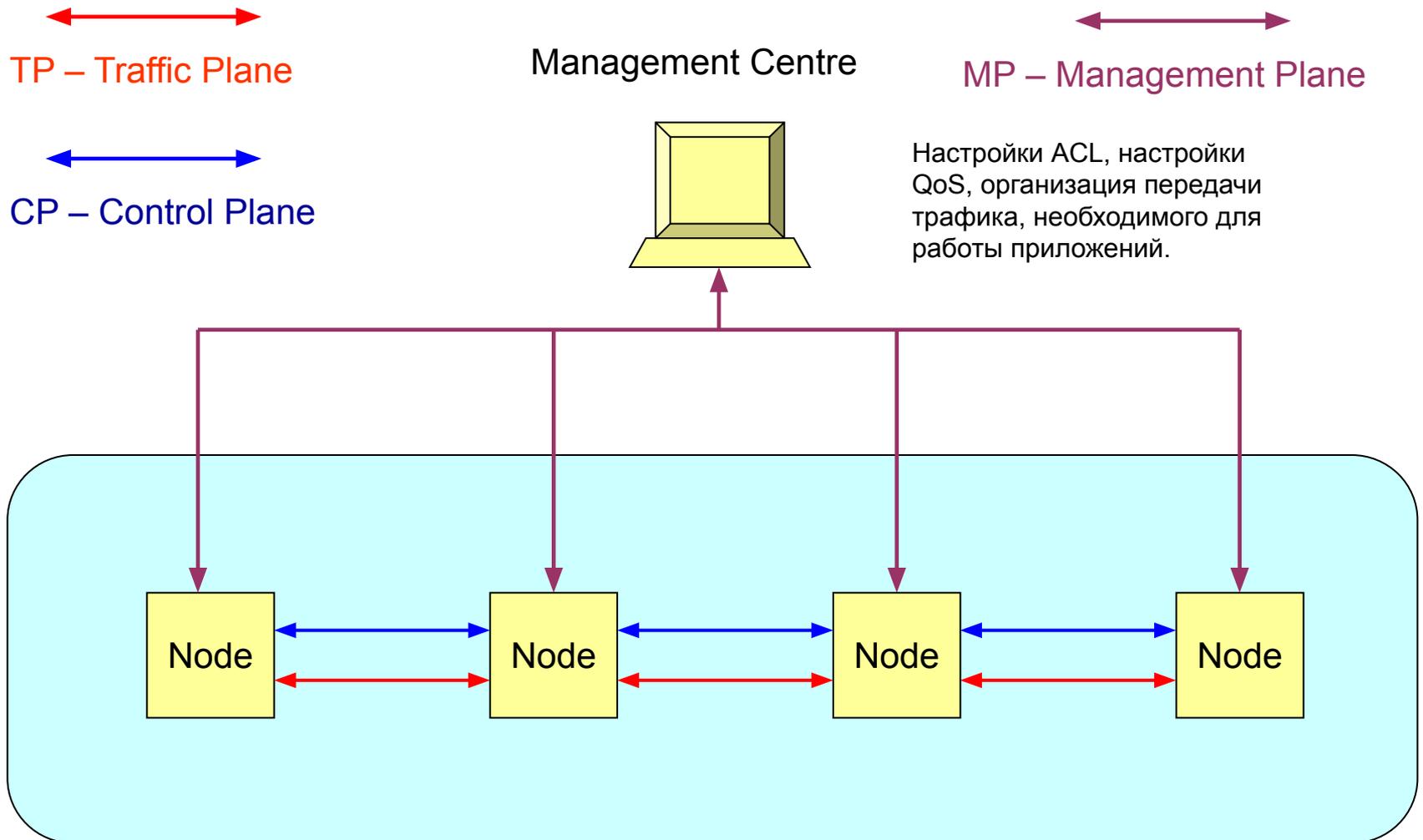
---



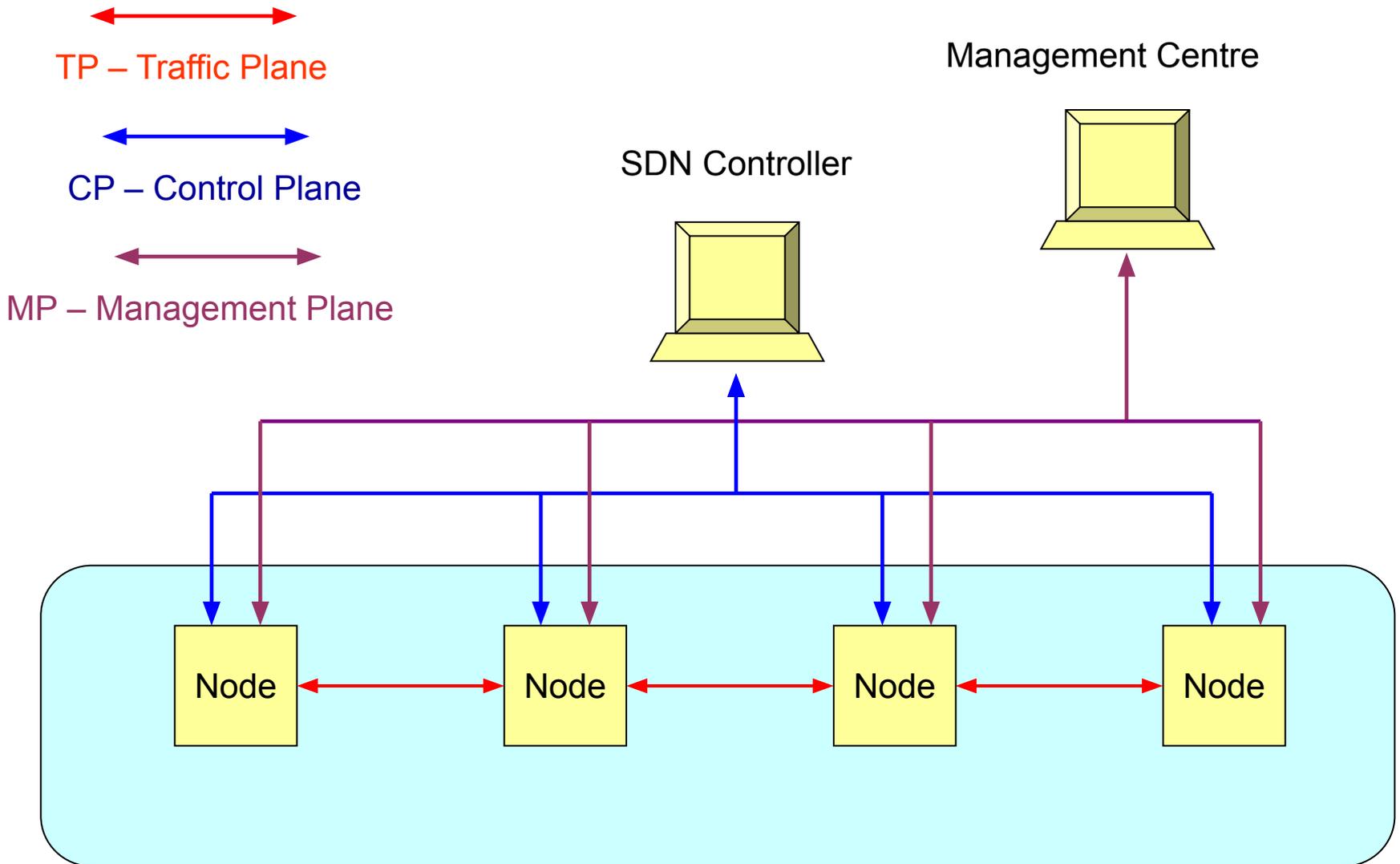
# Традиционная реализация трех планов.



# Задачи прикладного уровня.

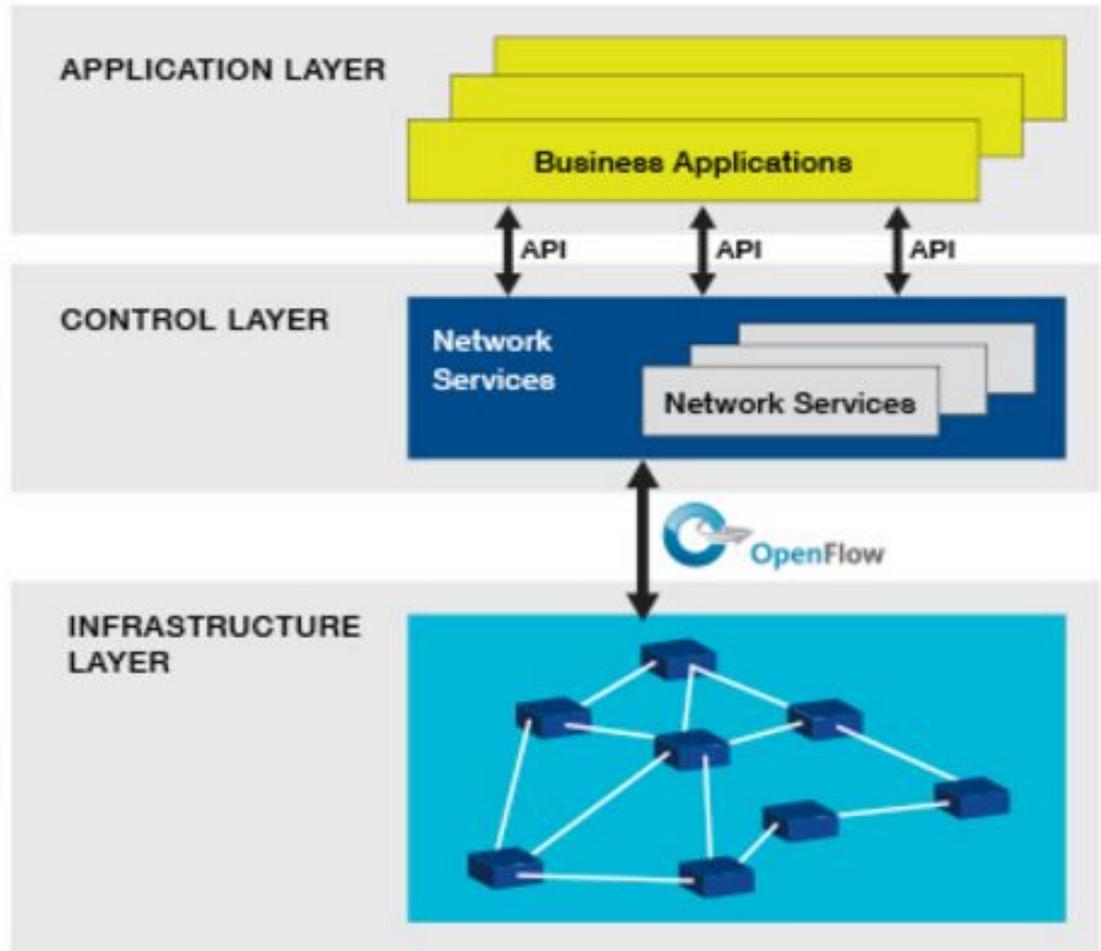


# Концепция SDN.



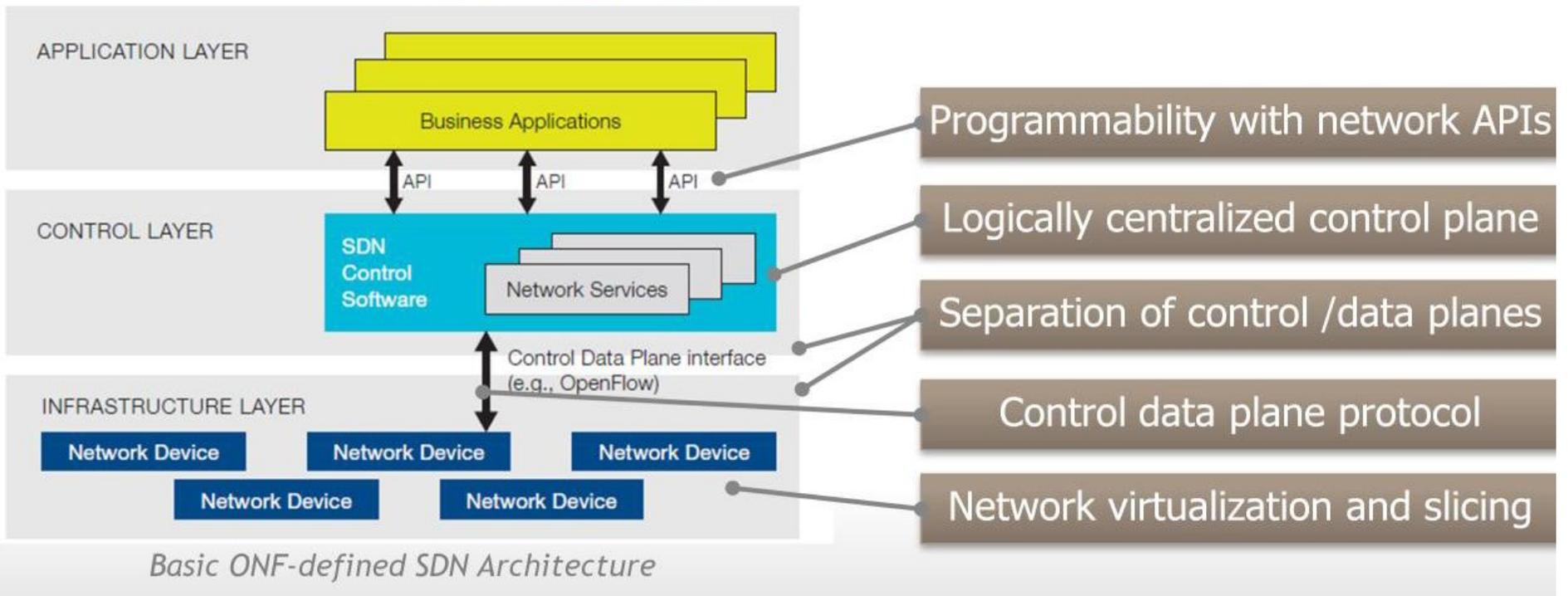
# Реализация приложений в сети SDN.

- Формирование условий передачи трафика для уровня управления
- Формирование правил передачи трафика в соответствии с условиями прикладного уровня
- Передача трафика в соответствии с правилами, сформированными на уровне управления



# Программно-конфигурируемые сети

## Software Defined Networking

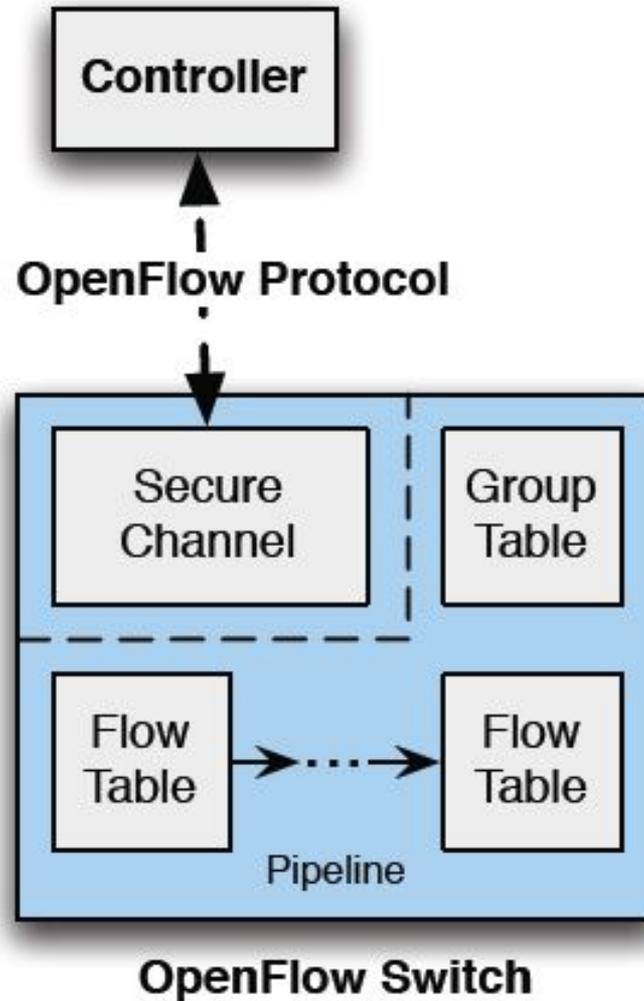


---

# Введение в OpenFlow

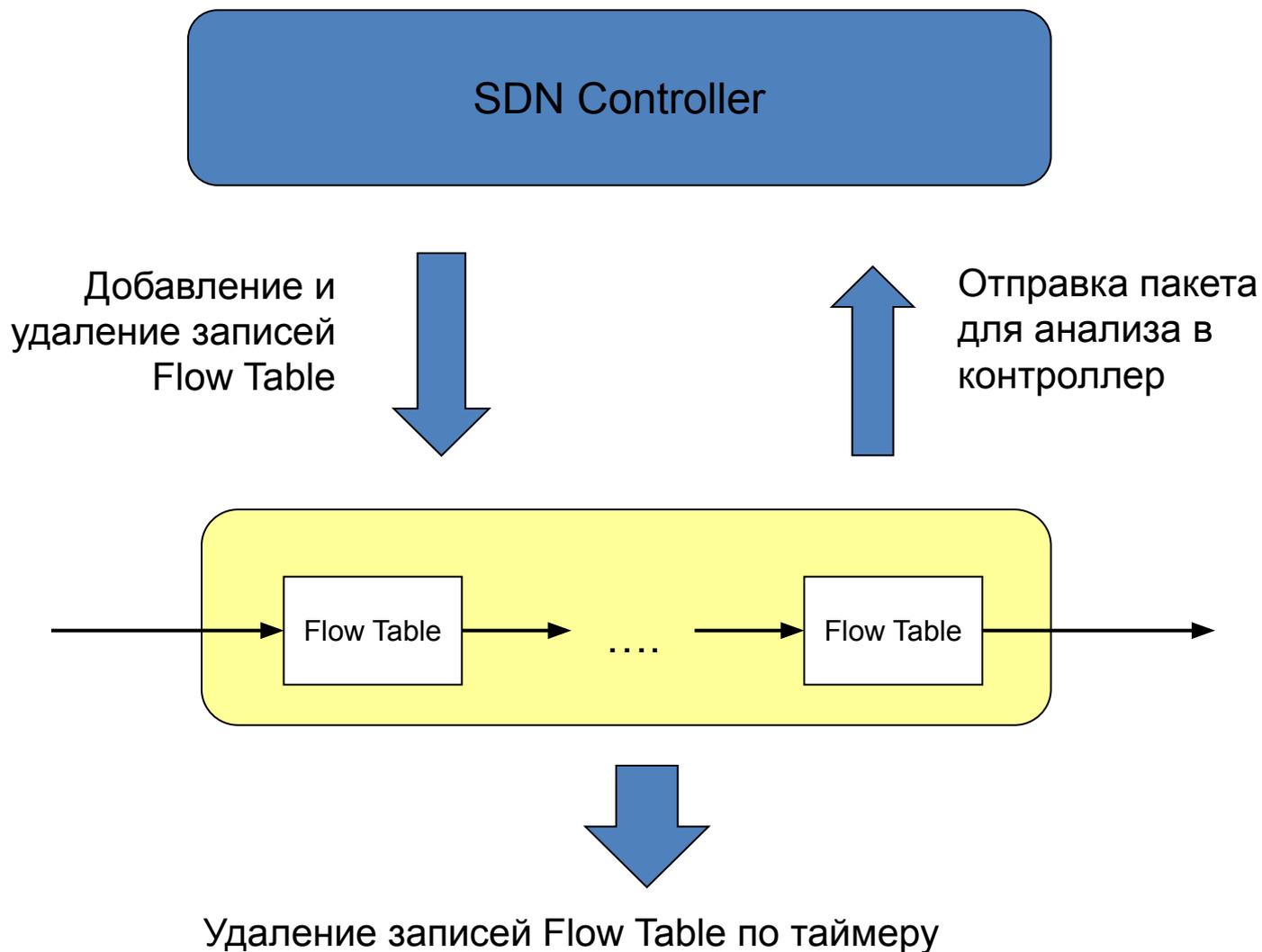
# Основные принципы организации OpenFlow.

---

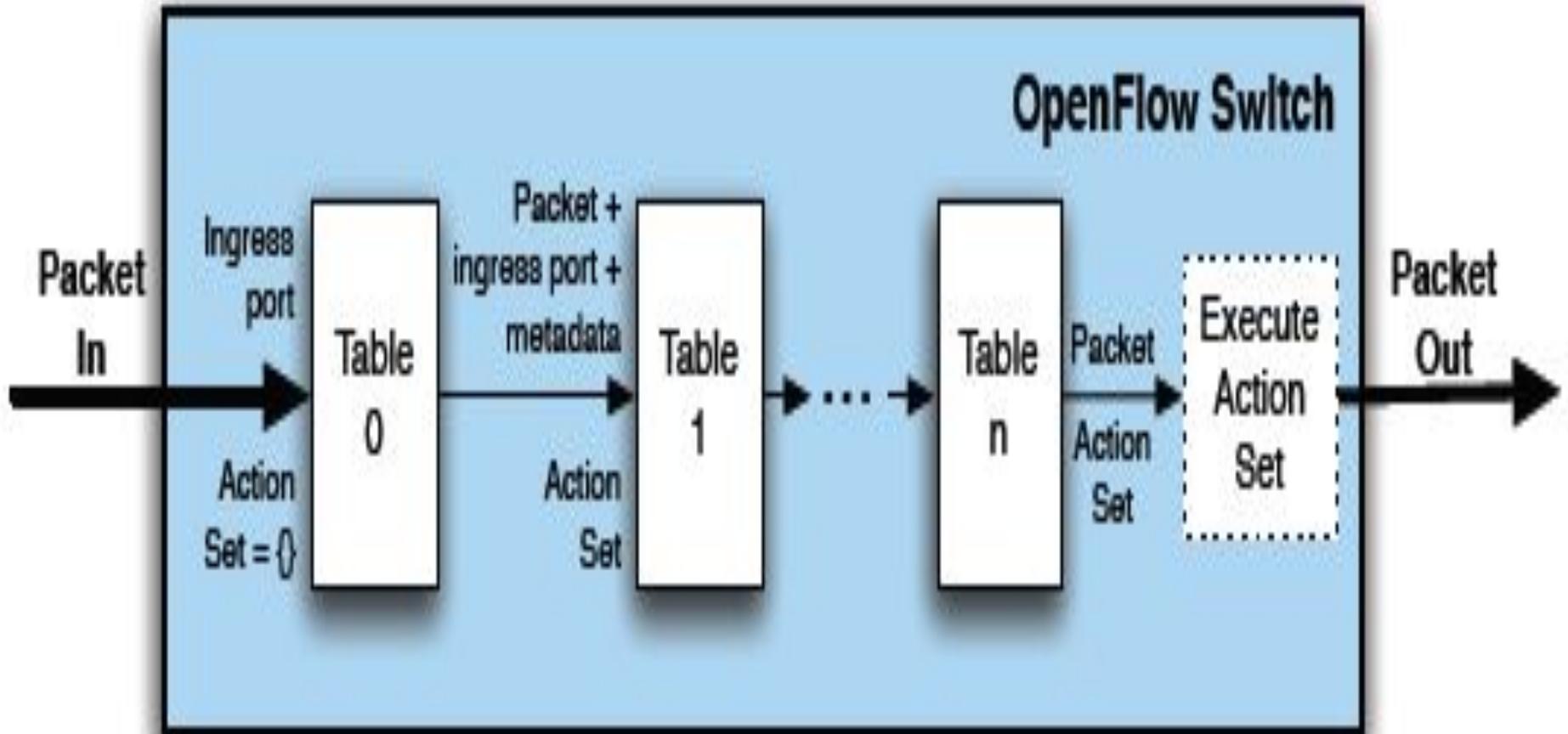


# Обработка потоков данных.

---



# Прохождение потока пакетов



# Параметры влияющие на прохождение потока

Ingress Port
Metadata
Ether src
Ether dst
Ether type
VLAN id
VLAN priority
MPLS label
MPLS traffic class
IPv4 src
IPv4 dst
IPv4 proto / ARP opcode
IPv4 ToS bits
TCP / UDP / SCTP src port
ICMP Type
TCP / UDP / SCTP dst port
ICMP Code

# Основные компоненты SDN

## ■ Application Program Interface (API)

■ Взаимодействия между приложением и сетью через SDN контроллер - Network Service Interface (NSI) типа REST и т.п.

■ Взаимодействие между сетевым контроллером и сетевыми элементами - интерфейсы Netconf/Yang, Open Flow (OF) и т.п.

**SDN:** развитие в направлении предоставления услуг по запросу приложений и поверх полностью автоматизированной виртуализированной сети



# Пример услуги E-Line

Пример услуги NSP. Эластичные услуги E-Line (динамическое изменение CIR/EIR)

