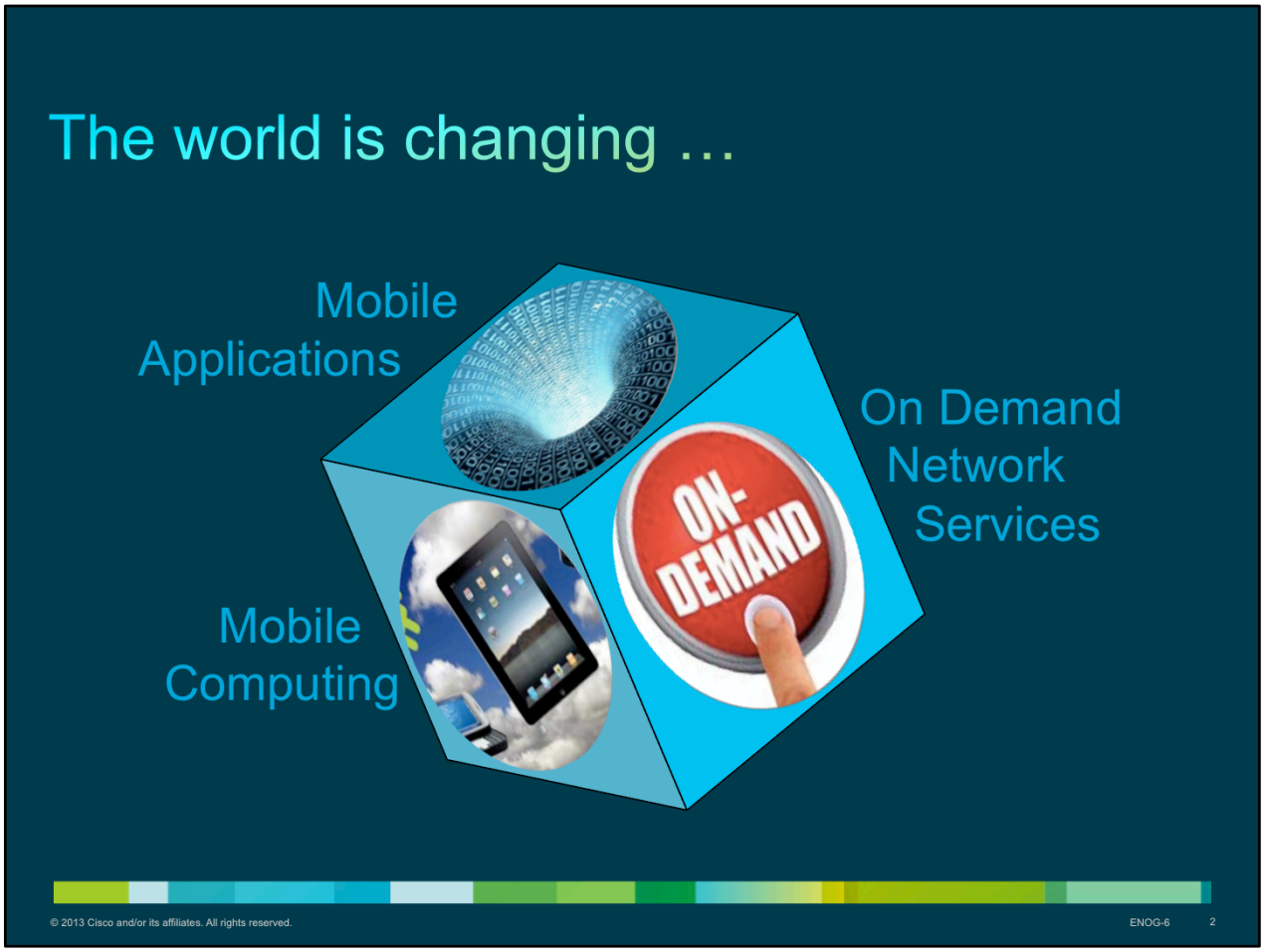
SDN and Network Virtualization

Vladimir Litovka vllitovk@cisco.com
Business Development Manager, Cisco Systems

September 2013



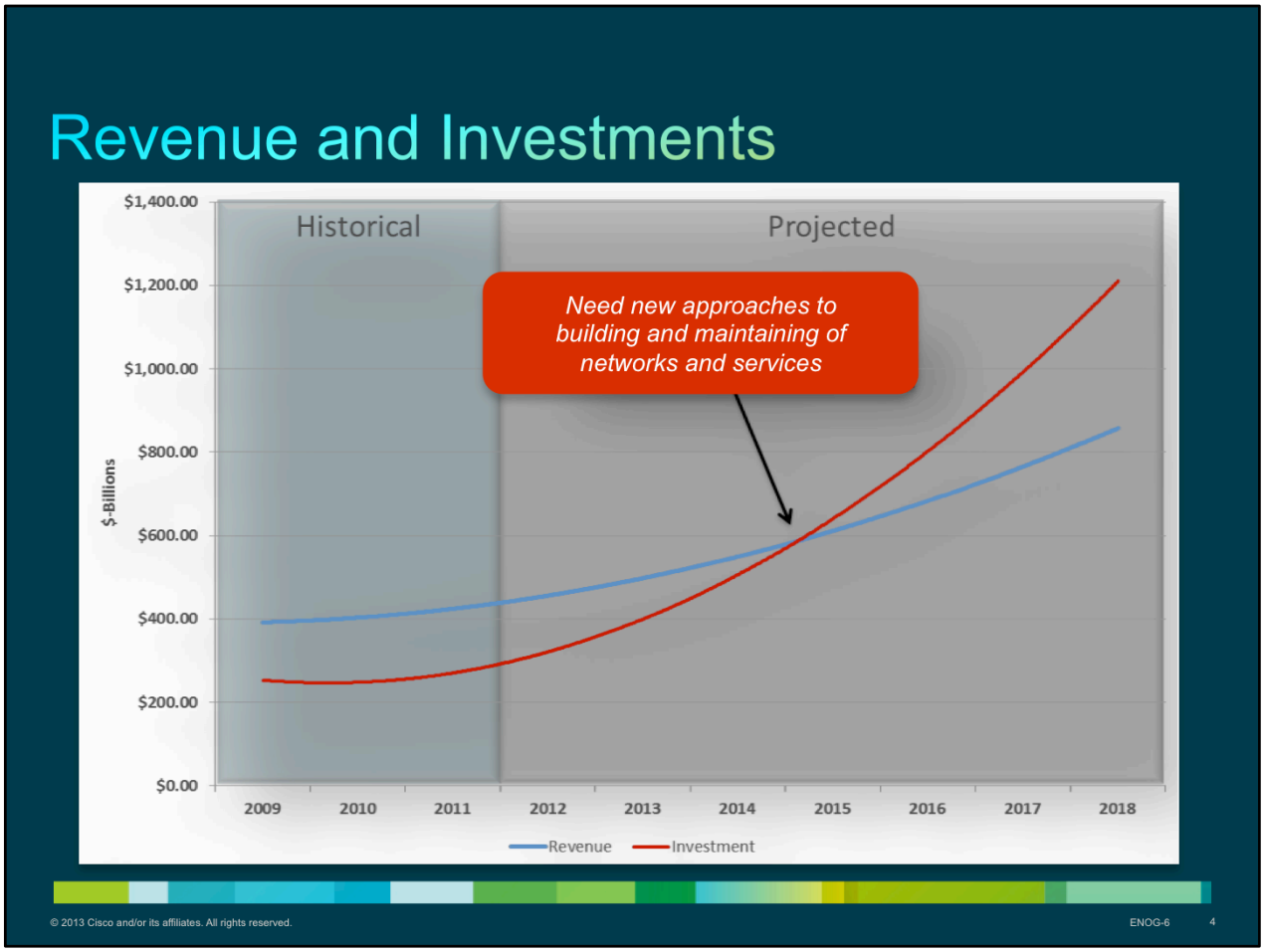
Мобильные абоненты и устройства (M2M коммуникации) и мобильные (облачные) приложения создают определенную непредсказуемость в матрице трафика, которая, при современных пропускных способностях и существующей тенденции роста трафика, становится существенным фактором нестабильности предоставления услуги. Также, нестабильность предоставления услуг возникает в точках предоставления сервиса (BNG, сервисные платформы видео, проч.), поскольку прогнозирование нагрузки может быть выполнено только с некоторой долей погрешности.

====

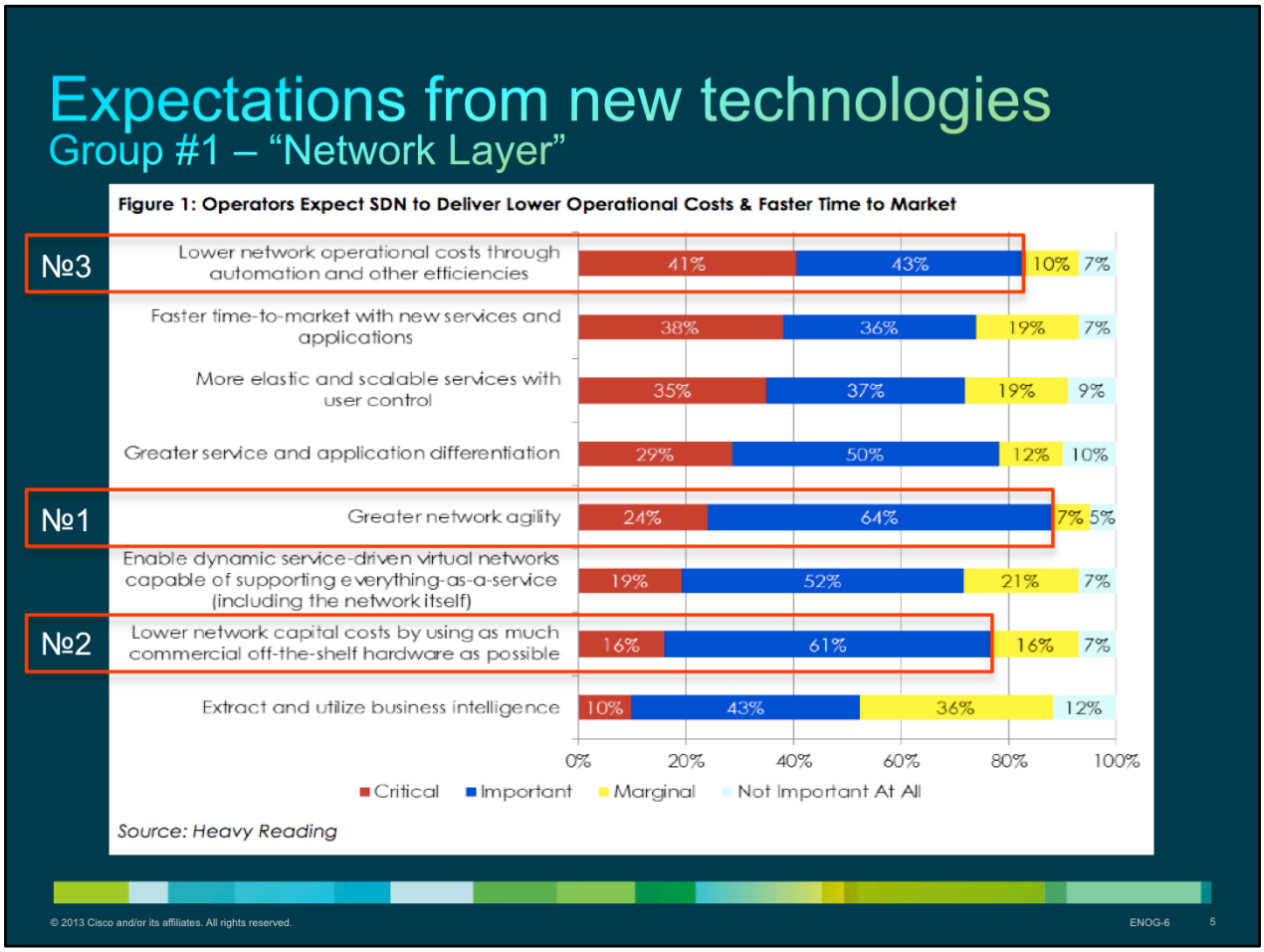
Mobility of subscribers (3G/LTE/WiFi on-board), devices (M2M) and application (cloud-aware) create instability in traffic's matrix. Keeping in mind current traffic volume and growth trends, it makes it costly to pre-provision network for all possible scenarios of traffic distribution, including both bandwidth allocation in packet and optical transport networks and resources allocation in service nodes (like BNG, mobile gateways, etc), since any traffic prediction can't be done with proper accuracy.

Нынешний подход к проектированию и эксплуатации сети становится неприемлемым – необходимость быстро подстраиваться под изменение матрицы трафика и под изменение нагрузки на сервисные узлы приводит к закладыванию излишней пропускной полосы каналов и мощности сервисных узлов (капитальные затраты). Кроме того, изменение конфигурации сети выполняется в полу-ручном режиме (что-то может быть сделано средствами, например, MPLS TE, но новые TE-туннели необходимо проектировать и конфигурировать вручную) – что существенно влияет на операционные затраты и снижает общий уровень SLA (время адаптации сети под новое состояние).
======
So, traditional approach doesn't do the job, since

- leads to overprovisioning of network and computing resources (higher CapEX)
- doesn't meet SLA requirements (missed profit)
- requires more resources (people, management systems, etc) to quickly respond to challenges (higher OpEX)

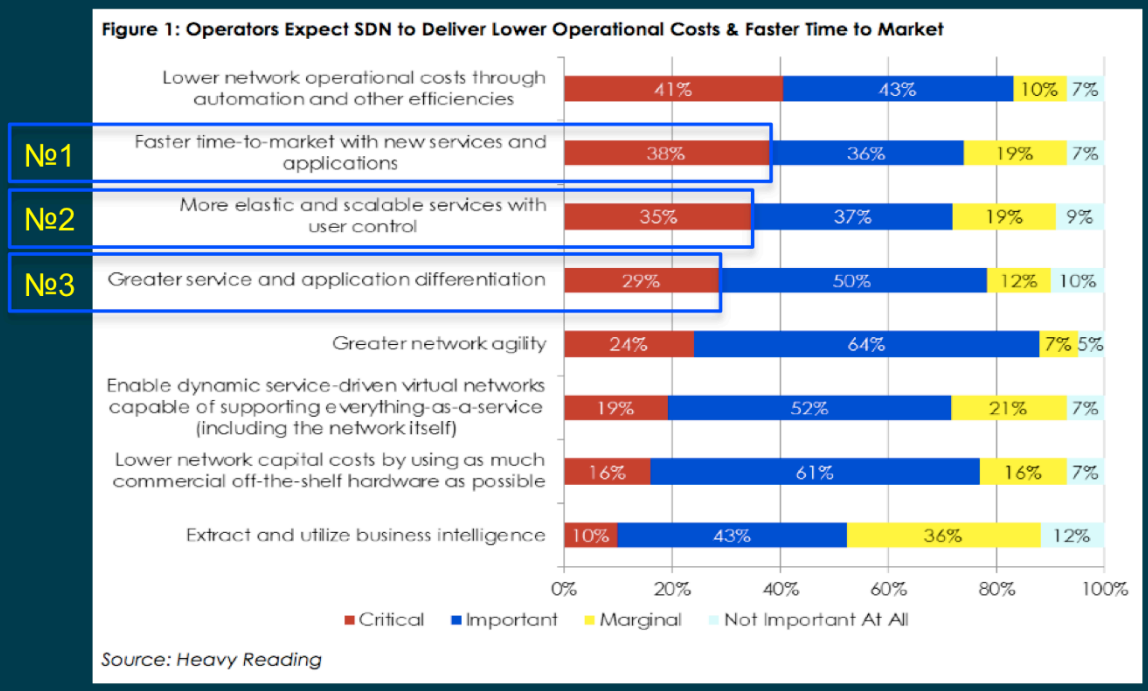


По оси Y – суммарные оценочные затраты мировой индустрии SP на расширение сети, в миллиардах долларов
При сохранении существующей схемы развития сетей, в 2015 году произойдет коллапс – доходы операторов превысят с объемом требуемых инвестиций.

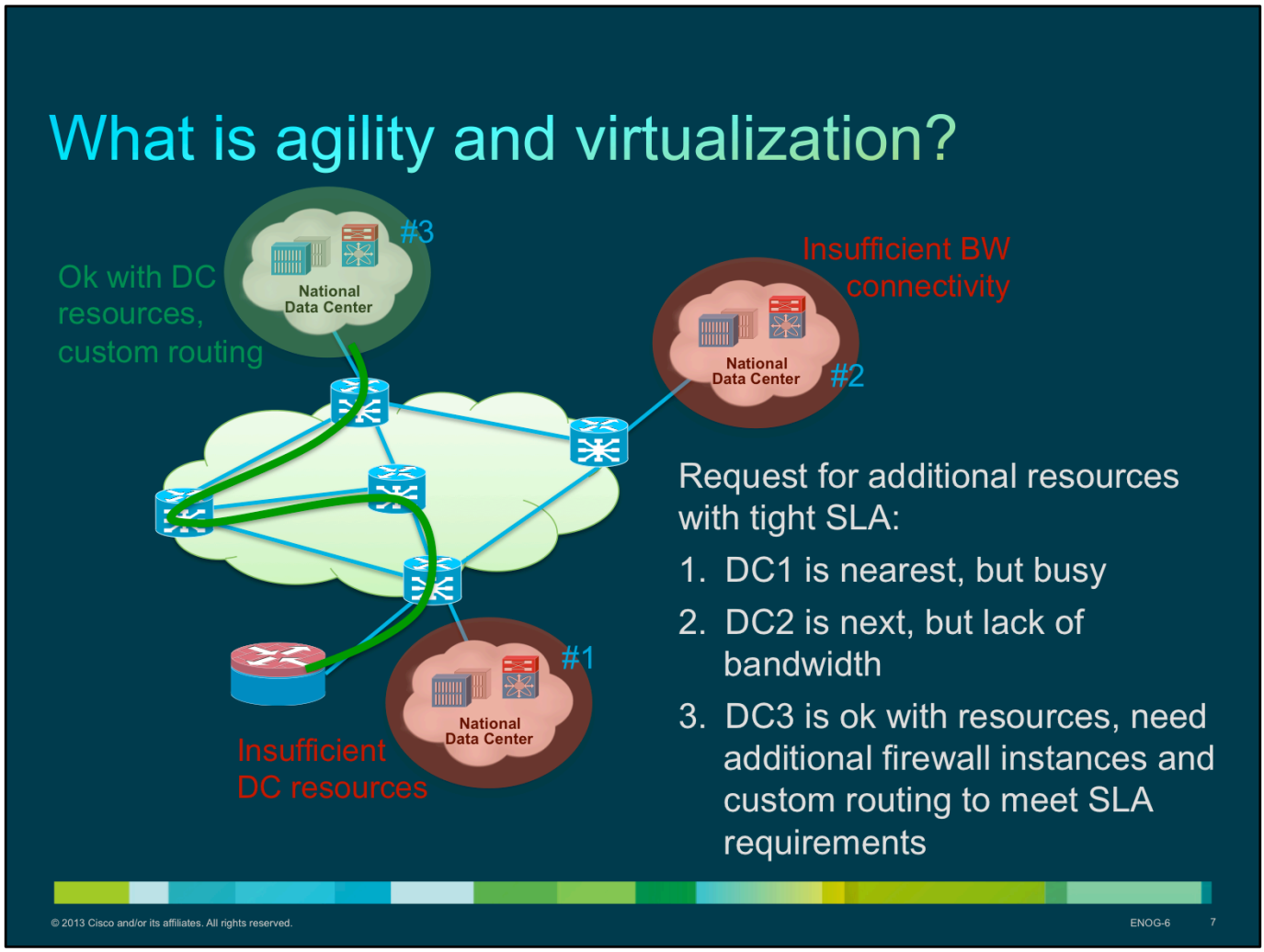


Опрос Heavy Readings об ожиданиях от нового.
Первая группа – ожидания от сетевого уровня.
Вторая группа – ожидания от уровня услуг.
=====
Heavy Readings report
First group – expectations from network layer
Second group – expectations from service layer

Expectations from new technologies
Group #2 – "Service Layer"



Опрос Heavy Readings об ожиданиях от нового.
Первая группа – ожидания от сетевого уровня.
Вторая группа – ожидания от уровня услуг.
=====
Heavy Readings report
First group – expectations from network layer
Second group – expectations from service layer



Например – клиент запрашивает дополнительные ресурсы с своем private cloud, расположенном в операторском датацентре. Выбор подходящего датацентра осуществляется по факту наличия требуемых ресурсов, наличия достаточной мощности сервисных функций (например, функций защиты от DoS) и соответствия SLA связи между офисом клиента и его private cloud.

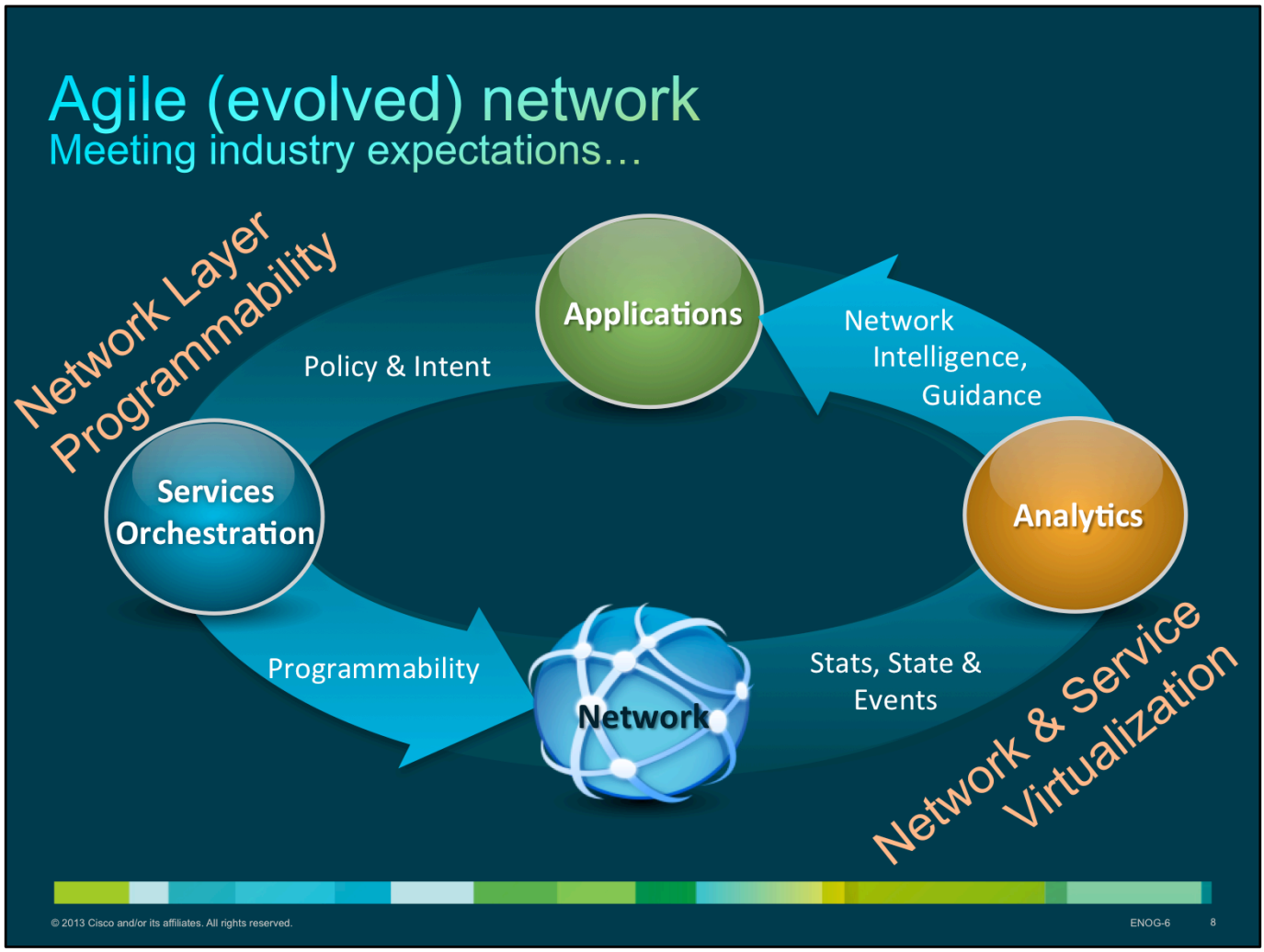
Для выполнения этой задачи требуется задействовать три компонента управления:

1) перенос частного облака клиента в другой датацентр (выполняется, например, посредством Openstack)
2) перенос сервисных функций (напр. DoS protection и firewall) к частному облаку (реализация т.н. Network Functions Virtualization)
3) переконфигурирование сети для соответствия требованиям SLA между офисом и частным облаком (задача для WAN Controller)

======

For example – customer asks for more resources in his private cloud. Network need to choose right place, which will provide both computing resources, service functions (e.g. DDoS protection) and network SLA. To get things done, operator need to orchestrate different activities:

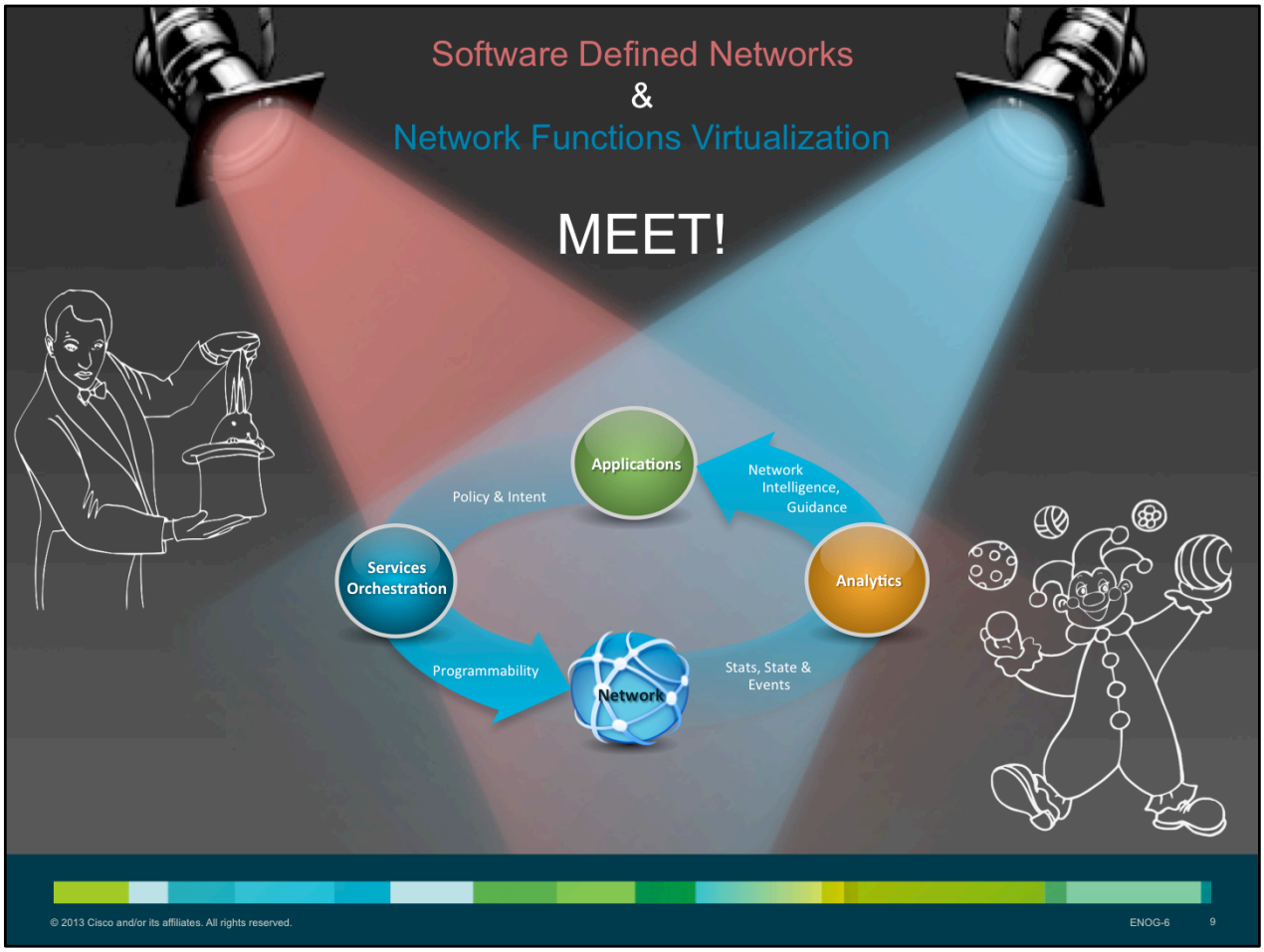
1) computing and storage resources mobility (can be done by Openstack)
2) service mobility (DDoS protection – this is about NfV)
3) network reconfiguration to meet SLA requirements



Хорошо было бы, чтобы сеть каким-либо образом сама перестраивалась, в соответствии с цепочкой «сбор статистики, состояния и событий» – «анализ» – «выработка решения» – «адаптация существущей конфигурации» – «конфигурирование сети»

=======

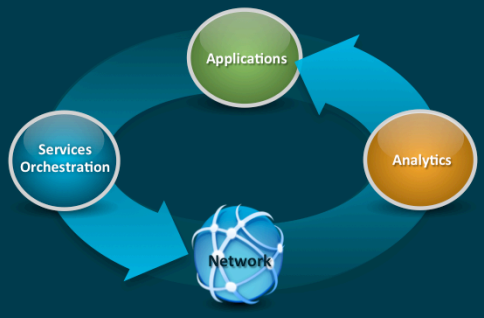
It's good to automate network configuration by building chain of events like "gather information" – "analyze it" – "make decision" – "create solution" – "configure network"



Now! There!! For you only!!! World's superstars!!!! Me-e-e-e-et! ☺

Network Functions Virtualization (NfV)

- Industry initiative (under ETSI)
- Moving network functions (Security, xGSN, DPI, NAT, BRAS, SBC, …) from specialized appliances to commoditized, x86-based architecture
- More @ http://en.wikipedia.org/wiki/Network_Functions_Virtualization

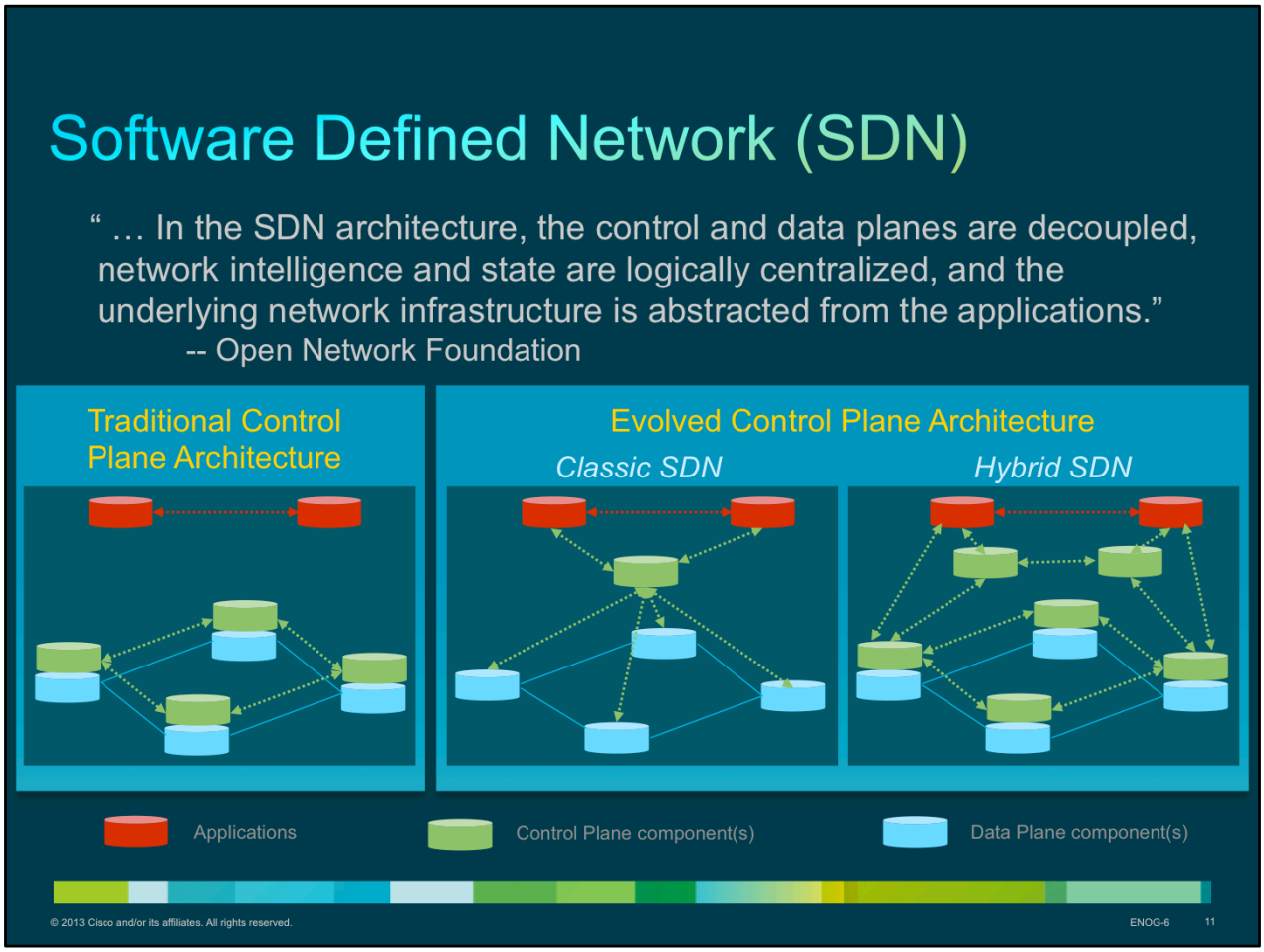


Key benefits of NfV:

- Virtualization decrease CapEx due to
  - ✓ optimal resources usage (multiplexing)
  - ✓ increasing equipment lifecycle (reuse)
- Faster Time to Revenue for new services (S/W development vs H/W appliances)
- Scalability and Distribution of Services
- OpEx decrease by automations and orchestration between service layer and network layer
- New revenue sources (deployment time, services bundling, …)

ключевые преимущества –

- сокращение капитальных затрат:
  - виртуализация позволяет мультиплексировать услуги в одном шасси, т.е. сокращается количество физического оборудования, энергопотребление и теплоотдача, нужно меньше места в стойках, проч. проч. проч.
  - если сервис потребляет больше ресурсов чем есть – его можно перенести на другой сервер, а освободившиеся ресурсы на сервере отдать другому сервису
- уменьшение времени вывода новой услуги на рынок (не требуется ждать нового оборудования – только новое программное обеспечение)
- быстрое масштабирование и дистрибуция услуги (виртуальные машины могут быть запущены, добавлены и отключены по требованию)
- уменьшение операционных затрат путем автоматизации управления сервисами и сетевой/ транспортной инфраструктурой (automation and orchestration)



Гибридная модель SDN отличается от классической модели следующими моментами:

- эволюция, а не революция. Существующая операторская сеть настроена и работает. Зачем ломать то, что работает и перестраивать все с нуля? Это неизбежно приведет к проблемам эксплуатации.
- есть целый класс задач, с которыми распределенная модель справляется намного лучше, чем централизованная, например - восстановление в аварийных ситуациях. С другой стороны, централизованная модель намного более подходит для сложных задач оптимизации сети. Таким образом, гибридная модель берет все лучшее от централизованной и классической распределенной модели.
- сохранение инвестиций. Операторы платили за «умное» оборудование, интеграцию, обучение - стоит-ли от этого отказываться? Гибридная модель позволяет сохранить существующие операционные процессы в компании неизменными.

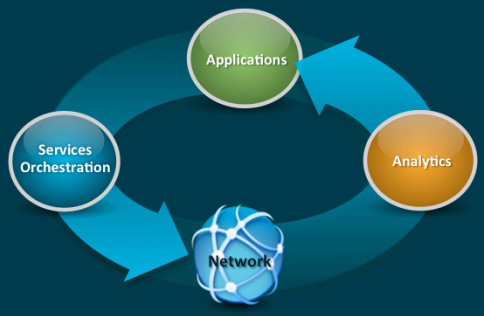
======

Classic SDN isn't scalable solution – there are plenty of cases where distributed model works better (e.g. fault recovery). Cisco's approach is Hybrid SDN – operator can continue to use what he already built (it's an evolutionary not revolutionary way, which will prevent existing technical and operational processes and will provide investments savings – both for hardware and staff), adding more intelligence by using external "brain" (applications).

Software Defined Network (SDN)

" … In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications."

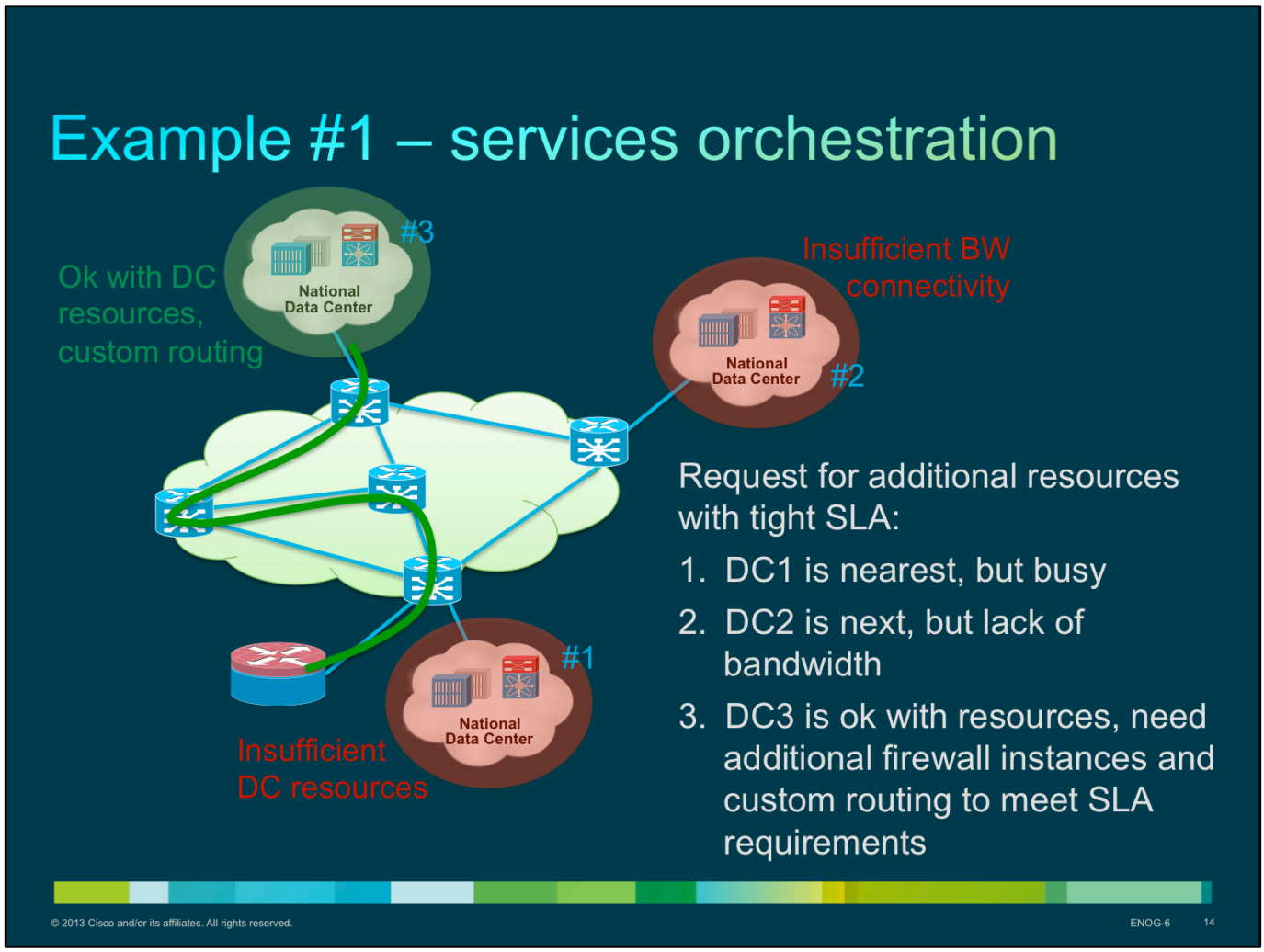
-- Open Network Foundation



Key benefit of SDN – it is enabler for:

- significant CapEx decrease (multilayer optimization of network resources' usage)
- significant OpEx decrease (automation of network maintenance and management)
- new revenue sources (improved SLA, services bundling, …)

SDN & NfV Use Cases



Например – клиент запрашивает дополнительные ресурсы с своем private cloud, расположенном в операторском датацентре. Выбор подходящего датацентра осуществляется по факту наличия требуемых ресурсов, наличия достаточной мощности сервисных функций (например, функций защиты от DoS) и соответствия SLA связи между офисом клиента и его private cloud.

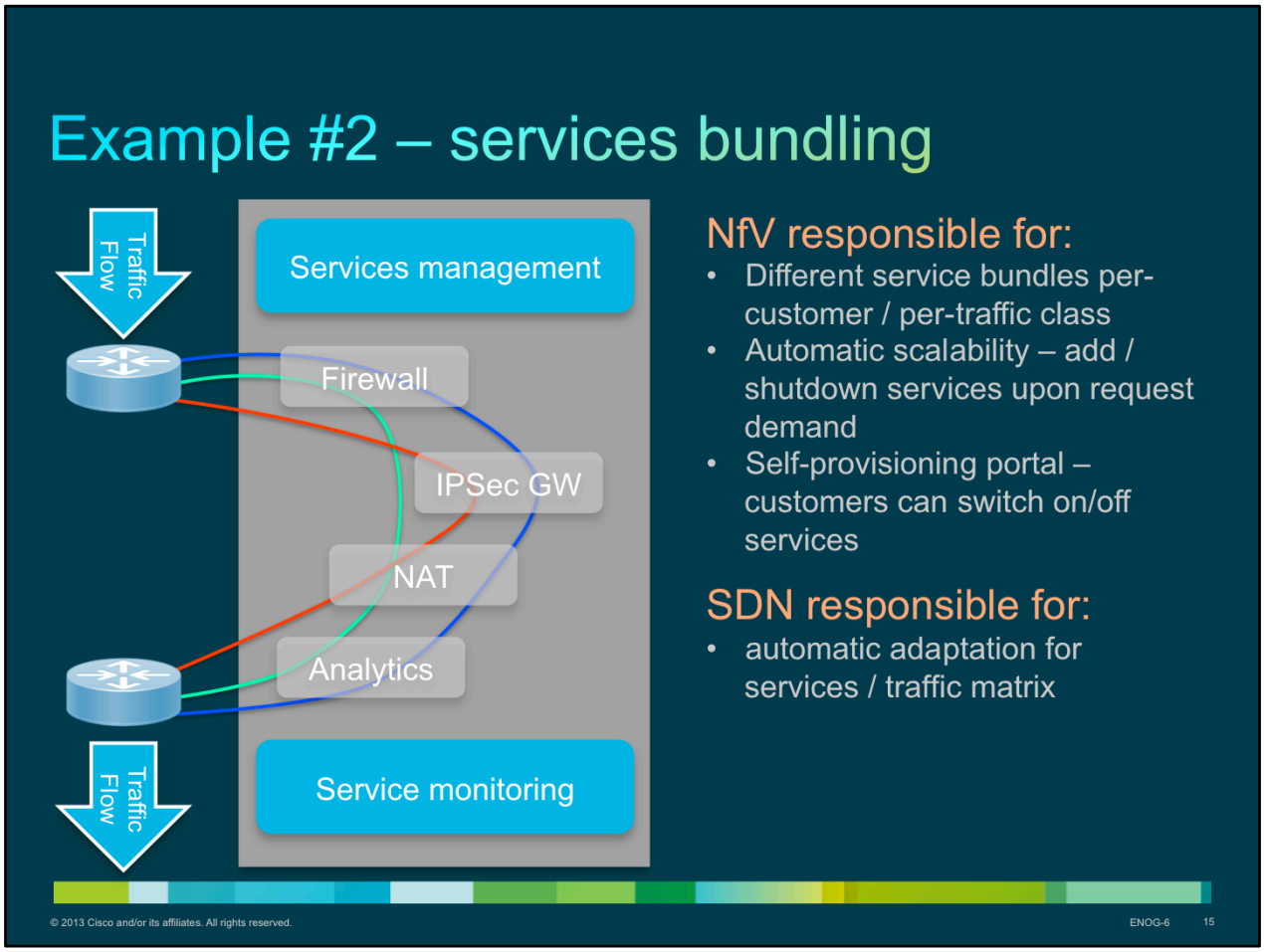
Для выполнения этой задачи требуется задействовать три компонента управления:

1) перенос частного облака клиента в другой датацентр (выполняется, например, посредством Openstack)
2) перенос сервисных функций (напр. DoS protection и firewall) к частному облаку (реализация т.н. Network Functions Virtualization)
3) переконфигурирование сети для соответствия требованиям SLA между офисом и частным облаком (задача для WAN Controller)

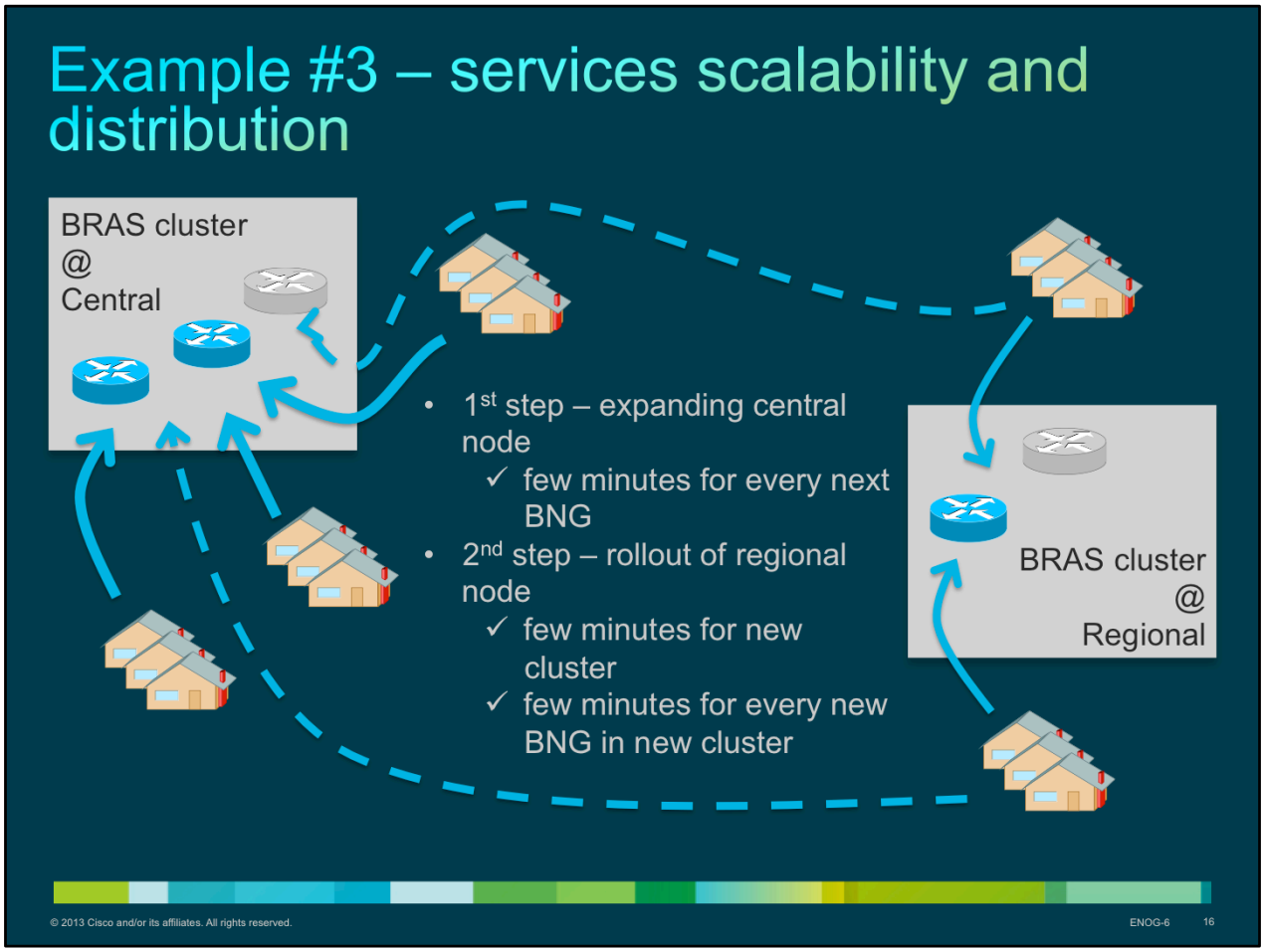
======

For example – customer asks for more resources in his private cloud. Network need to choose right place, which will provide both computing resources, service functions (e.g. DDoS protection) and network SLA. To get things done, operator need to orchestrate different activities:

1) computing and storage resources mobility (can be done by Openstack)
2) service mobility (DDoS protection – this is about NfV)
3) network reconfiguration to meet SLA requirements

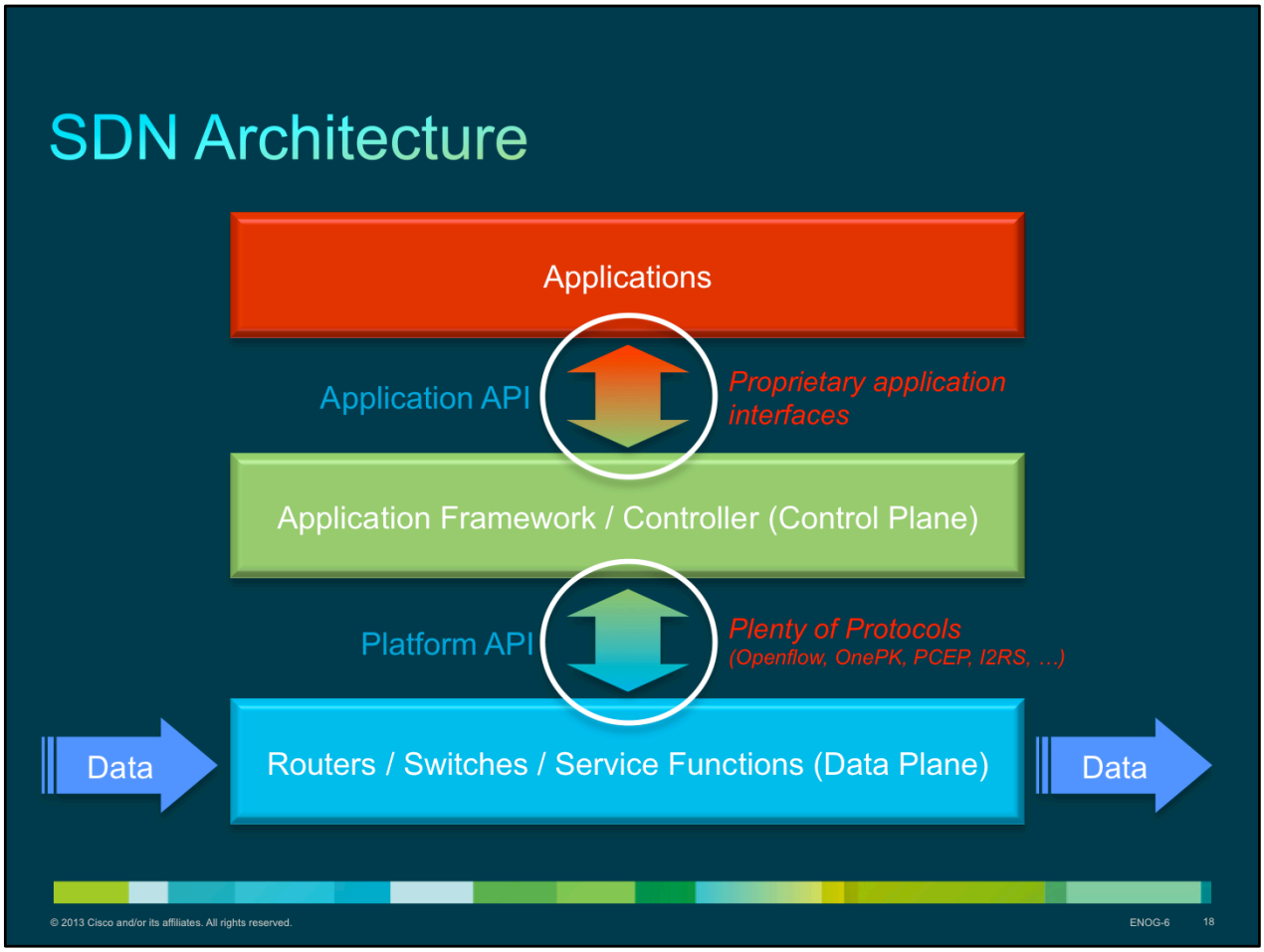


Services bundling allows to build quickly customized chains of services for customers, launching enough qty of virtual service nodes and adapting network for changed traffic matrix.

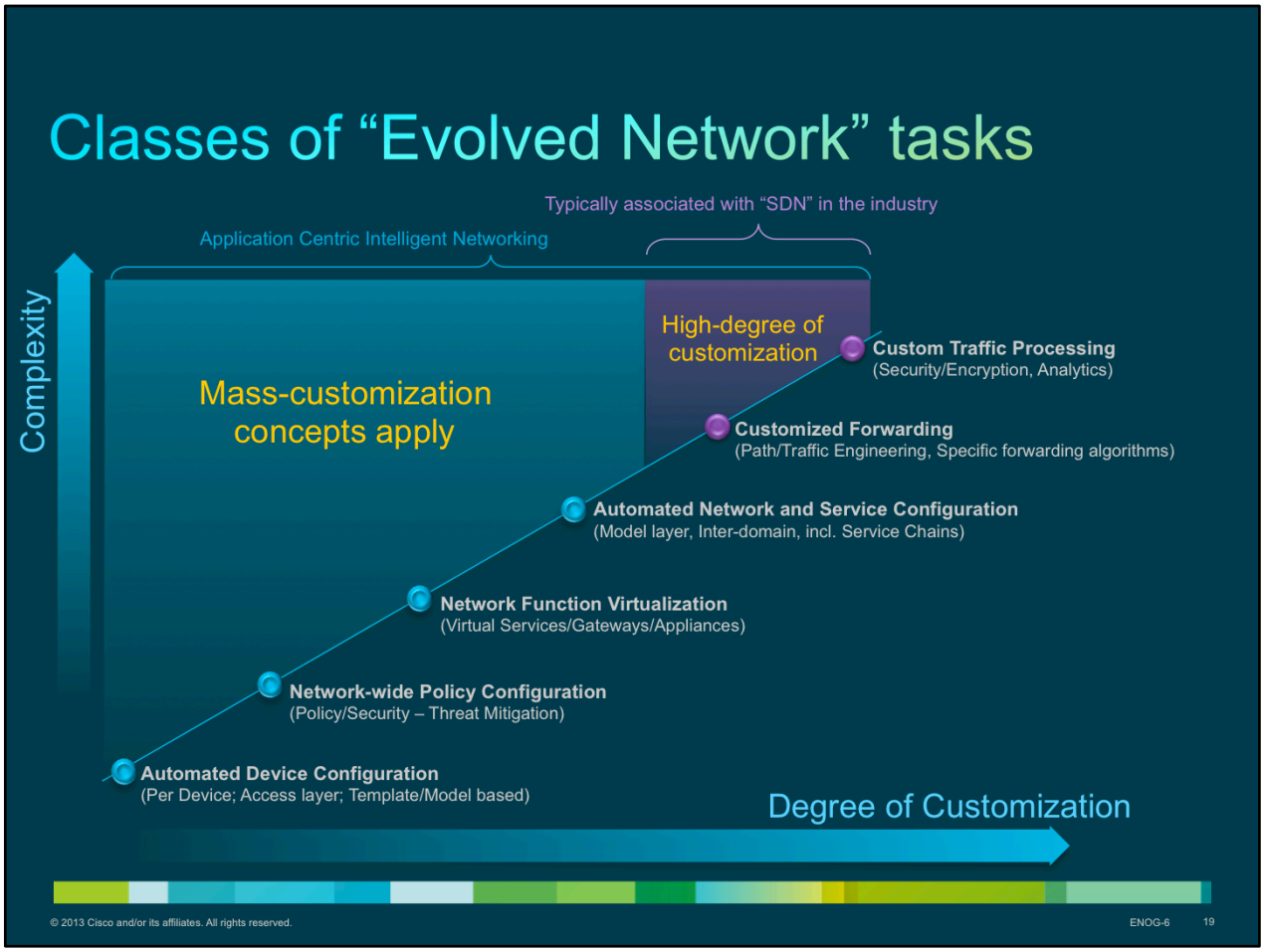


Again – distribution of services upon growth of customer base. Using NfV and SDN allow to do this in very fast way – just by launching new virtual service nodes (e.g. BNGs) closer to customers and reflecting changes on transport layer by using SDN.

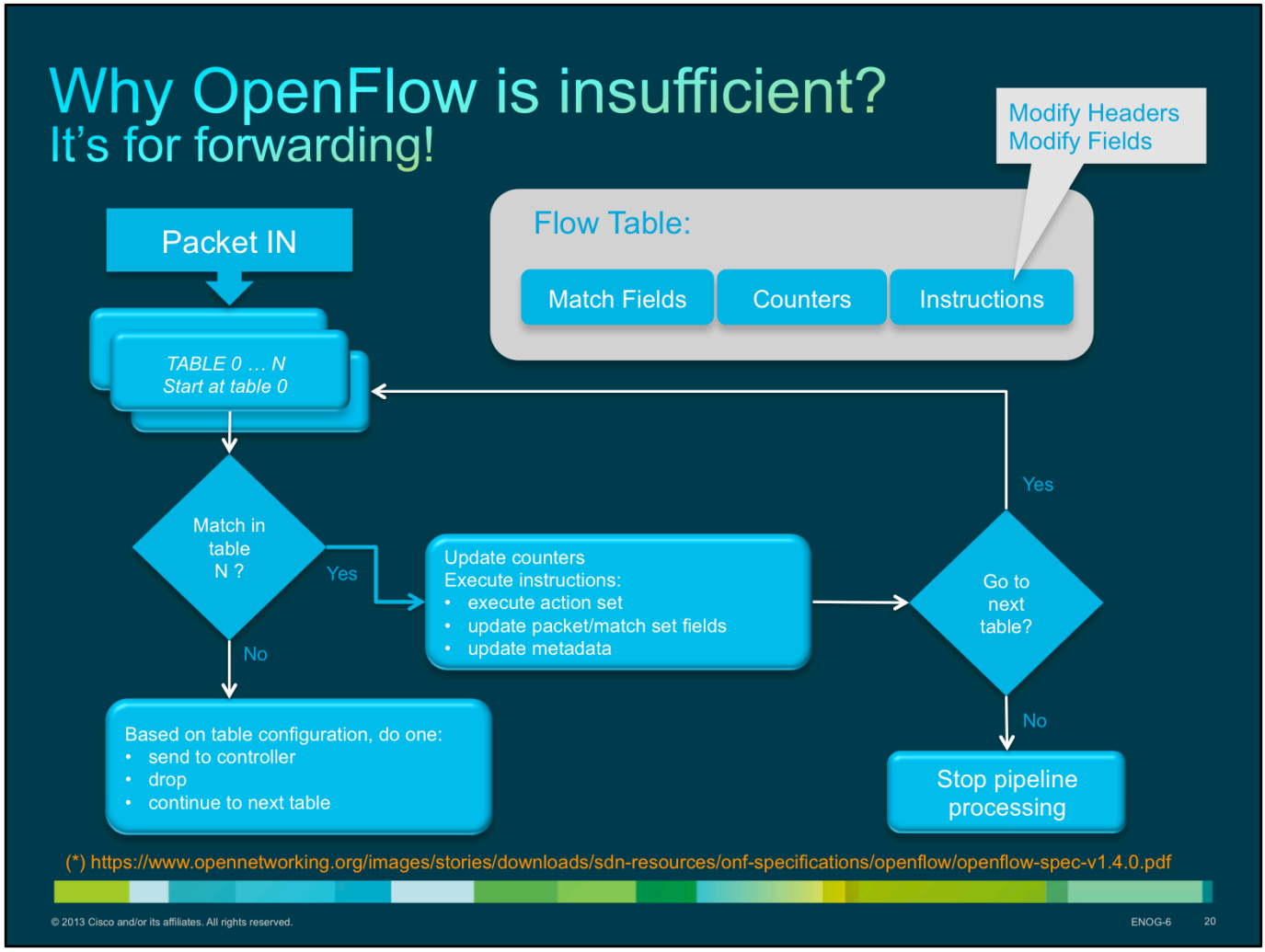
Architecture



Application-facing protocols are vendor's proprietary (ONF develops Openflow).
Plenty of network-facing protocols derived from plenty of different tasks, required to be done on the network (see next slide) (Openflow, at the moment, is just for managing of forwarding)



There are more tasks can be done or accompanied with SDN, than just custom forwarding and custom traffic processing – and these tasks require more than Openflow

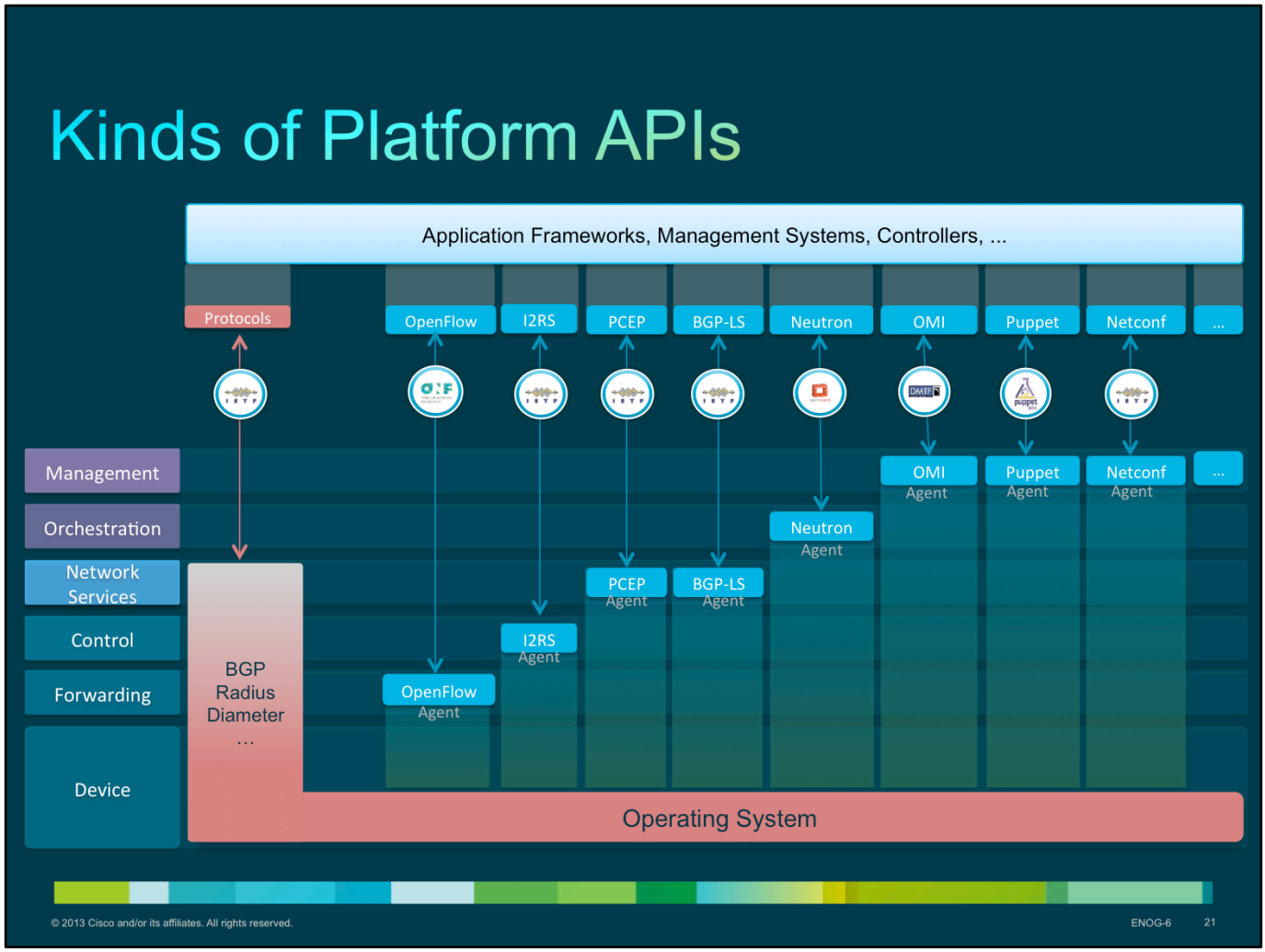


https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

Как видно, openflow специализирован под custom traffic routing и custom traffic forwarding, а остальные задачи решать неспособен.

===

As you can see, Openflow targeted for traffic forwarding and unable to deal with any other kinds of tasks.



Respecting different tasks, there are different interfaces to control network:
- I2RS to manage routing tables
- PCEP to optimize network topology
- Neutron to control virtual network nodes
- Netconf to manage nodes configuration
- etc

Welcome back to an engineerium?

Who

Controls

What ?

So well, uncertainty on application-facing side and lots of different protocols on network-facing side – will developers like it?

http://www.opendaylight.org/

OpenDayLight Controller

- Open source project formed by industry leaders and others under the Linux Foundation
- Goals:
  - ✓ Furthering the adoption and innovation of Software Defined Networking (SDN)
  - ✓ Accelerate real, deployable solutions for the industry:
    - Leveraging open source development best practices
    - Enable agile networks that can flexibly adapt to application requirements

Platinum

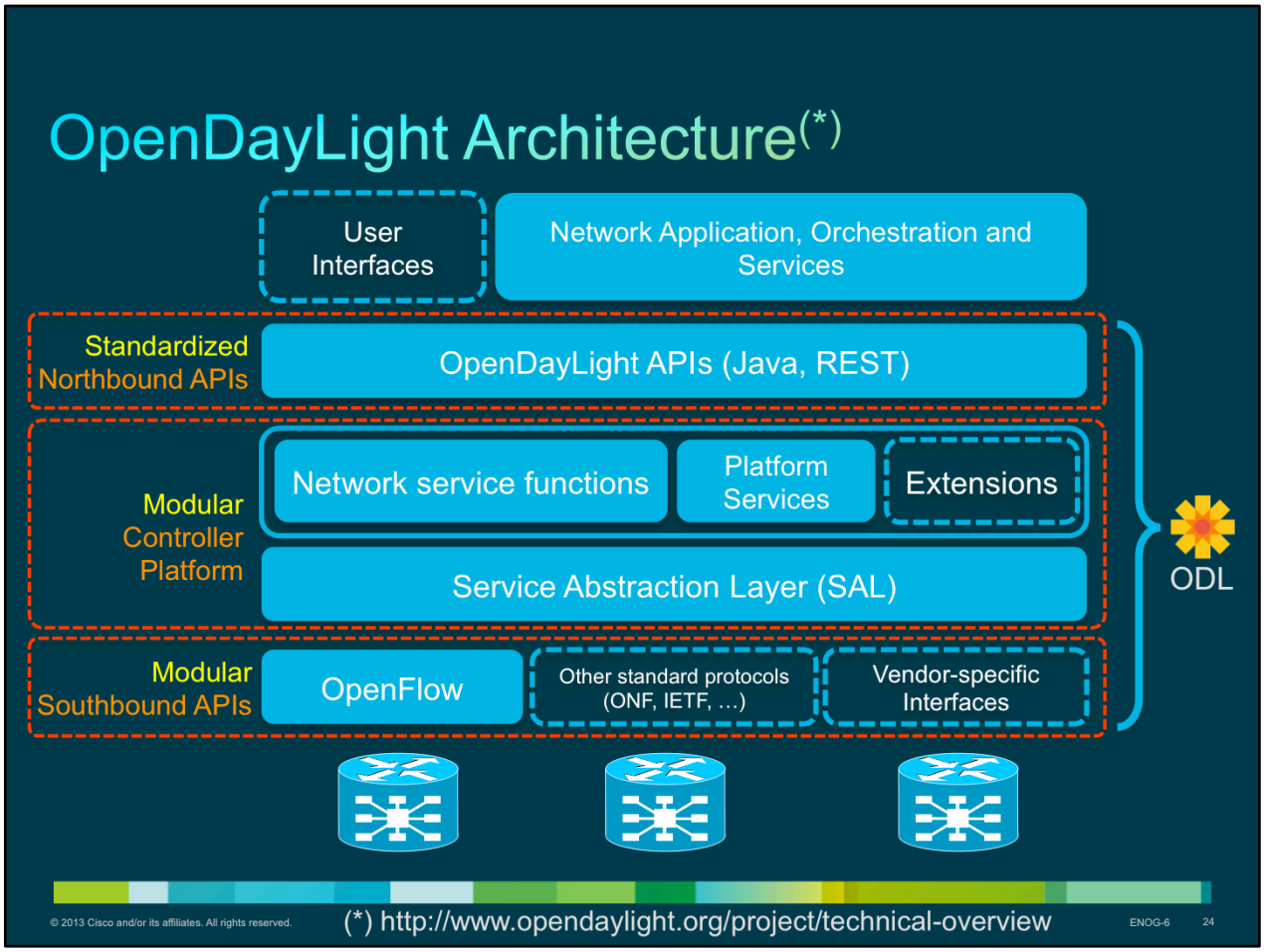
BROCADE, CISCO, CITRIX, ERICSSON, IBM, JUNIPER NETWORKS, Microsoft, redhat

Gold

NEC, vmware

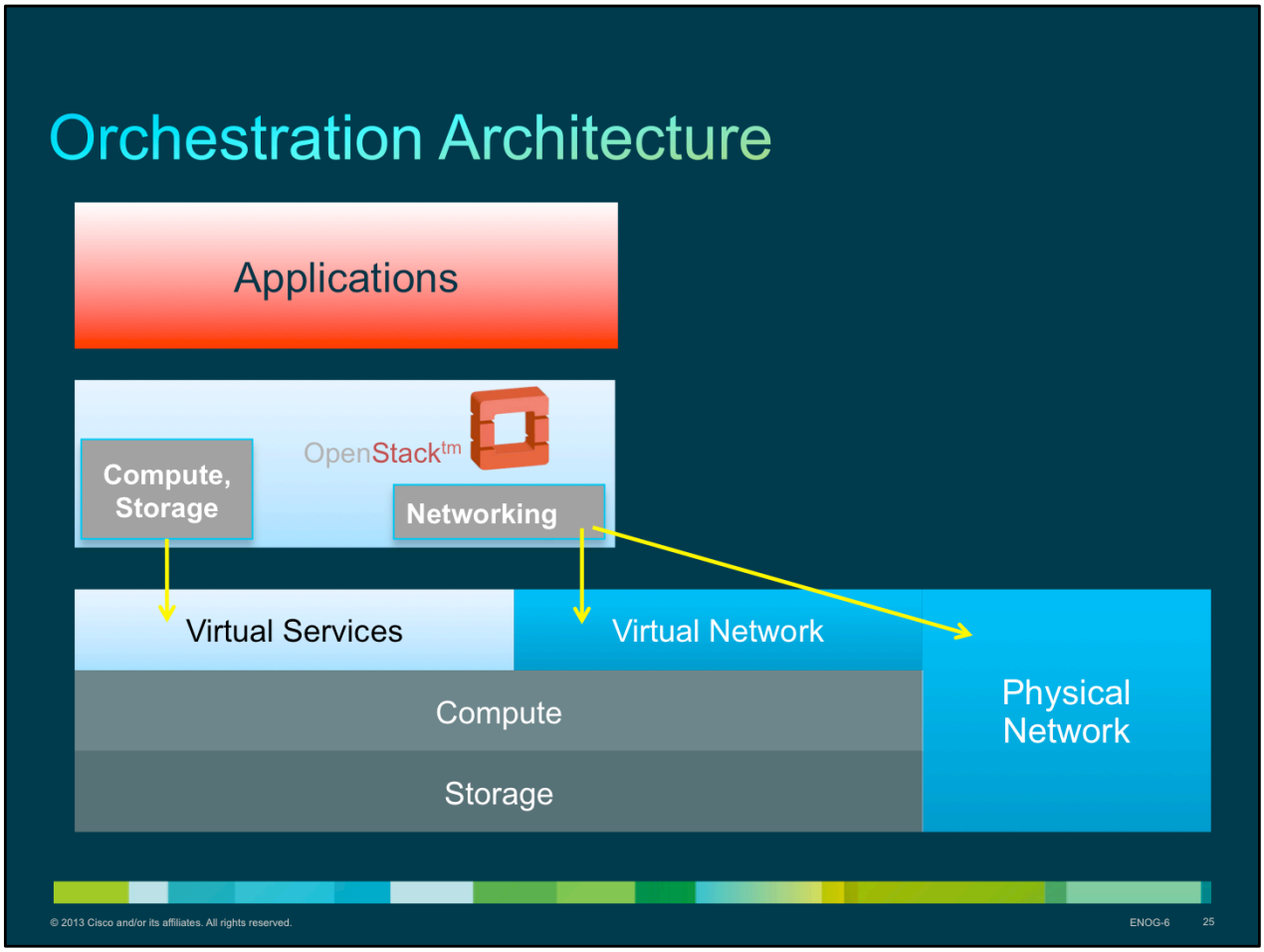
Silver

ADVA Optical Networking, ARISTA, ciena, CYAN, DELL, FUJITSU, pantheon, PLEXXI, PLUMgrid, guavus, hp, HUAWEI, INOCYBE TECHNOLOGIES, intel, nuagenetworks, radware, VERSA NETWORKS

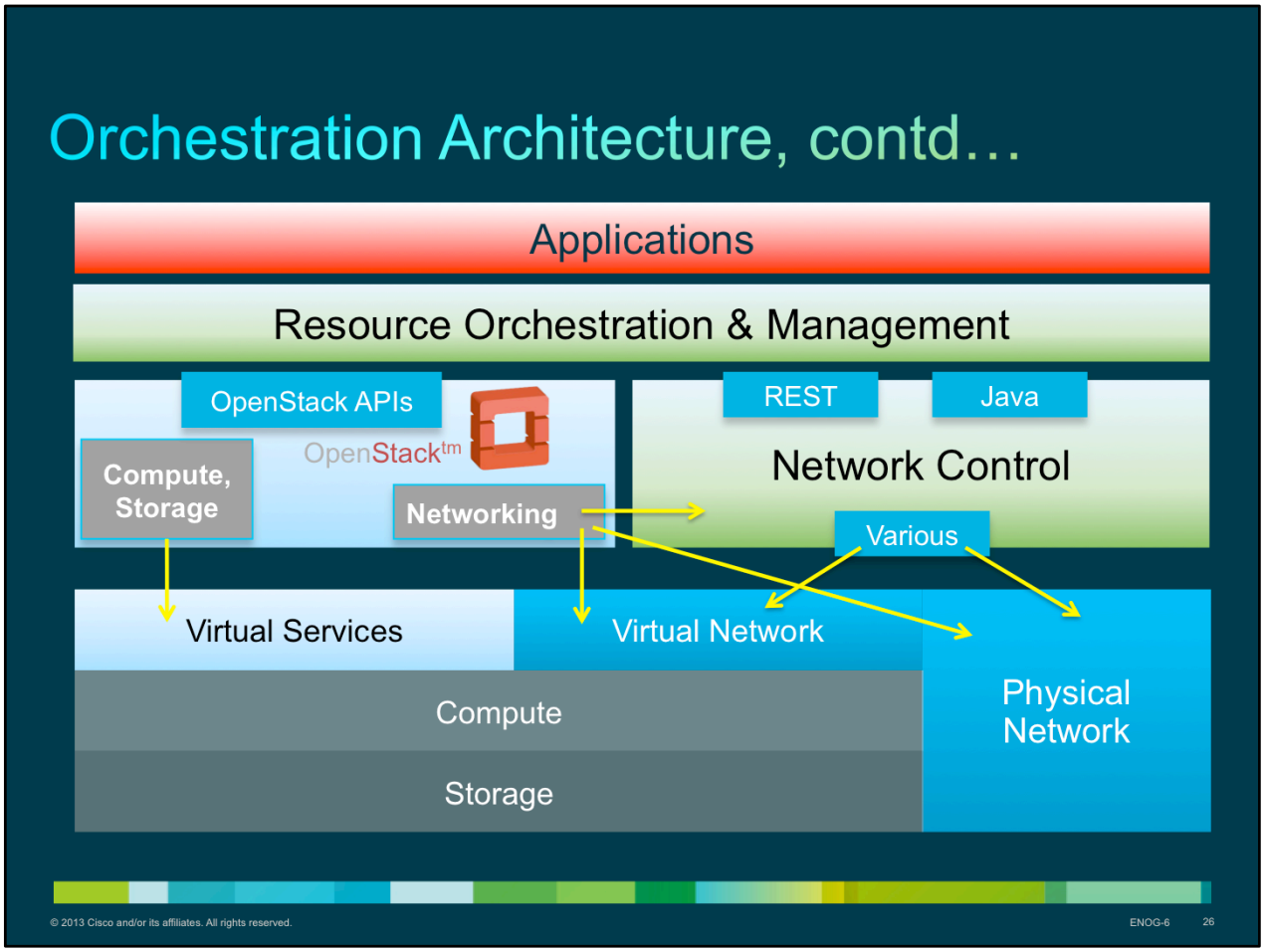


Modular architecture:

1) expandability
2) reliability (e.g. ISSU)
3) standardized NB (northbound, application-facing) interfaces lead to great simplification on development side
4) controller will choose SB (southbound, network-facing) protocol automagically, depending on task it need to do and capabilities of device which it need speak to

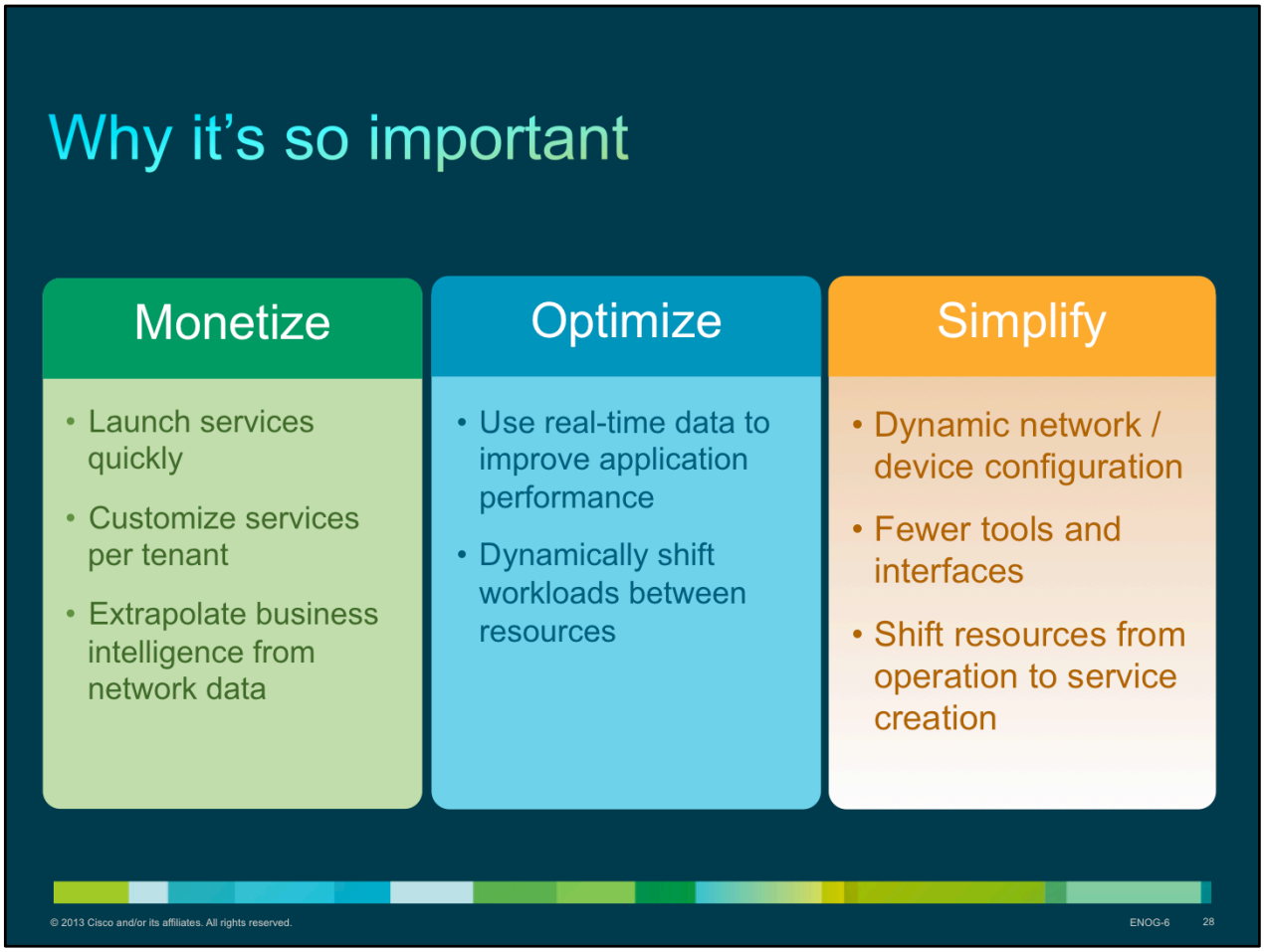


OpenStack Networking (Neutron, formerly Quantum) is a system for managing networks and IP addresses. Like other aspects of the cloud operating system, it can be used by administrators and users to increase the value of existing datacenter assets. OpenStack Networking ensures the network will not be the bottleneck or limiting factor in a cloud deployment and gives users real self-service, even over their network configurations. Administrators can take advantage of software-defined networking (SDN) technology like OpenFlow to allow for high levels of multi-tenancy and massive scale.



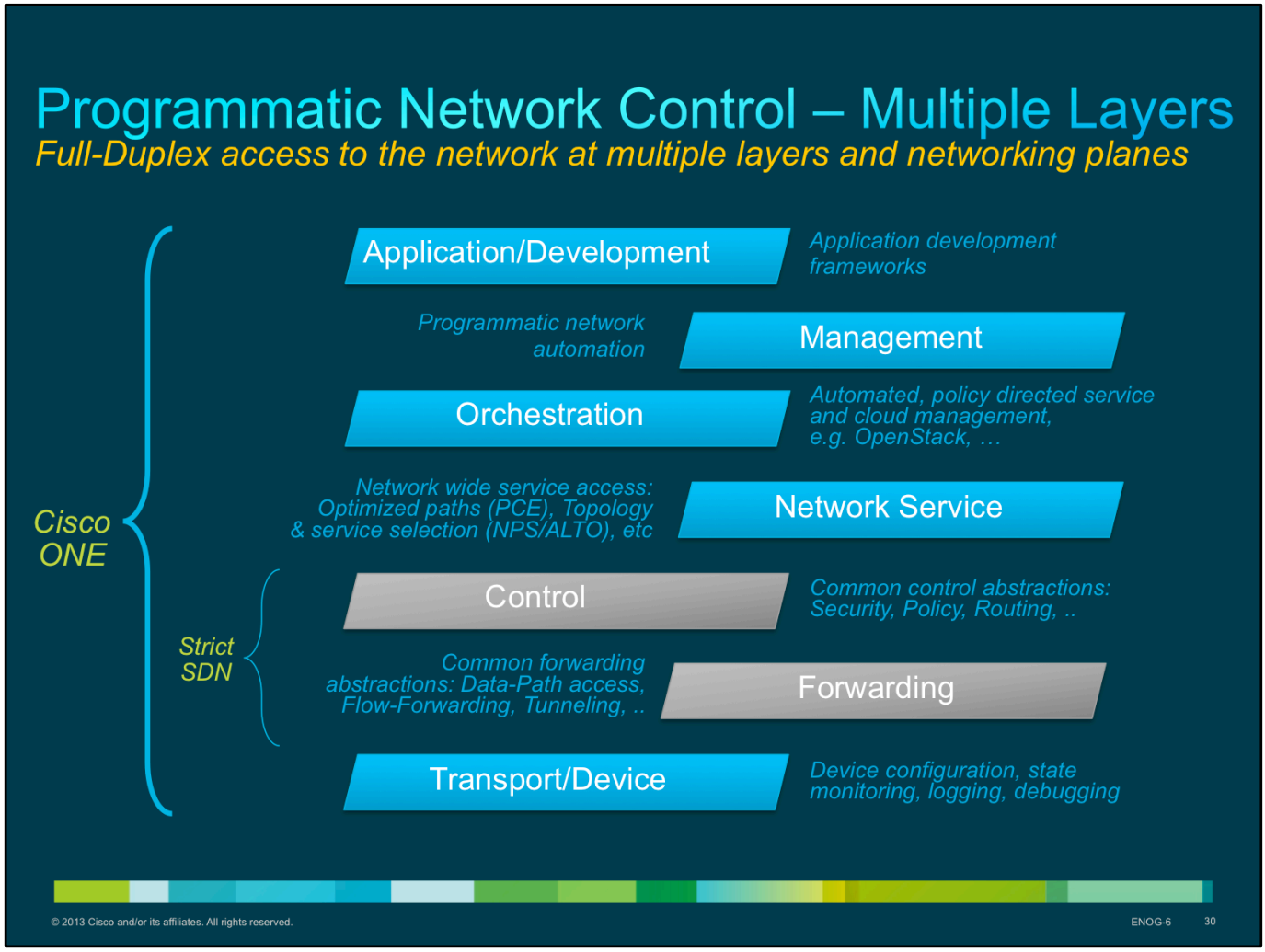
Cisco is investing in OpenStack, particularly – in Networking module (“Neutron”, former “Quantum”) to make it capable to interop with SDN.

In summary …

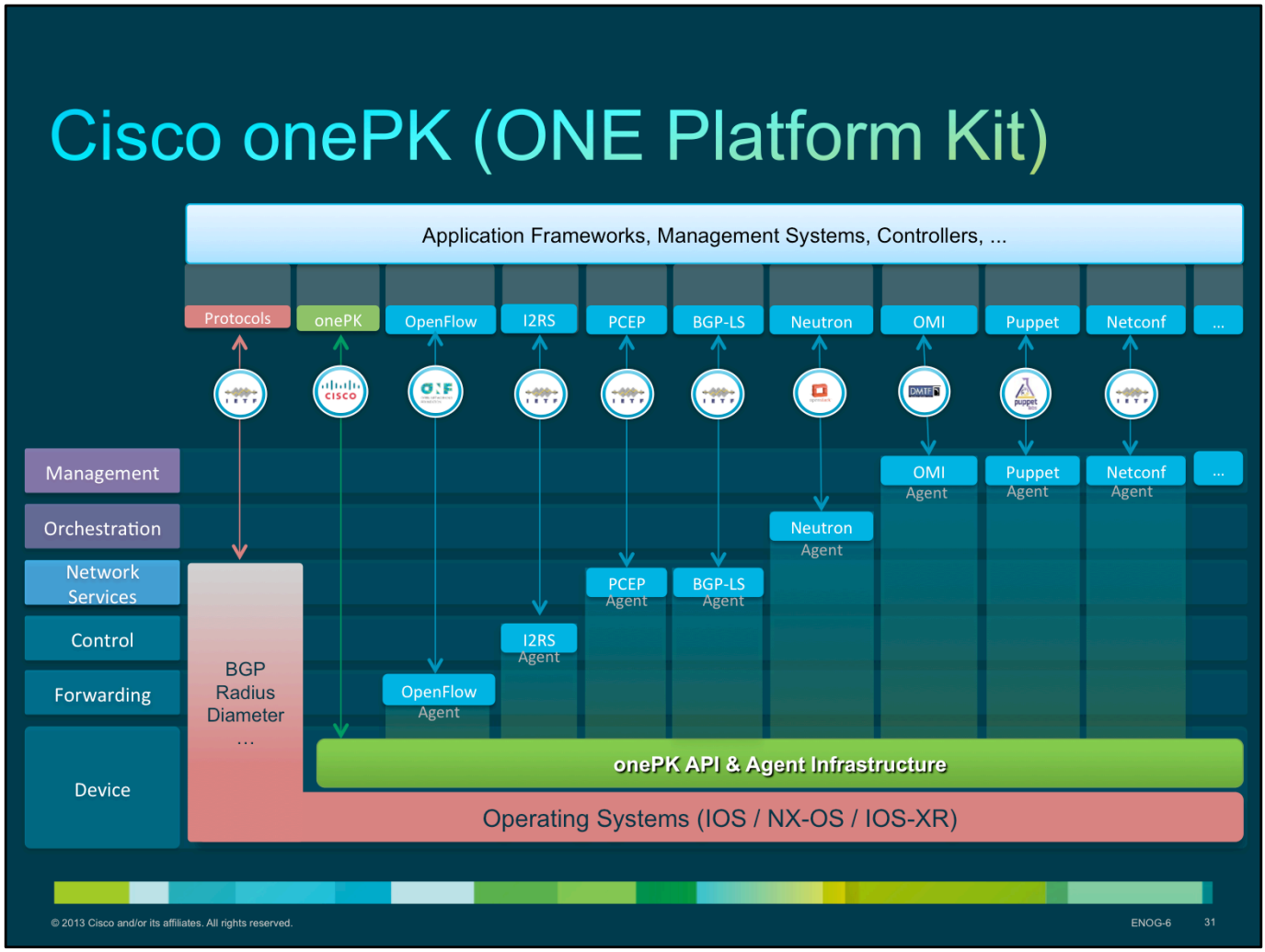


SDN and NfV – not service itself, but rather way for great optimization: automation of services' provisioning, adaptation of transport network, optimal resources usage, increasing performance of services, decreasing operational expenditures, etc.

Cisco ONE
Open Network Environment



When considering the objective of SDN then it can be interpreted incorrectly that the evolution only involves the control plane and data plane of the network. Network environments are more complicated that simply considering standalone applications, SDN Controllers and physical network hardware as shown in figure 4. An SDN environment needs to encompass the entire “network stack” from transport, to the management, to the automation up to the application layer. For SDN to help realise some of the current time to market issues then we need clear linkage between the application layers right down to the transport layers. This will include the development of multiple API’s and protocols from the transport/ Optical layer i.e. GMPLS to the network layer i.e. PCEP, BGP-LS, to the orchestration layer i.e. Open Stack, to the management layer i.e. REST API to the Application layer i.e. ALTO.

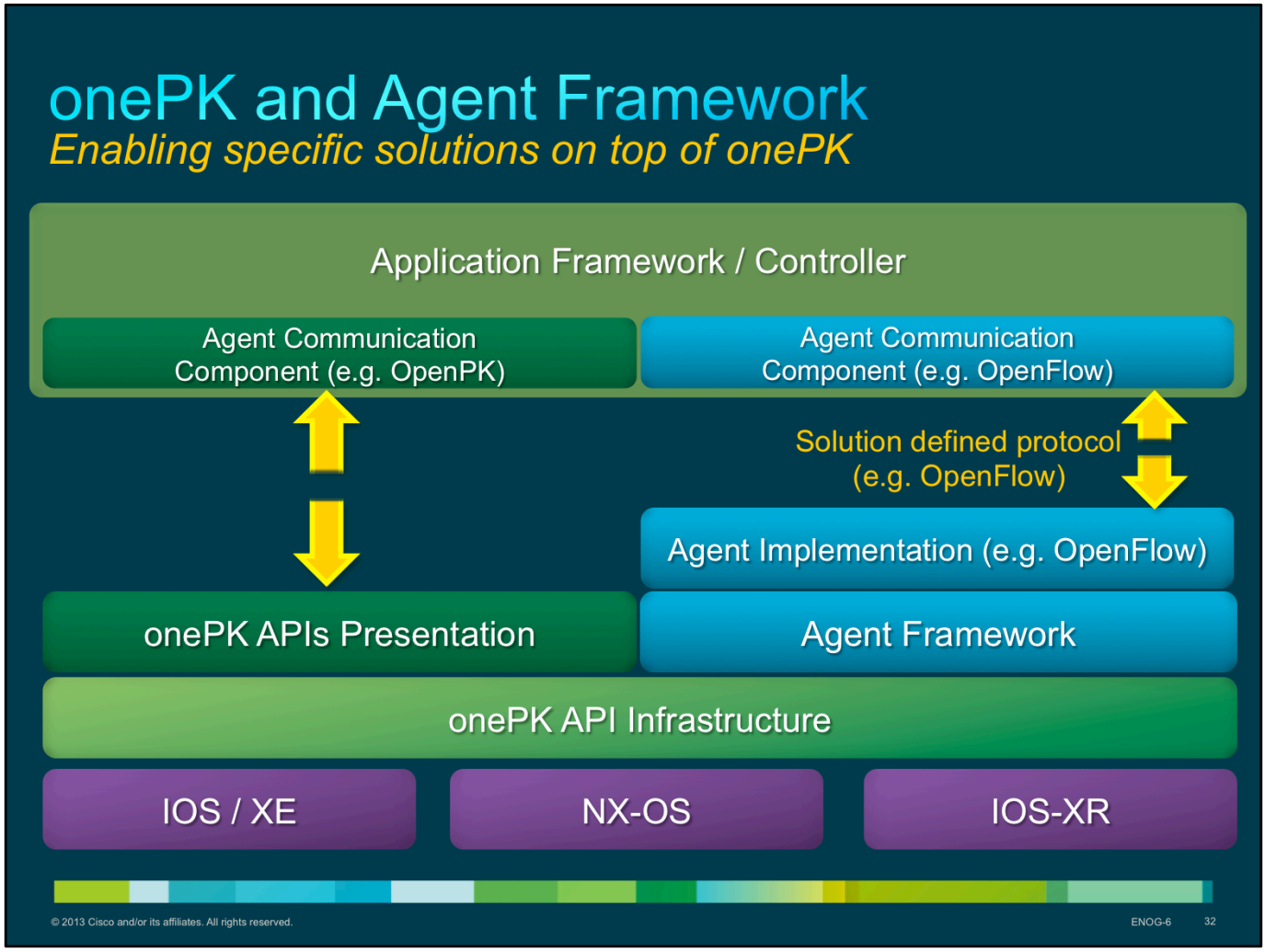


Cisco OnePK is programming infrastructure, implemented in Cisco equipment and can be used both directly (using corresponding API) and as basis to create corresponding agents for any of upper protocols.

=========

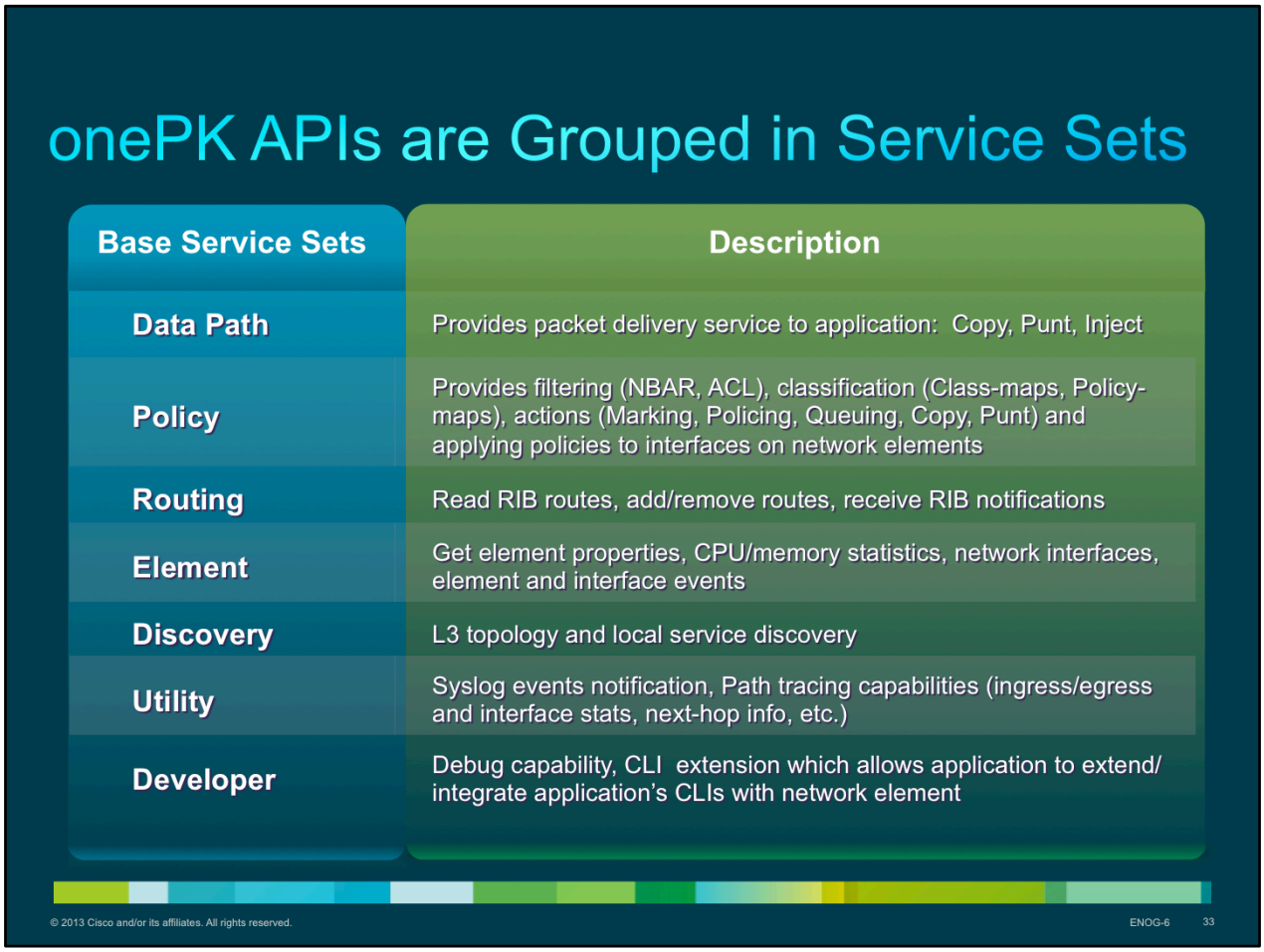
Respecting different tasks, there are different interfaces to control network:

- I2RS to manage routing tables
- PCEP to optimize network topology
- Neutron to control virtual network nodes
- Netconf to manage nodes configuration
- etc

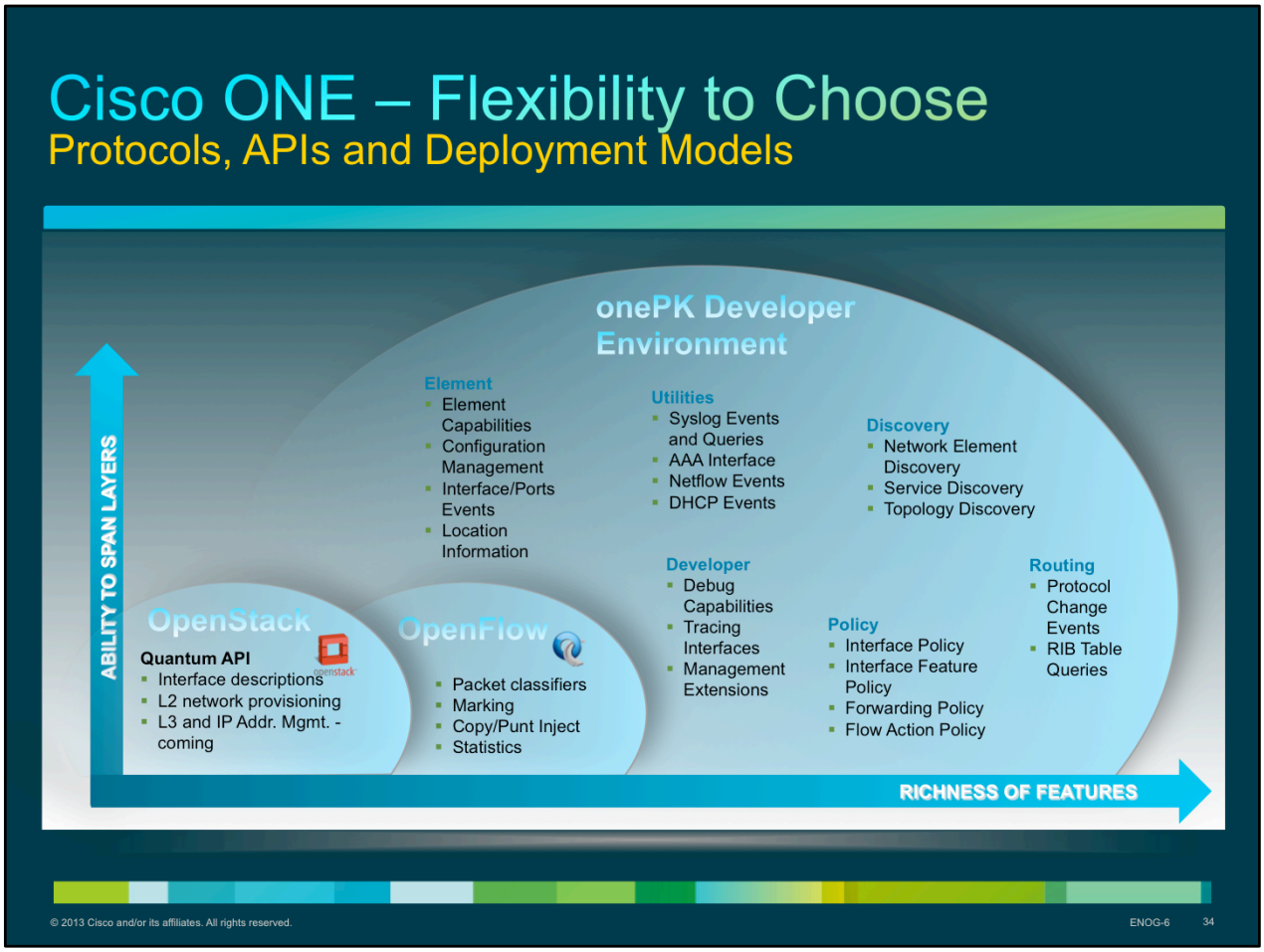


OnePK agent resides on equipment (switches, routers, etc) and interacts with application using OnePK or other protocols on top of the OnePK.

Modular architecture of OnePK Agent allows to support any protocols on the top. Here you see how support for Openflow is implemented in this architecture. Other protocols (I2RS, PCEP, BGP-LS, etc) can be implemented in similar way.

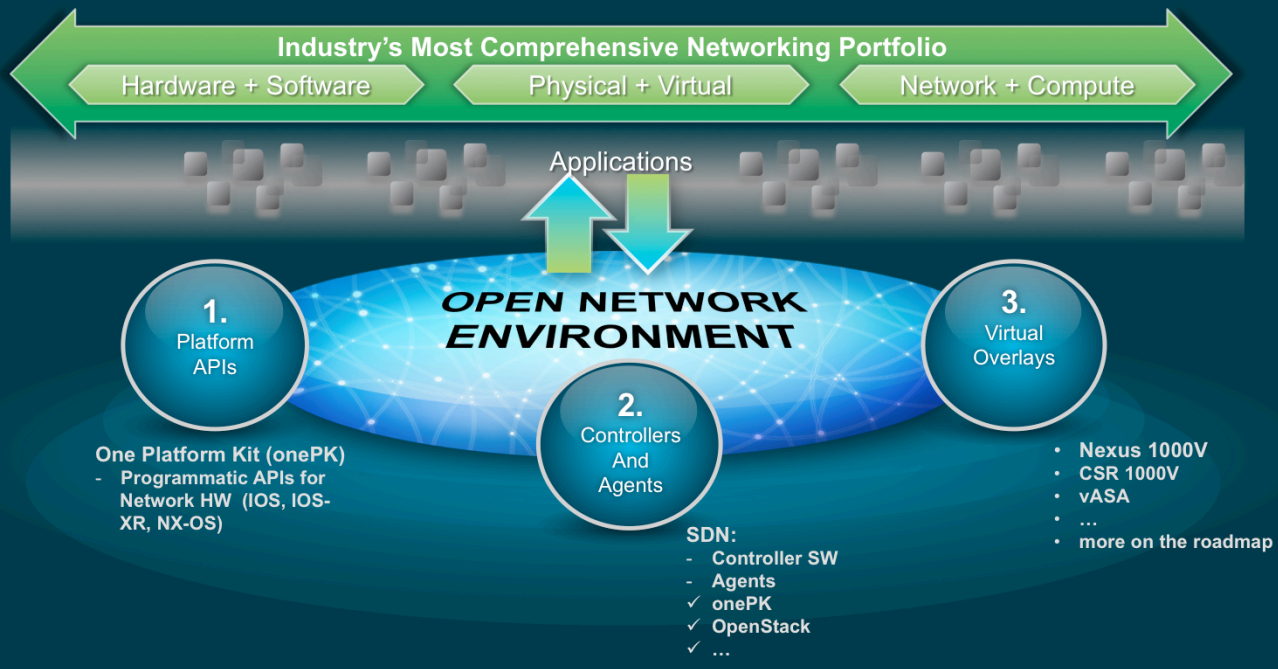


To be capable do most of network tasks, onePK consists of set of abstractions (APIs) to different kinds of OS's functionality (modify routing table, modify forwarding table, get/set parameters, read/change/write config, etc, etc, etc)



OnePK is Cisco's framework, built on top of network elements' (routers, switches, etc) operating system. It gives full programmatic access to entire set of functions of every particular kind of equipment, which make it possible to support any kind of networking tasks.

Cisco Open Network Environment



http://www.cisco.com/go/one

http://www.cisco.com/go/onepk

Thank you.