

Управление услугами с помощью NETCONF/YANG и оркестратора Cisco NSO

Владислав Патенко

Содержание

➤ Обзор NETCONF

➤ Обзор YANG

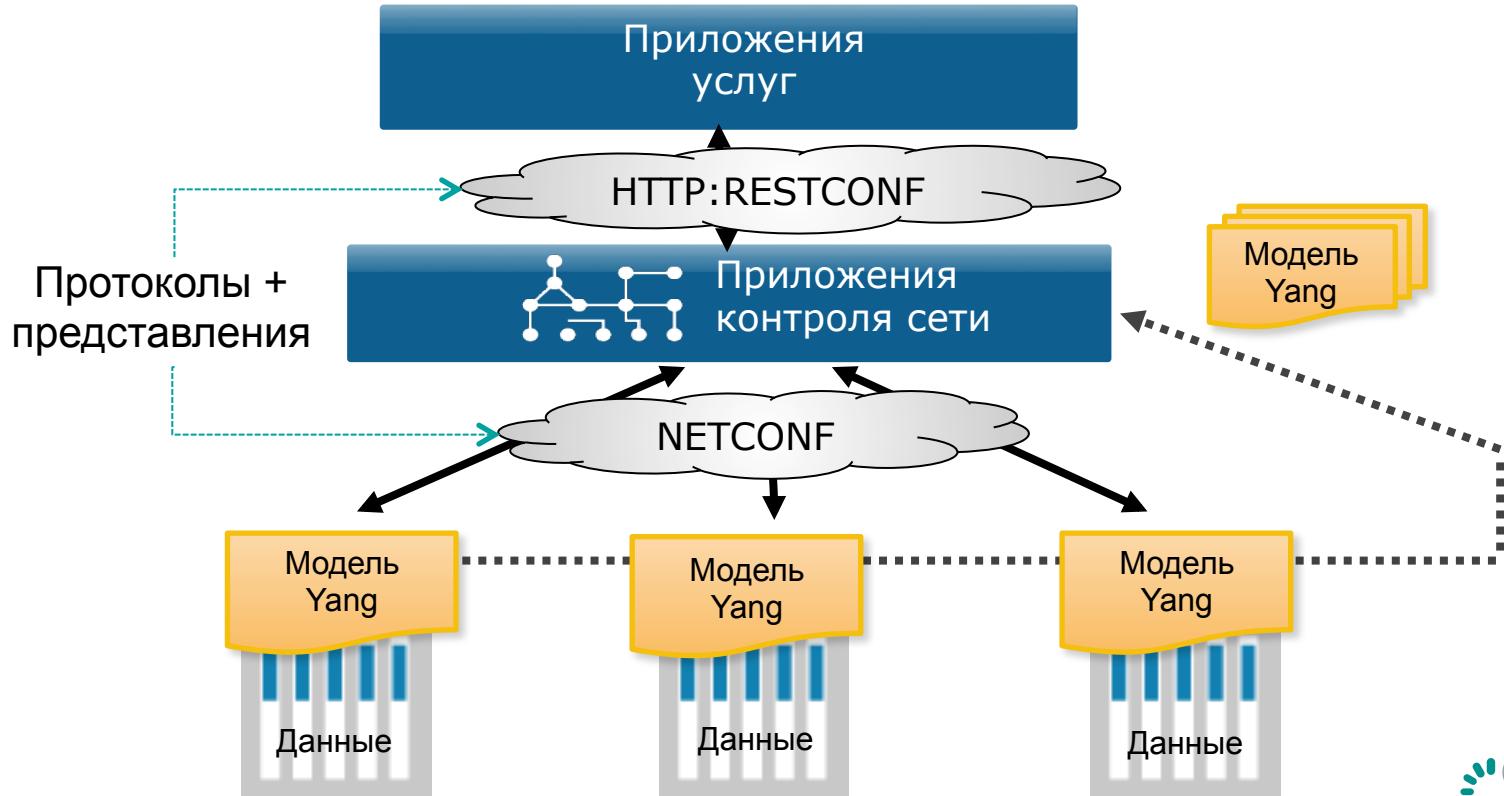
➤ Пример модели YANG

➤ Обзор Cisco NSO

➤ Дополнительная информация

NETCONF, RESTCONF и YANG

Модели данных + протоколы для разных задач



Модель данных и протокол

Протокол

Модель данных



Модель данных

- Определение структуры, синтаксиса и семантики данных
- Полнота и согласованность

Протокол

- Механизм передачи данных
- Кодирование информации, определенной моделью данных

Обзор NETCONF

Что такое NETCONF?

- Netconf – протокол, ориентированный на соединение
 - SSH, TLS как транспорт
- Клиент Netconf (“manager”) устанавливает сессию с сервером (“agent”)
- Данные кодируются в виде XML
- Базируется на RPC
 - <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
- Определен в RFC4741 (NETCONF 1.0) и RFC6241 (NETCONF 1.1)
- Функция Call-home в процессе стандартизации
 - Возможность инициировать соединение со стороны устройства

Операция NETCONF <hello>

- Обмен возможностями
- Обмен идентификаторами моделей данных
- Кодирование XML
- Фреймы
 - NETCONF 1.0 EOM,]]>]]>
 - NETCONF 1.1 Chunked Framing

```
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.1</capability>
  </capabilities>
</hello>
```

Операция NETCONF <hello> – ответ агента

```
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:candidate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:confirmed-commit:1.0</capability>
<capability>urn:ietf:params:netconf:capability>xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability>
<capability>http://tail-f.com/ns/netconf/with-defaults/1.0</capability>
<capability>http://tail-f.com/ns/netconf/actions/1.0</capability>
<capability>http://tail-f.com/ns/netconf/commit/1.0</capability>
<capability>http://tail-f.com/ns/example/dhcpd?module=dhcpd</capability>
<capability>urn:ietf:params:xml:ns:yang:ietf-inet-types?revision=2010-09-24&module=ietf-inet-
types</capability>
</capabilities>
<session-id>5</session-id>
</hello>
```

Операция NETCONF <get-config>

- Фильтрация по ветке дерева
- Фильтрация XPATH

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.1" message-id="1">
  <get-config>
    <source>
      <running/>
    </source>
    <filter xmlns="http://tail-f.com/ns/aaa/1.1">
      <aaa/>
    </filter>
  </get-config>
</rpc>
```

Операция NETCONF <get-config> - ответ

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.1" message-id="1">
  <data>
    <aaa xmlns="http://tail-f.com/ns/aaa/1.1">
      <authentication>
        <users>
          <user>
            <name>admin</name>
            <uid>9000</uid>
            <gid>0</gid>
            <password>$1$3ZHhR60w$acznSyClFc0keo3B3BVjx/</password>
            <ssh_keydir>/var/confd/homes/admin/.ssh</ssh_keydir>
            <homedir>/var/confd/homes/admin</homedir>
          </user>
          <user>
            <name>oper</name>
            ...
          </user>
        </users>
      </authentication>
    </aaa>
  </data>
</rpc-reply>
```

Операция NETCONF <edit-config>

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.1" message-id="1">
  <edit-config>
    <target><running/></target>
    <config>
      <dhcp xmlns="http://tail-f.com/ns/example/dhcpd"
            xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.1">
        <defaultLeaseTime nc:operation="merge">PT1H
        </defaultLeaseTime>
      </dhcp>
    </config>
  </edit-config>
</rpc>
```

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.1" message-id="1">
  <ok/>
</rpc-reply>
```

Операция NETCONF <edit-config> - опции

nc:test-option (:validate)
test-then-set (*default*)
set
test-only
nc:error-option
stop-on-error (*default*)
continue-on-error
rollback-on-error
(:rollback-on-error)

nc:operation
merge
replace
create
delete
remove (:base:1.1)

Ошибка, если объект для удаления не существует

Нет ошибки, если объект для удаления не существует

Операция NETCONF <copy-config>

Копирование конфигурации между местами хранения данных или через URL

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <copy-config>
    <target><running/></target>
    <source>
      <url>https://user@example.com:passphrase/cfg/new.txt
      </url>
    </source>
  </copy-config>
</rpc>
```

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <ok/>
</rpc-reply>
```

Операция NETCONF <delete-config>

Полное удаление конфигурации (не running)

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <delete-config>
    <target>
      <startup/>
    </target>
  </delete-config>
</rpc>
```

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <ok/>
</rpc-reply>
```

Операция NETCONF <lock>, <unlock>

Блокировка/разблокировка файла конфигурации

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <lock>
    <target>
      <candidate/>
    </target>
  </lock>
</rpc>
```

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <ok/>
</rpc-reply>
```

Операция NETCONF <get>

Получение конфигурации и состояния

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  <get>
    <filter type="subtree">
      <top xmlns="http://example.com/ns/dhc">
        <interfaces>
          <interface>
            <ifName>eth0</ifName>
          </interface>
        </interfaces>
      </top>
    </filter>
  </get>
```

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <top xmlns="http://example.com/ns/dhc">
      <interfaces>
        <interface>
          <ifName>eth0</ifName>
          <ifInOctets>45621</ifInOctets>
          <ifOutOctets>774344</ifOutOctets>
        </interface>
      </interfaces>
    </top>
  </data>
</rpc-reply>
```

Операция NETCONF <close-session>

Корректный путь закрыть сессию

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <close-session/>
</rpc>
```

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <ok/>
</rpc-reply>
```

Операция NETCONF <kill-session>

Не очень корректный метод закрыть другую сессию

Освобождение блокировок, отмена всех изменений, связанных с сессией

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <kill-session>
    <session-id>17</session-id>
  </kill-session>
</rpc>
```

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <ok/>
</rpc-reply>
```

Дополнительные операции NETCONF

<commit>, <discard-changes> (:candidate)

<validate> (:validate)

- Копирование candidate в running
- Отмена изменений в candidate (копирование running в candidate)

<create-subscription> (:notification)

<partial-lock>, <partial-unlock> (:partial-lock)

<commit>, <cancel-commit> (:commit)

<get-schema> (:ietf-netconf-monitoring)

Уведомления

NETCONF

- Требует открытое соединение
- Поддерживает повторные сообщения

SNMP

- Широко и успешно используется
- Небольшие пакеты
- Не требует постоянного соединения

SYSLOG

- Широко и успешно используется
- Проблемы в стандартизации формата событий

Почему NETCONF?

Набор поддерживаемых сценариев делают Netconf привлекательным

Сценарий	SNMP	NETCONF
Получение группы параметров о состоянии	Да	Да. Гораздо быстрее, чем SNMP
Изменение группы параметров	Да, до 64kB	Да
Транзакционное изменение параметров	Нет	Да
Транзакция через группу сетевых устройств	Нет	Да
Вызов административных функций	Теоретически	Да
Отсылка уведомление	Да	Да
Резервное копирование и восстановление	Обычно нет	Да
Защищенный протокол	V3	Да
Тестирование конфигурации перед применением	Нет	Да

Netconf, дополнительная информация

<http://datatracker.ietf.org/wg/netconf/>

Важные RFCs

- RFC 6241 – Network Configuration Protocol 1.1 (NETCONF)
- RFC 6242 – Using the NETCONF Protocol over Secure Shell (SSH)
- RFC 5277 – NETCONF Event Notifications
- RFC 6536 – NETCONF Access Control Model (NACM)

В процессе разработки

- Протокол RESTCONF
- Zero Touch Provisioning для NETCONF (Call Home)

Обзор YANG

Что такое YANG?

YANG – Yet Another Next-Generation...

- ... сетевой протокол?
- ... шина сообщений?
- ... язык для описания данных!

Цель

- Определение конфигурационных данных для Netconf...
- ... Изначально фокусируется на конфигурационной информации, но не ограничена ею
- ... Может использоваться отдельно от Netconf
(не цель спецификации, но важный аспект)

YANG и XML

YANG использует XML для кодирования данных

- Определяет правила генерации XML
- Использует некоторые расширенные возможности XML (например, Xpath)
- Хорошо увязывается с NETCONF

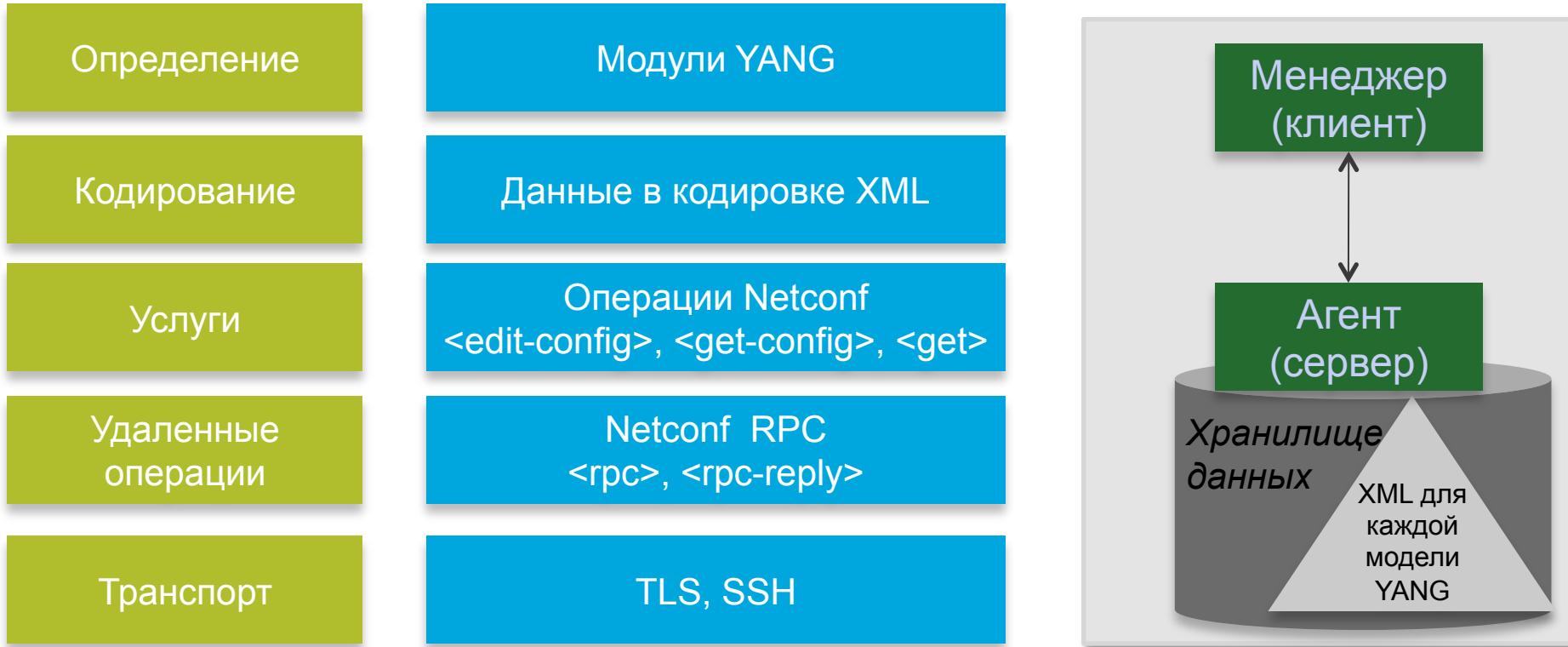
YANG не является XML

- Акцент на читаемости документа
 - Структура, удобная для программистов. Похожа на C/C++ или Java
- Грамматика XML определена в YIN (Yang-Independent Notation)
 - Похожая семантика
 - Трансляция семантики YANG <-> YIN
- Определены альтернативные методы кодирования (например, JSON для RESTconf)

Сравнение YANG и SNMP

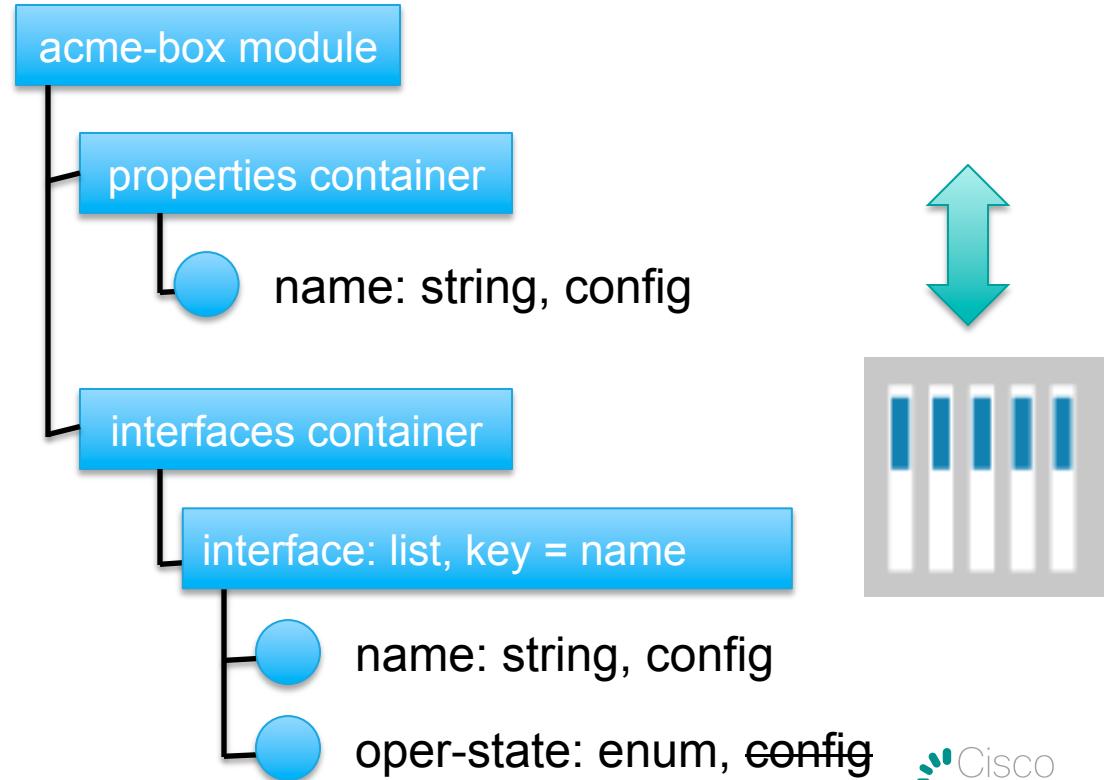


YANG в контексте Netconf



YANG

- Язык моделирования данных
 - Данные о конфигурации
 - Данные о состоянии
- Древовидная структура
- Данные и типы



Модель YANG - заголовок

Пример

```
module acme-module {
    namespace "http://acme.example.com/module";
    prefix acme;

    import "ietf-yang-types" {
        prefix yang;
    }
    include "acme-system";

    organization "ACME Inc.";
    contact joe@acme.example.com;
    description "Module describing the ACME products";
    revision 2007-06-09 {
        description "Initial revision.";
    }
}
```

Базовые типы данных YANG

- Большинство элементов YANG имеют определенный тип данных
- Тип данных может быть базовым или наследованным
 - Существует более 20 базовых типов

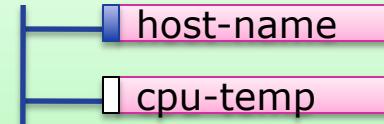
Тип данных	Значение
int8/16/32/64	Integer
uint8/16/32/64	Unsigned integer
decimal64	Non-integer
string	Unicode string
enumeration	Set of alternatives
boolean	True or false
bits	Boolean array
binary	Binary BLOB
leafref	Reference “pointer”
identityref	Unique identity
empty	No value, void
	...и больше

Оператор Leaf

Пример

Содержит одиночное значение
определенного типа

```
leaf host-name {  
    type string;  
    mandatory true;  
    config true;  
    description "Hostname for this system";  
}  
  
leaf cpu-temp {  
    type int32;  
    units degrees-celsius;  
    config false;  
    description "Current temperature in CPU";  
}
```



Представление данных:

```
<host-name>my-host</host-name>
```

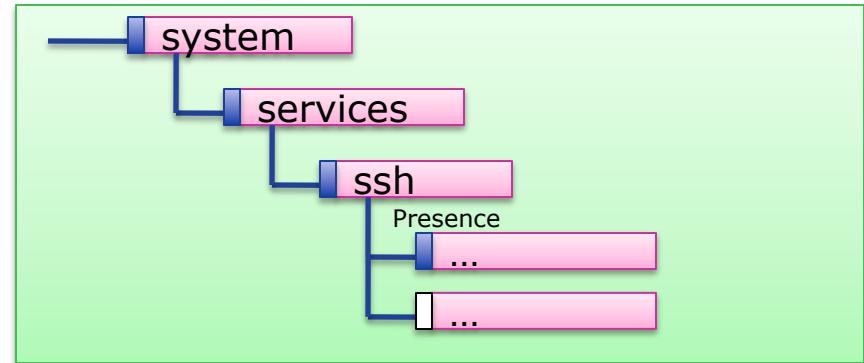
```
<cpu-temp>62</cpu-temp>
```

Оператор Container

Пример

Группирует объекты leaf и container

```
container system {  
    container services {  
        container ssh {  
            presence "Enables SSH";  
            description "SSH service specific configuration";  
            // more leafs, containers and other things here...  
        }  
    }  
}
```



Другие операторы YANG

- Import/Include
- Typedef
- Union
- Grouping
- Choice
- List
- Leaf-list
- Key
- Unique
- RPC
- Notification
- Identity
- Must
- Augments
- Extension
- Feature
- Deviation

YANG, дополнительная информация

<http://datatracker.ietf.org/wg/netmod/>

Важные RFCs

- RFC 6020: YANG – A Data Modeling Language for the Network Configuration Protocol
- RFC 6021: Common YANG Data Types
- RFC 6087: Guidelines for Authors and Reviewers of YANG Data Model Documents
- RFC 6110: Mapping YANG to Document Schema Definition Languages and Validating NETCONF Content
- RFC 6643: Translation of SMIv2 MIB Modules to YANG Modules

В процессе разработки

- Моделирование JSON с YANG
- Разные модули YANG

Пример: Моделирование услуги IPSec

Услуга IPSec Hub Пример

Конфигурация устройства:

```
crypto isakmp key MY_K3Y
address 10.194.126.2
!
crypto ipsec transform-set TS
esp-des esp-md5-hmac
!
crypto map CRYPTO 10 ipsec-
isakmp
    set peer 10.194.126.2
    set transform-set TS
    match address Spoke1
!
interface Gig0/0
    ip address 10.194.128.1
    255.255.255.0
    crypto map CRYPTO
```

Spoke1:
IP: 10.194.126.2/24



Peer: Hub1
Subnet
IP: 192.168.1.0/24

IPSec Tunnel



Приложение создания услуги
Услуга безопасного VPN:
Spoke1, Spoke2, ...

Услуга IPSec (Модель)

Узлы:

Hub1, Spoke1, Spoke2, ...

Параметры:

Адрес IP, Алгоритм шифрации, Ключи, и т.д.

Конфигурация устройства (Модель)

Производитель, модель/версия, функции, и
т.д.

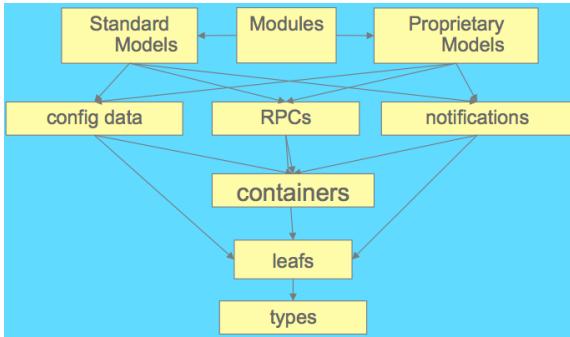
IPSec Tunnel



Spoke2:
IP: 10.194.127.1/24
Peer: Hub1
Subnet
IP: 192.168.2.0/24

Моделирование услуги IPSec*

Базовые параметры

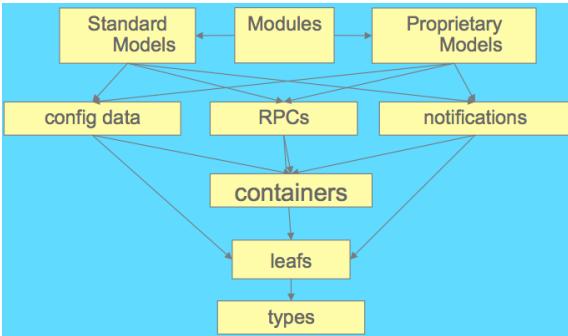


```
module ipsec-service {  
    namespace "com.example.ipsec";  
    prefix sipsec;  
  
    import ietf-inet-types {  
        prefix inet;  
    }  
}
```

*упрощенная модель для примера

Моделирование услуги IPSec

Данные (1/2)



услуга IPSec (Модель)

Узлы:

Hub1, Spoke1, Spoke2, ...

Параметры:

```
container ipsec {
    presence "Setting IPSec service";
    description "A simple IPSec service";

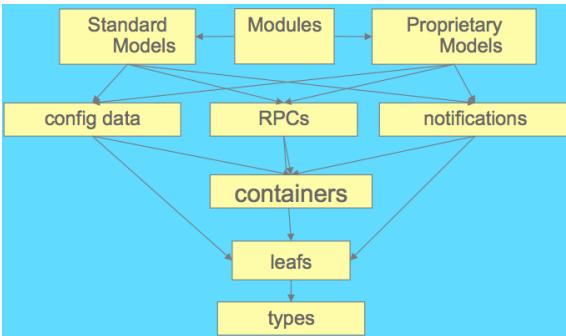
    list node-list {
        description "Name of customer hub router";
        key node-name;

        leaf node-name {type string;}

        leaf node-ip {
            description "Hub IP address";
            type inet:ipv4-address;
        }
    }
}
```

Моделирование услуги IPSec

Данные (2/2)



услуга IPSec (Модель)

Узлы:

Hub1, Spoke1, Spoke2, ...

Параметры:



Peer: Hub1
Subnet
IP: 192.168.1.0/24

```
list peer-node {  
    key peer-name;  
    leaf peer-name {type string;}  
  
list node-subnets {  
    description "IP and mask behind route";  
    key "ip inv-mask";  
    leaf ip {type inet:ipv4-address;}  
    leaf inv-mask {type inet:ipv4-address;}  
}  
  
list shared-key {  
    key shared-key;  
    leaf shared-key {type string;}  
    leaf peer-address {type inet:ip-address;}  
}  
  
list encryption-protocols{  
    key set-name;  
    leaf set-name {type string;}  
}
```

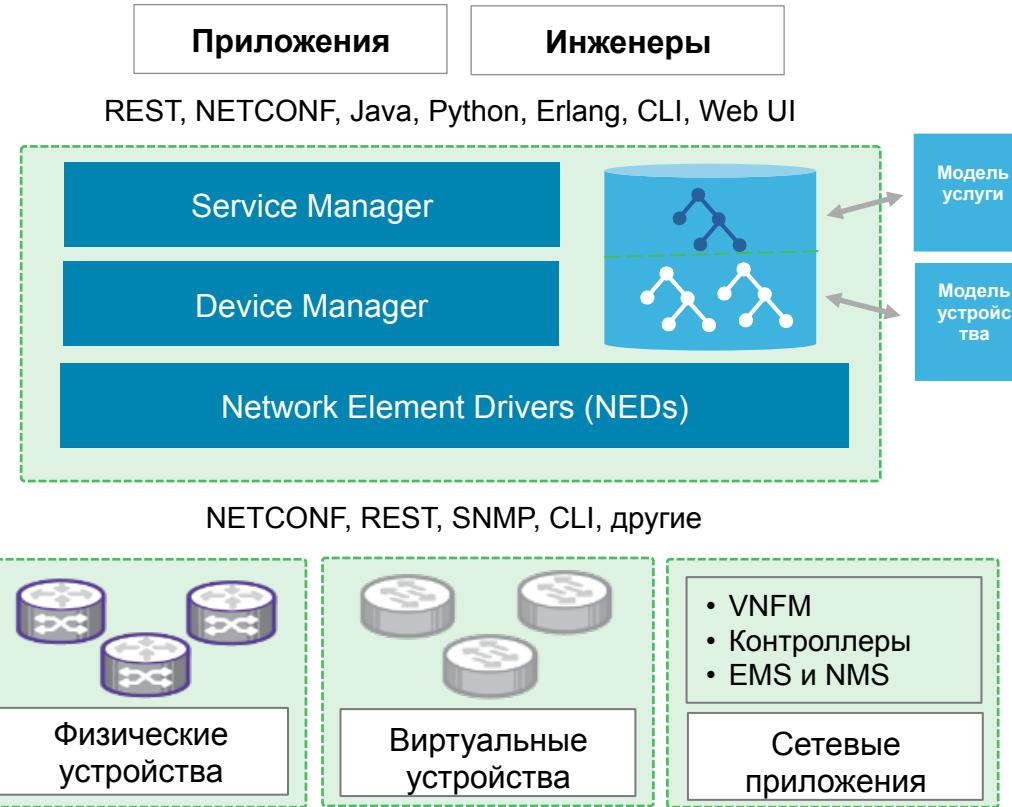
Полная модель

Пример услуги IPSec VPN

```
module ipsec-service {  
    namespace "com.example.ipsec";  
    prefix ipsec;  
  
    import ietf-inet-types {  
        prefix inet;  
    }  
  
    container ipsec {  
        presence "Setting IPSec service";  
        description "A simple IPSec service";  
  
        list node-list {  
            description "Name of customer hub router";  
            key node-name;  
  
            leaf node-name {type string;}  
  
            leaf node-ip {  
                description "Hub IP address";  
                type inet:ipv4-address;  
            }  
        }  
  
        list peer-node {  
            key peer-name;  
            leaf peer-name {type string;}  
        }  
  
        list node-subnets {  
            description "IP and mask behind route";  
            key "ip inv-mask";  
            leaf ip {type inet:ipv4-address;}  
            leaf inv-mask {type inet:ipv4-address;}  
        }  
  
        list shared-key {  
            key shared-key;  
            leaf shared-key {type string;}  
            leaf peer-address {type inet:ip-address;}  
        }  
  
        list encryption-protocols{  
            key set-name;  
            leaf set-name {type string;}  
        }  
    }  
}
```

Обзор Cisco NSO Практическое применение Netconf/Yang

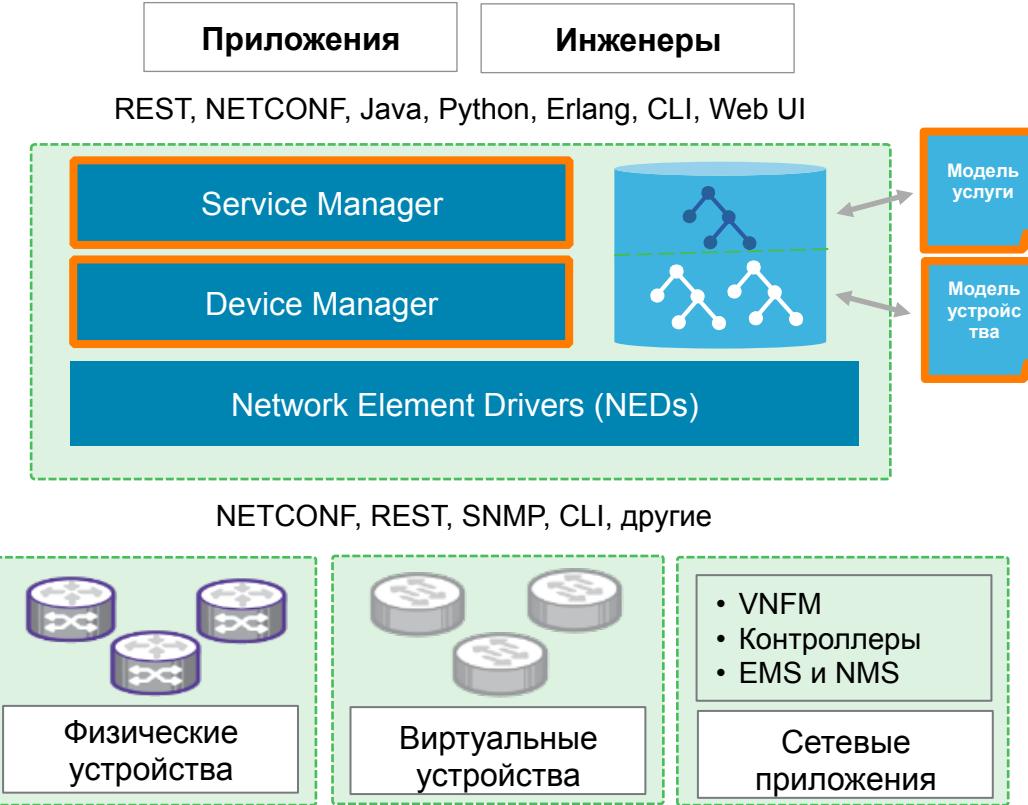
Обзор Cisco Network Service Orchestrator (NSO)



- Модели данных для услуги и устройства
- Структурированное представление:
 - Экземпляра услуги
 - Параметров и состояния сети
- Связывание процедуры изменения услуги и конфигурации устройства
- Транзакционность
- Поддержка разных производителей и протоколов

Основные функции NSO

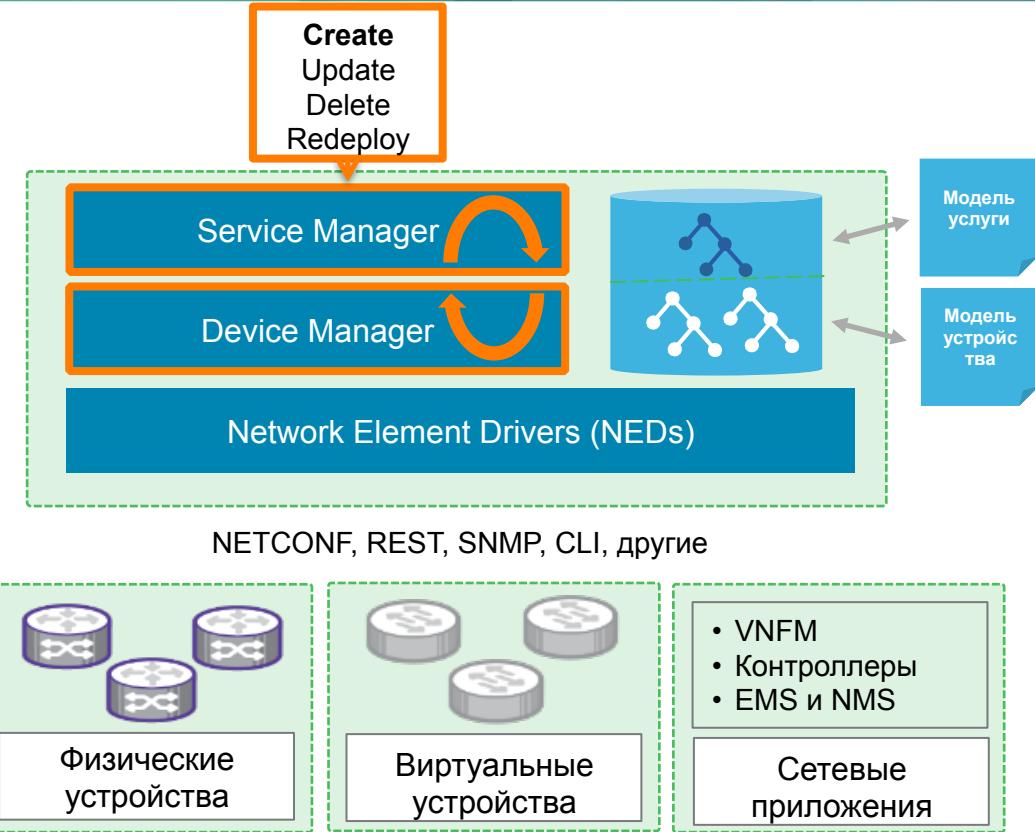
#1 Архитектура, основанная на подходе с использованием моделирования



- Отсутствие жестко запрограммированных функций для:
 - Сетевых услуг
 - Сетевой архитектуры
 - Сетевых устройств
- Вместо этого:
 - Модели данных с использованием YANG (RFC 6020)

Основные функции NSO

#2 Fastmap



- FastMap:

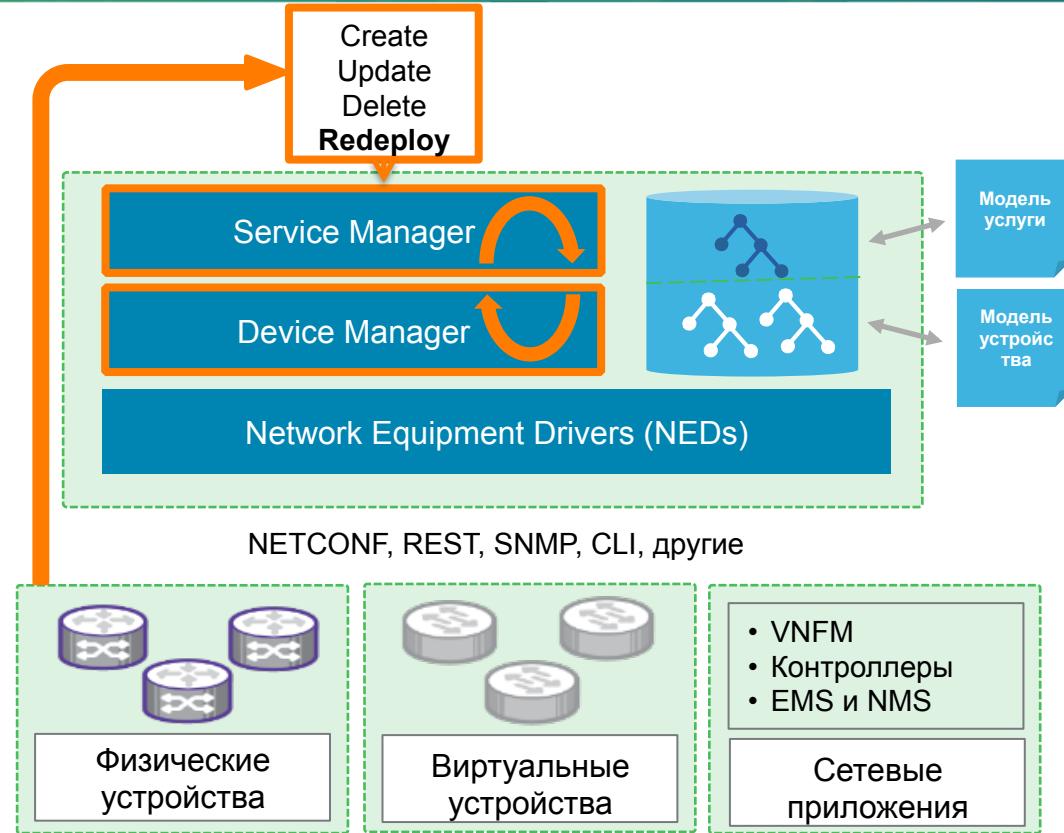
- Настраивается только операция создания услуги (Create)
- Операции изменения, удаления и пересоздания настраиваются автоматически

- Преимущества:

- Уменьшение объема работ при внедрении
- Оперативная модификация услуг на промышленной системе

Основные функции NSO

#3 Реактивный Fastmap (RFM)



- Возможности:
 - Пересоздание услуги на измененной сети
 - Idempotent
- Поддержка сценариев и технологий:
 - Провиженинг
 - Оркестрация
 - Эластичность
 - Виртуальные машины и мобильность VNF
 - Самовосстановление сети

Поддерживаемые производители в NSO

- Cisco
- Alcatel-Lucent
- Citrix Systems
- Infinera
- Ericsson
- Nominum
- Ciena
- Juniper Networks
- Huawei

- Sonus
- VMware
- Nec
- Fortinet
- Palo Alto
- Accedian Networks
- Open vSwitch
- Avaya
- ADTRAN

- Affirmed Networks
- A10 Networks
- Riverbed
- Allied Telesis
- Overture
- ADVA
- Vyatta
- Brocade

Дополнительная информация

- Документы IETF. Принятые и разрабатываемые RFC NETCONF/YANG:
<http://datatracker.ietf.org/wg/netconf/>
- Документация на Cisco NSO
<http://www.cisco.com/go/nso>



Ждем ваших сообщений с хештегом
#CiscoConnectRu

Спасибо

Пожалуйста, заполните анкеты.

Ваше мнение очень важно для нас.

Контакты:

Патенко Владислав
vpatenko@cisco.com



CiscoRu



Cisco



CiscoRussia



CiscoRu



Cisco Connect

Москва | 17–18 ноября 2015