



Základy sieťovej automatizácie

CCNAv7: modul 14

Martin Kontšek





Obsah

- **Automatizácia**
- **Dátové formáty**
- **API**
- **REST API**
- **Nástroje na správu konfigurácie**
- **IBN (intent-based networking)**
 - Cisco DNA Center
- **Praktická ukážka použitia API**

Úvod do automatizácie

- Konfigurácia sieťového zariadenia
 - CLI – konzola/SSH
 - GUI

- Konfigurácia veľkého množstva sieťových zariadení
 - Viac administrátorov?
 - Použitie automatizácie
 - Softvér
 - Úvodne nastavenie
 - Správa konfigurácie
 - Monitorovanie
 - Dizajn siete

Automatizácia okolo nás

- Samoobslužné pokladne
- Inteligentné domy/budovy
 - Regulácia teploty, otváranie okien, zavlažovanie
 - Na základe podnetov
 - Pomocou aplikácie
- Samojazdiace autá
- Autopilot v lietadlách
- Podpora rozhodnutia lekárov
- Zabezpečenie objektov
 - Rozpoznávanie ľudí z kamier
- Automatické výrobné linky



Výhody automatizácie

- Redukuje zapojenie človeka do činností
- Pôvodne vo výrobe
 - často sa opakujúce činnosti
 - Eliminácie únavy človeka
 - Eliminácia chýb
- Stroje
 - Môžu pracovať 24 hodín denne bez prestávky
 - Zvyšujú produkciu
 - Každý výrobok rovnaký
- Zbieranie a analýza veľkého množstva dát
- Roboty pracujú v podmienkach nebezpečných pre človeka
 - Požiare, ťažba v bani
 - odstraňovanie následkov katastrof
 - chemikálie, výbušniny



Smart zariadenia

- Vedia zariadenia „premýšľať“?
- Definícia slova myslieť
 - Pospájať informácie dohromady a na ich základe vykonať rozhodnutie
- Smart zariadenia
 - Využívajú technológiu na riadenie svojho správania
 - Na základe vonkajšej získanej informácie
 - Od termostatov riadiacich kúrenie na základe teploty v miestnosti
 - Chladnička objedná mlieko keď dochádza
 - Až po zložité samojazdiace autá
- Správanie smart zariadení musí byť naprogramované
 - počítače

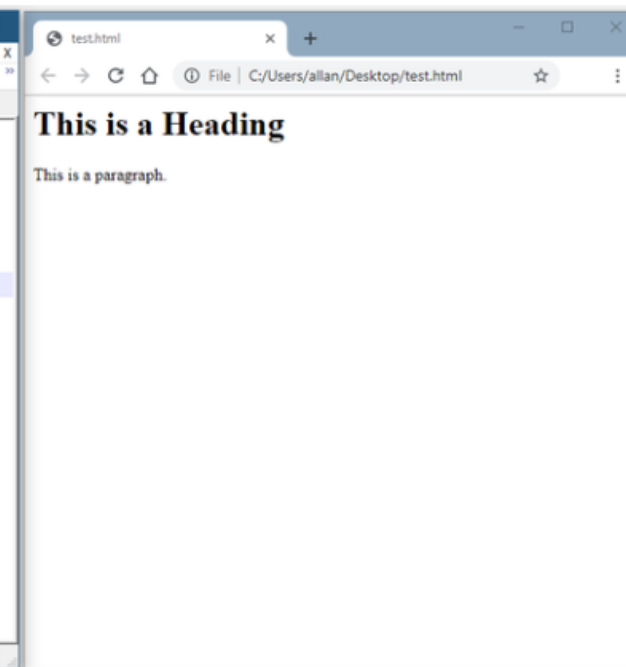


Dátové formáty

- Smart zariadenie prijíma a interpretuje údaje od iných smart zariadení
 - Senzory
 - Riadiace informácie
 - ...
- Spôsoby reprezentácie informácie
 - V podstate nekonečné
 - Príklad: formát jedálneho lístka
- Dátové formáty
 - Spoločný „jazyk“
 - Definujú spôsob ako ukladať a zdieľať údaje
 - Štrukturovaný formát
 - Výmena údajov medzi zariadeniami
 - Príklad: HTML



```
C:\Users\allan\Desktop\test.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X
test.html
1 <html>
2 <body>
3 <h1>This is a Heading</h1>
4 <p>This is a paragraph.</p>
5 </body>
6 </html>
7
length: 91 line Ln: 7 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 IN5
```



Pravidlá dátových formátov

- Pravidlá a štruktúra podobná ako programovacie jazyky
- Každý dátový formát – špecifické charakteristiky
 - Syntax
 - Použitie zátvoriek
 - Prázdne znaky (SPACE, TAB)
 - Odsadenie
 - Úvodzovky
 - Čiarky
 - Reprezentácia objektov
 - Znaky, reťazce, zoznamy, polia
 - Reprezentácia typu kľúč/hodnoty
 - Kľúč – identifikuje/popisuje dáta
 - Hodnota – samotné dáta
 - Najpoužívanejšie formáty sieťovej automatizácie
 - JavaScript Object Notation (JSON)
 - eXtensible Markup Language (XML)
 - YAML Ain't Markup Language (YAML)

```
{  
  "message": "success",  
  "timestamp": 1560789260,  
  "iss_position": {  
    "latitude": "25.9990",  
    "longitude": "-132.6992"  
  }  
}
```

JSON Format

```
message: success  
timestamp: 1560789260  
iss_position:  
  latitude: '25.9990'  
  longitude: '-132.6992'
```

YAML Format

```
<?xml version="1.0" encoding="UTF-8" ?>  
<root>  
<message>success</message>  
<timestamp>1560789260</timestamp>  
<iss_position>  
  <latitude>25.9990</latitude>  
  <longitude>-132.6992</longitude>  
</iss_position>  
</root>
```

XML Format

JavaScript Object Notation (JSON)

- Dátový formát ľahko čitateľný pre človeka
- Ukladanie, prenos a čítanie údajov
- Populárny vo webových službách a verejných API
- Jednoduché spracovanie
- Podporuje väčšina programovacích jazykov
 - Python, JavaScript, JAVA, C#...
- Príklad:
 - Info o rozhraní (show interface Gi0/0)
 - Každý objekt (kľúč/hodnota) samostatná informácia
 - Názov rozhrania
 - Popis
 - Či je zapnuté
 - IP adresa

```
{  
  "ietf-interfaces:interface": {  
    "name": "GigabitEthernet0/0",  
    "description": "Wide Area Network",  
    "enabled": true,  
    "ietf-ip:ipv4": {  
      "address": [  
        {  
          "ip": "172.16.0.2",  
          "netmask": "255.255.255.0"  
        }  
      ]  
    }  
  }  
}
```

```
GigabitEthernet0/0 is up, line protocol is up (connected)  
Description: Wide Area Network  
Internet address is 172.16.0.2/24
```

Syntaktické pravidlá JSONu

- Hierarchická štruktúra s vnorenými položkami
- Kučeravé zátvorky {} – objekty
- Hranaté zátvorky [] – polia
- Údaje zapísané ako kľúč/hodnota
- Syntax objektu
 - Kľúč – reťazec ohraničený úvodzovkami
 - Hodnoty – reťazec, číslo, pole, bool, null, objekt
 - Viaceré páry kľúč/hodnota oddelené čiarkou
 - Medzery nepodstatné
- Syntax poľa
 - Zoradený zoznam hodnôt
 - Za kľúčom nasleduje dvojbodka
 - Zoznam hodnôt uzavretý hranatými zátvorkami
 - Jedno pole môže obsahovať rôzne dátové typy
 - Hodnoty poľa oddelené čiarkou

```
{
  "ietf-interfaces:interface": {
    "name": "GigabitEthernet2",
    "description": "Wide Area Network",
    "enabled": true,
    "ietf-ip:ipv4": {
      "address": [
        {
          "ip": "172.16.0.2",
          "netmask": "255.255.255.0"
        },
        {
          "ip": "172.16.0.3",
          "netmask": "255.255.255.0"
        },
        {
          "ip": "172.16.0.4",
          "netmask": "255.255.255.0"
        }
      ]
    }
  }
}
```

YAML Ain't Markup Language (YAML)

- Dátový formát ľahko čitateľný pre človeka
- Ukladanie, prenos a čítanie údajov
- Veľmi podobný JSONu
- Považovaný za nadmnožinu JSONu
- Minimalistický formát
- Jednoduchý na zápis a čítanie
- Štruktúra definovaná odsadením
 - Bez zátvoriek a čiarok
 - Pozor na medzery a tabulátory
- Objekt z jedného alebo viacerých párov kľúč/hodnota
- Kľúč oddelený od hodnoty dvojbodkou a medzerou
- Pomlčka – prvok zoznamu

```
ietf-interfaces:interface:  
  name: GigabitEthernet2  
  description: Wide Area Network  
  enabled: true  
  ietf-ip:ipv4:  
    address:  
      - ip: 172.16.0.2  
        netmask: 255.255.255.0  
      - ip: 172.16.0.3  
        netmask: 255.255.255.0  
      - ip: 172.16.0.4  
        netmask: 255.255.255.0
```

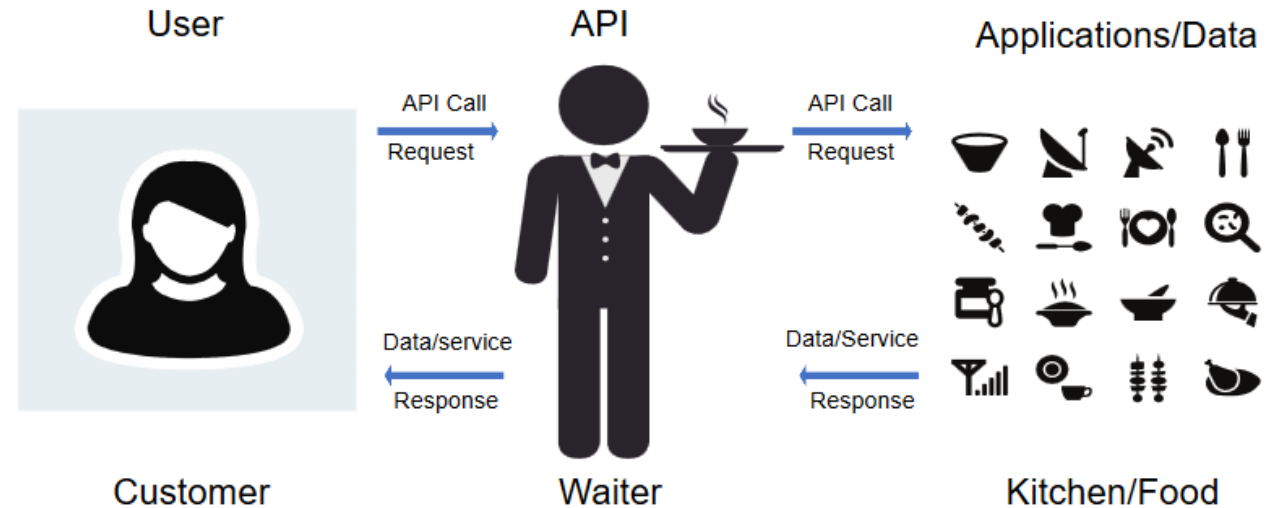
eXtensible Markup Language (XML)

- Dátový formát ľahko čitateľný pre človeka
- Ukladanie, prenos a čítanie údajov
- Podobný HTML
- Samopopisný
- Údaje uložené medzi zodpovedajúcimi tagmi
 - `<tag>data</tag>`
- XML na rozdiel od HTML
 - Nemá preddefinované tagy
 - Bez štruktúry dokumentu
- XML objekty
 - Jeden alebo viac párov párov kľúč/hodnota
 - Začiatkový a konečný tag reprezentuje kľúč
 - Hodnota medzi tagmi
 - `<kľúč>hodnota</kľúč>`
- Odsadenie nie je vyžadované (vhodné pre čitateľnosť človekom)

```
<?xml version="1.0" encoding="UTF-8" ?>
<ietf-interfaces:interface>
  <name>GigabitEthernet2</name>
  <description>Wide Area Network</description>
  <enabled>true</enabled>
  <ietf-ip:ipv4>
    <address>
      <ip>172.16.0.2</ip>
      <netmask>255.255.255.0</netmask>
    </address>
    <address>
      <ip>172.16.0.3</ip>
      <netmask>255.255.255.0</netmask>
    </address>
    <address>
      <ip>172.16.0.4</ip>
      <netmask>255.255.255.0</netmask>
    </address>
  </ietf-ip:ipv4>
</ietf-interfaces:interface>
```

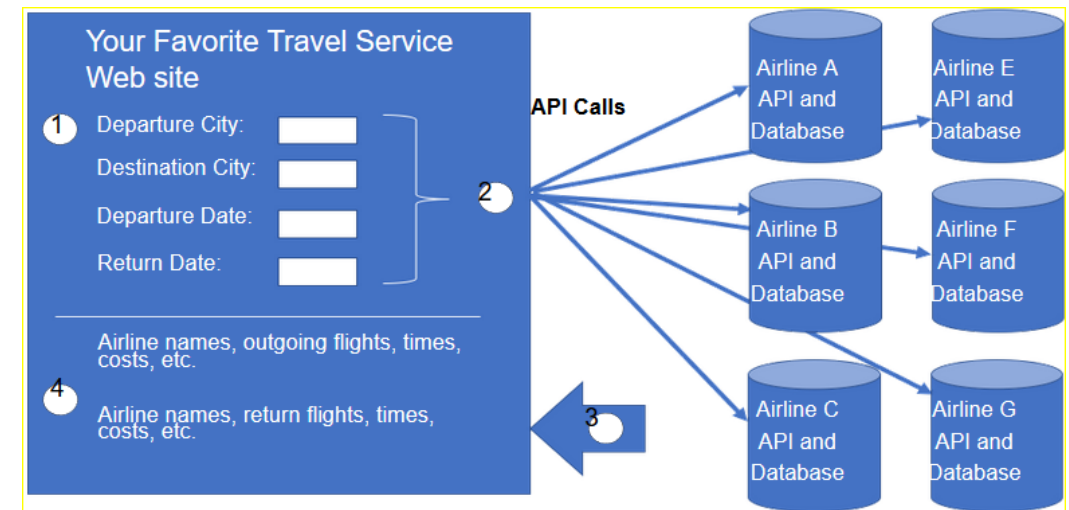
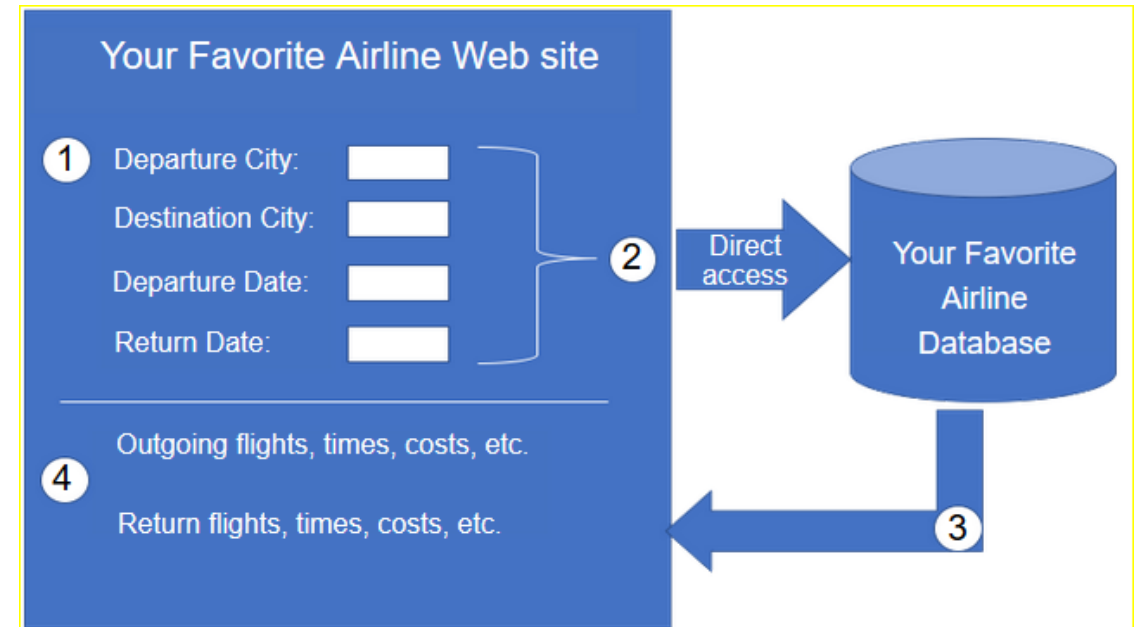

Application Programming Interface (API)

- Softvér prístupujúci aplikácia prístup k jeho údajom a službám
- Súhrn pravidiel popisujúcich interakciu medzi aplikáciami a inštrukcie ako interakcia prebehne
- Príklad čašníka
- Inteligentné domácnosti
 - Senzory
 - Termostaty
 - Chladničky
 - Osvetlenie
 - Reprodukory: Google Home, Amazon Echo
 - Televízia: Google Chromecast
- Sieťové zariadenia
 - Fortinet
 - Juniper
 - Cisco: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/166/b_166_programmability_cg/restconf_prog_int.html
 - Mikrotik: <https://help.mikrotik.com/docs/display/ROS/REST+API>
- Veľké web služby
 - Facebook: <https://developers.facebook.com/docs/apis-and-sdks/>
 - Google: <https://developers.google.com/apis-explorer>
 - Dropbox: <https://www.dropbox.com/developers/documentation/http/documentation>
 - Office 365: <https://docs.microsoft.com/en-us/previous-versions/office/office-365-api/>



Príklad bez API a s API

- Vyhládanie a zakúpenie letenky
- Bez API
 - Stránka aerolínii
 - Info o lete z vlastnej databázy
- S API
 - Stránka vyhľadávača letov od rôznych aerolínii (Pelikán)
 - Dotaz na stránke vyvolá komunikáciu cez API rozhranie ku rôznym aerolíniám
 - Stránka zobrazí agregované informácie
 - API – komunikácia medzi vyhľadávačom a databázami aerolíniek



Typy API

- **Otvorené/verejné API**
 - Dostupné verejnosti bez obmedzení
 - Často vyžadované vyžiadanie bezplatného kľúča (API key/token)
 - Limitovanie počtu žiadostí
 - Identifikácia čo ktorého používateľa najviac zaujíma
- **Interné/privátne API**
 - Použité vo firme na prístup k firemným údajom len oprávnenými osobami
- **Partnerské API**
 - Prístup k údajom medzi firmami na základe zmlúv

Open/Public APIs



APIs that are publicly available

Internal/Private APIs



APIs used within an organization

Partner APIs



APIs between a company and its business partners

Najpoužívanejšie štandardizované API

- Simple Object Access Protocol (SOAP)
 - Posielanie správ vo formáte XML
 - Pomocou HTTP alebo SMTP
 - Pomalé, komplexné
- Representational State Transfer (REST)
 - Jednoduchšie ako SOAP
 - Používa HTTP
 - Nevyžaduje XML, vie použiť XML, JSON, YAML
 - Jednoduché, výkonné, spoľahlivé
 - Najpoužívanejšie API, hlavne vo Web aplikáciách (80%)
- NETCONF
 - Náhrada SNMP
 - Používa SSH a XML
- RESTCONF
 - API podobné REST použité na správu sieťových zariadení
 - Používa XML, JSON a YANG modely na reprezentáciu údajov
- eXtensible Markup Language-Remote Procedure Call (XML-RPC)
 - Vzdialené volanie procedúr, najstarší spôsob, XML
- JavaScript Object Notation-Remote Procedure Call (JSON-RPC)

REST a RESTful API

- Webový prehliadač používa HTTP na vyžiadanie stránky (GET metóda)
 - Úspech indikovaný odpoveďou 200 OK a server zašle obsah stránky
- REST funguje nad HTTP
- Definuje súbor funkcií na žiadosti a odpovede pomocou HTTP
- API je považované za RESTful, ak dodržiava REST architektúru a tieto funkcie
 - Klient-server
 - Klient vizualizuje údaje (front end)
 - Server generuje údaje (back end)
 - Oba môžu byť nahradené bez zmeny druhého
 - Stateless(Bezstavovosť)
 - Klientské dáta medzi žiadosťami nie sú uložené na serveri
 - Stav relácie ukladá klient
 - Cacheable
 - Klient môže predpripraviť odpovede
 - Zvýšenie výkonu



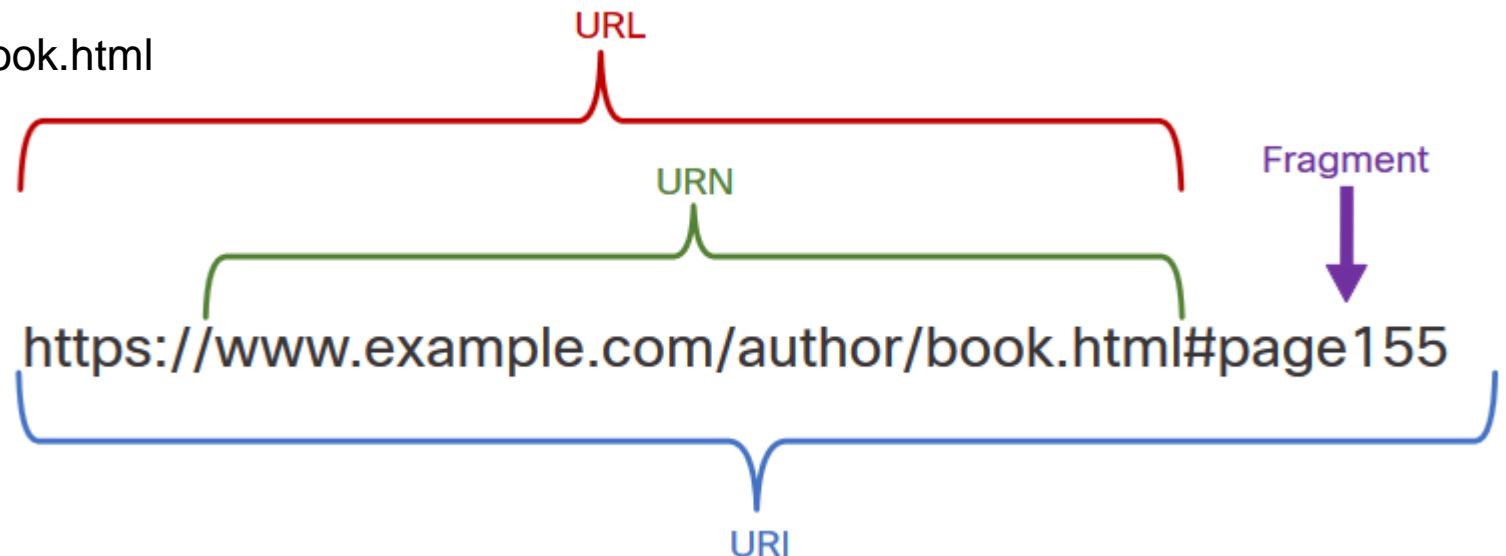
Implementácia RESTful API

- RESTful web služba používa HTTP
- Pozostáva z prostriedkov s nasledujúcimi definovanými parametrami:
 - Základná URI (Uniform Resource Identifier)
 - Napríklad <https://router.example.com/resources>
 - Dátový formát podporovaný službou
 - JSON, YAML, XML a iné definované v rámci hypertextového štandardu
 - Operácie podporované web službou pomocou volania HTTP metód
 - API musí byť spĺňať princípy hypertextu
- RESTful API používa štandardné HTTP metódy
 - Korešpondujú s RESTful operáciami

HTTP metóda	RESTful operácia
POST	Vytvoriť (Create)
GET	Načítať (Read)
PUT/PATCH	Aktualizovať (Update)
DELETE	Vymazať (Delete)

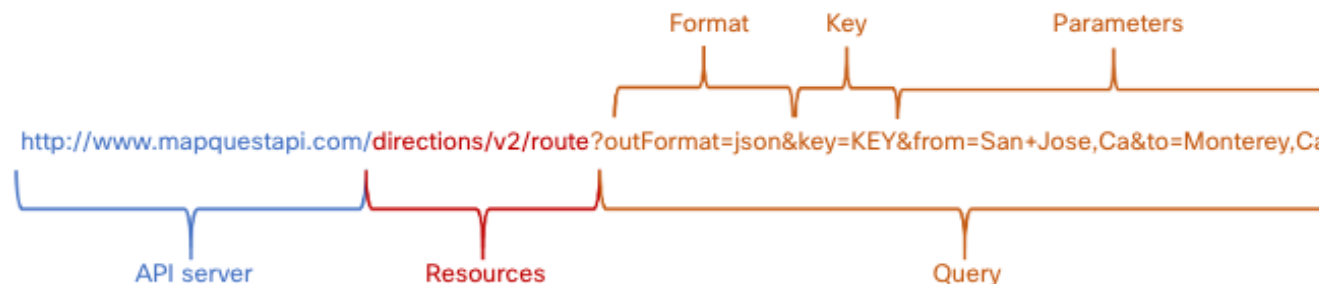
Rozdiely medzi URI, URN a URL

- Uniform Resource Identifier (URI)
 - Reťazec identifikujúci sieťový prostriedok alebo službu
- Uniform Resource Name (URN)
 - Časť URI, identifikuje len menný priestor prostriedku, bez info o protokole
- Uniform Resource Locator (URL)
 - Časť URI, identifikuje menný priestor prostriedku, obsahuje aj info o protokole
- Samotná URI pozostáva z
 - **Protokol** – https://
 - **Hostname** – www.example.com
 - **Cesta a názov súboru** – /author/book.html
 - **Fragment** - #page155



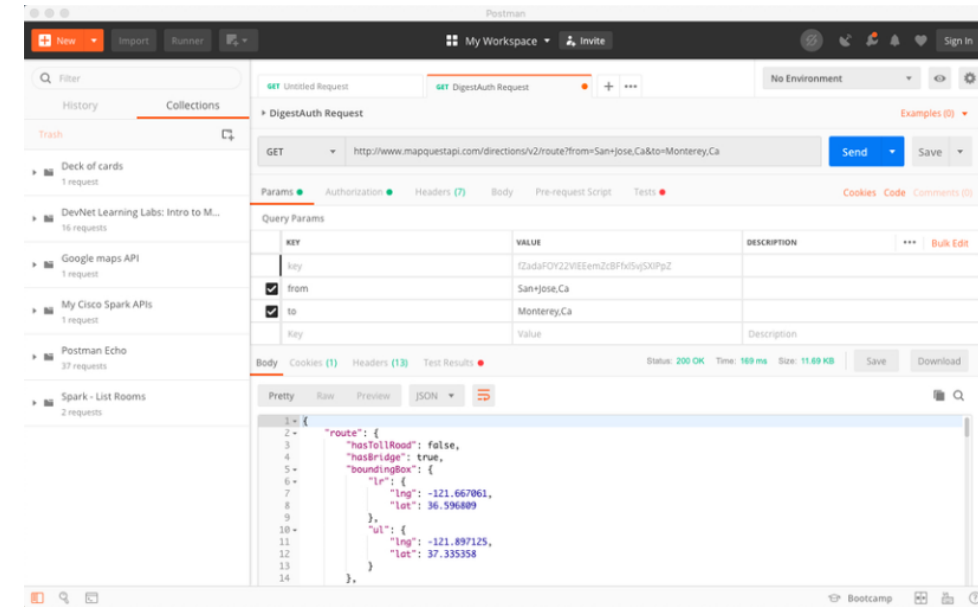
Časti RESTful žiadosti

- Žiadosť na URI RESTful web služby vyvolá odpoveď
 - Väčšinou JSON odpoveď, môže byť aj XML, YAML
- Časti žiadosti:
 - API Server - URL servera poskytujúceho službu
 - Resource - Špecifikuje volané API
 - Format - V akom formáte má byť odpoveď
 - Key
 - identifikácia používateľa vytvárajúceho žiadosť
 - Väčšinou je potrebná registrácia
 - Autentifikácia, limit množstva používateľov, limit počtu žiadostí, zistenie info o používateľoch API
 - Parameters – doplňujúce informácie (parametre) žiadosti



Ako používať RESTful API

- Pomocou webového prehliadača
 - Vykonanie GET žiadosti priamo z adresného riadku prehliadača
- Pomocou stránky API pre vývojárov
 - Mnohé API poskytujú stránky s info ako používať ich API
 - Často aj možnosť vyskúšať jednotlivé volania zadaním parametrov
 - <https://developer.mapquest.com/documentation/directions-api/>
- Postman
 - Aplikácia na testovanie REST API
 - <https://www.postman.com/downloads/>
- CURL
 - CLI aplikácia
- Programovací jazyk
 - API je možné volať priamo z programovacieho jazyka
- Sieťový operačný systém
 - Správa OS na smerovači pomocou rôznych protokolov
 - NETCONF, RESTCONF



```
$ ssh admin@192.168.0.1 -p 830 -s netconf
admin@192.168.0.1's password:
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
  <capability>urn:ietf:params:netconf:base:1.1</capability>
  <capability>urn:ietf:params:netconf:capability:candidate:1.0</capability>
  <capability>urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring</capability>
  <capability>urn:ietf:params:xml:ns:yang:ietf-interfaces</capability>
  [output omitted and edited for clarity]
</capabilities>
<session-id>19150</session-id></hello>
```

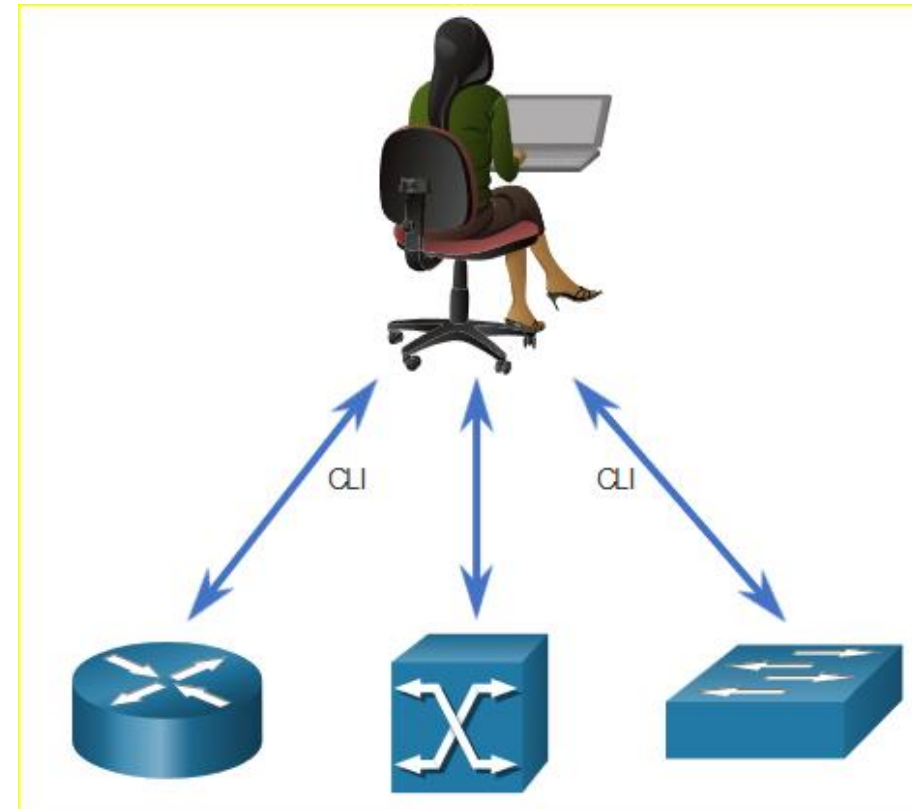
Príklady jednoduchých verejných API

- <https://ip-api.com/>
- <http://open-notify.org/>
- <https://api.nasa.gov/>
- <https://api.chucknorris.io/>
- <https://geocode.xyz/api>
- https://github.com/15Dkatz/official_joke_api
- <https://developer.mapquest.com/documentation/>

- Zoznam rôznych verejných API
 - <https://apis.io/>
 - <https://rapidapi.com/>

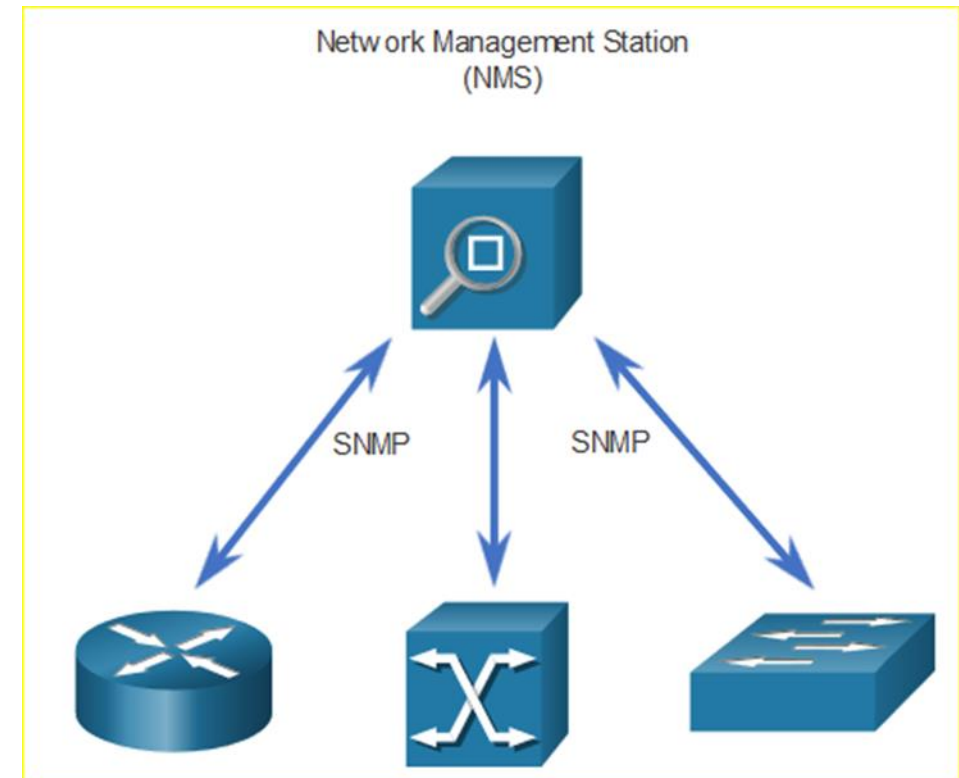
Tradičný spôsob konfigurácie sieťových zariadení

- Administrátorom pomocou CLI
 - Zmeny a nové funkcie
 - Manuálna konfigurácia na každom zariadení
 - Problém vo veľkých sieťach
 - Komplexná konfigurácia



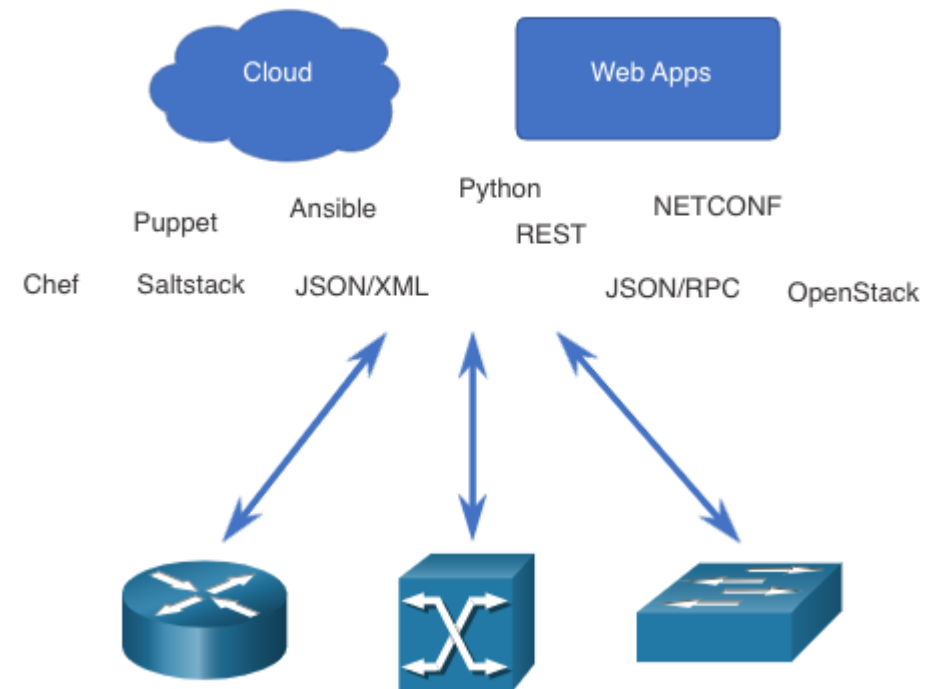
Tradičný spôsob konfigurácie sieťových zariadení

- Snaha riešenia manuálnej konfigurácia pomocou SNMP
- SNMP
 - Protokol na správu sieťových zariadení
 - NMS – centrálny uzol, z ktorého sa zariadenia spravujú
 - Monitoring stavu a výkonu
 - Zmena konfigurácia
- SNMP sa na zmenu konfigurácie v praxi nepoužíva
 - Problém s bezpečnosťou
 - Zložitosť implementácie



Automatizácia sieťových zariadení

- Administrátor spravuje veľké množstvo zariadení
- Zložité topológie (logické aj fyzické)
- Za pomoci softvéru a protokolov
 - REST
 - NETCONF
 - Ansible
 - Python
 - OpenStack
 - Cisco DNA Center



Nástroje na správu konfigurácie

- Používajú RESTful API na automatizáciu úloh
 - Správa veľkého množstva zariadení
 - Riadenie softvéru a aktualizácii
 - Konfiguračné parametre zariadení – hostname, zabezpečenie, adresácia
 - Konfigurácia protokolov
 - Konfigurácia ACL
- Nástroje na správu konfigurácie zvyčajne vykonávajú
 - Automatizáciu
 - Automatické vykonanie úlohy na zariadení
 - Orchestráciu
 - Usporiadanie automatických úloh do celkov, ktoré formujú proces alebo workflow
- Ciele nástrojov na správu konfigurácie
 - Zníženie komplexnosti a času konfigurácie
 - Použitie na veľkom počte zariadení (aj malom)
 - Vytvorenie znovupoužiteľných procesov na konfiguráciu a monitorovanie

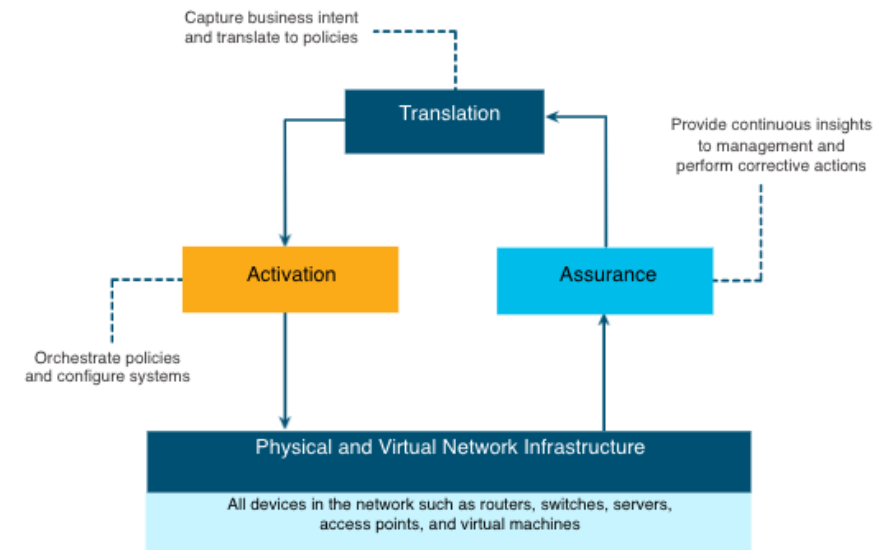
Porovnanie nástrojov na správu konfigurácie



	Ansible	Puppet	Chef	SaltStack
Použitý programovací jazyk	Python + YAML	Ruby	Ruby	Python
S agentom/bez agenta	Bez agenta	oba	S agentom	oba
Kto spravuje zariadenia	Hoci ktoré zariadenie môže byť správca	Puppet Master	Chef Master	Salt Master
Čo nástroj vytvára	Playbook	Manifest	Cookbook	Pillar

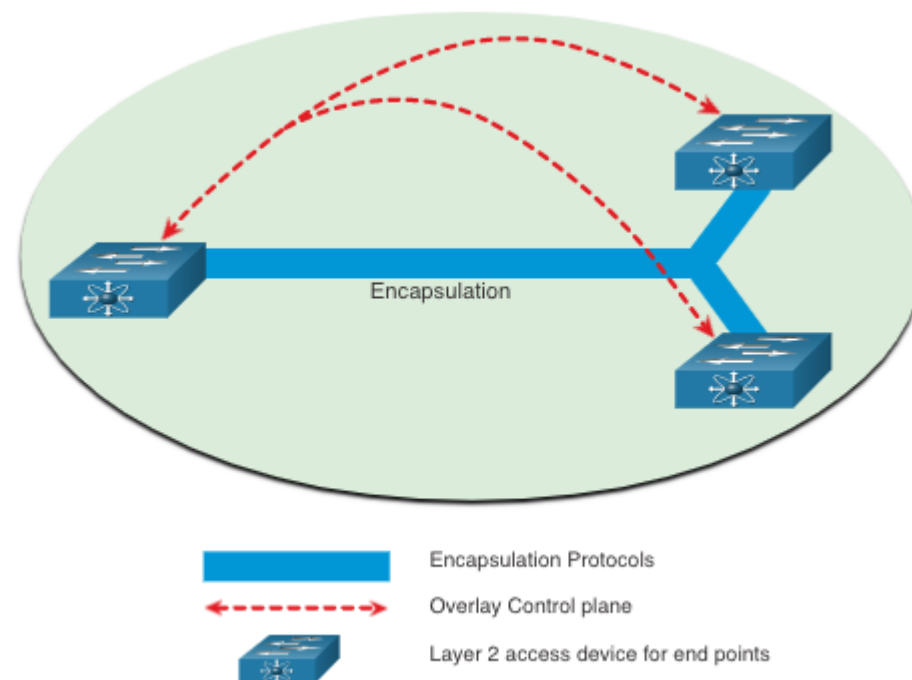
Intent-Based Networking (IBN)

- IBN je novovznikajúci model sietí novej generácia
- Stavia na princípe Software-Defined Networking (SDN)
- Prevádza hardvérový a manuálny prístup dizajnu a správy na softvérový a automatizovaný
- Podľa Cisca pozostáva z 3 základných fáz:
 - Translation (preklad)
 - Zachytí používateľskú požiadavku (business intent) na zmeny v sieti
 - Preloží ju na politiky, ktoré je možné aplikovať na zariadenia
 - Strojové učenie, automatizácia
 - Activation (aplikácia)
 - Aplikuje vytvorené politiky na sieťové zariadenia
 - automatizácia
 - Assurance (kontrola aplikácie požiadaviek)
 - Neustále kontroluje, monitoruje korektnosť nastavení
 - Podľa používateľskej požiadavky



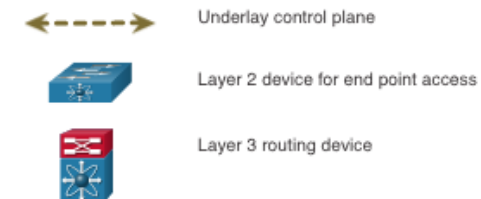
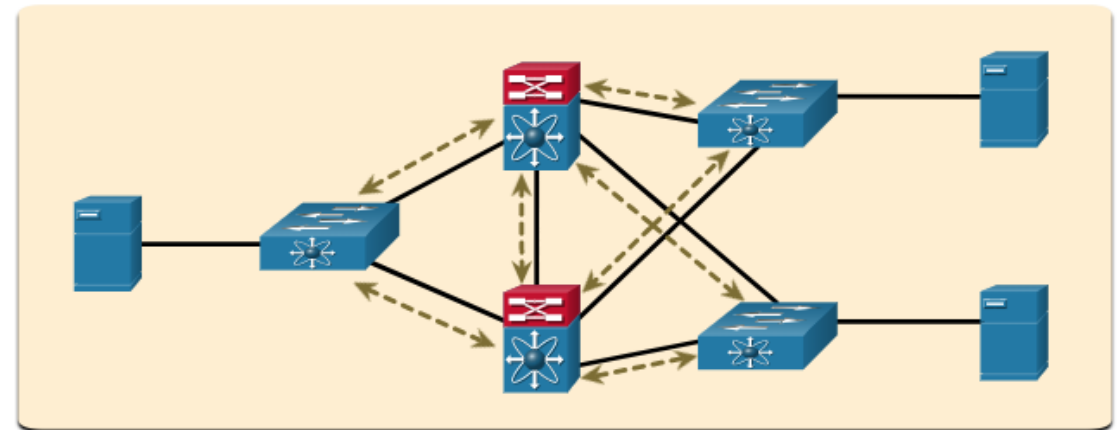
Sieťová infraštruktúra ako štruktúrovaný systém (Fabric)

- IBN vníma fyzickú a virtuálnu infraštruktúru ako štruktúrovaný systém (Fabric)
 - Overlay
 - logická topológia na virtuálne prepojenie zariadení
 - Limituje počet zariadení na konfiguráciu
 - Poskytuje služby nad rámec fyzických zariadení
 - Používa sa enkapsulačný protokol
 - IPSec, CAPWAP
 - IBN špecifikuje politiky
 - Čo sa udeje na riadiacej rovine



Sieťová infraštruktúra ako štruktúrovaný systém (Fabric)

- Underlay
 - Fyzická topológia
 - Všetok hardvér potrebný na zabezpečenie služieb
 - Špecifikuje prepojenie fyzických zariadení
 - Servery sú pripojené ku prepínačom
 - Riadiaca rovina zodpovedná za jednoduché úlohy preposielania paketov



Cisco Digital Network Architecture (DNA)

- Cisco DNA
 - Implementácia IBN Fabric
 - Pomocou Cisco DNA Center je zadaná používateľská požiadavka
 - Následne preložená na politiky
 - Automaticky nakonfigurovaná na sieťové zariadenia
 - Cisco DNA neustále monitoruje zariadenia
 - Správnosť konfigurácie na základe požiadavky
 - Monitoring funkčnosti
 - Zber diagnostických dát
 - V prípade problémov sprístupní informácie
 - Navrhuje spôsob riešenia problému

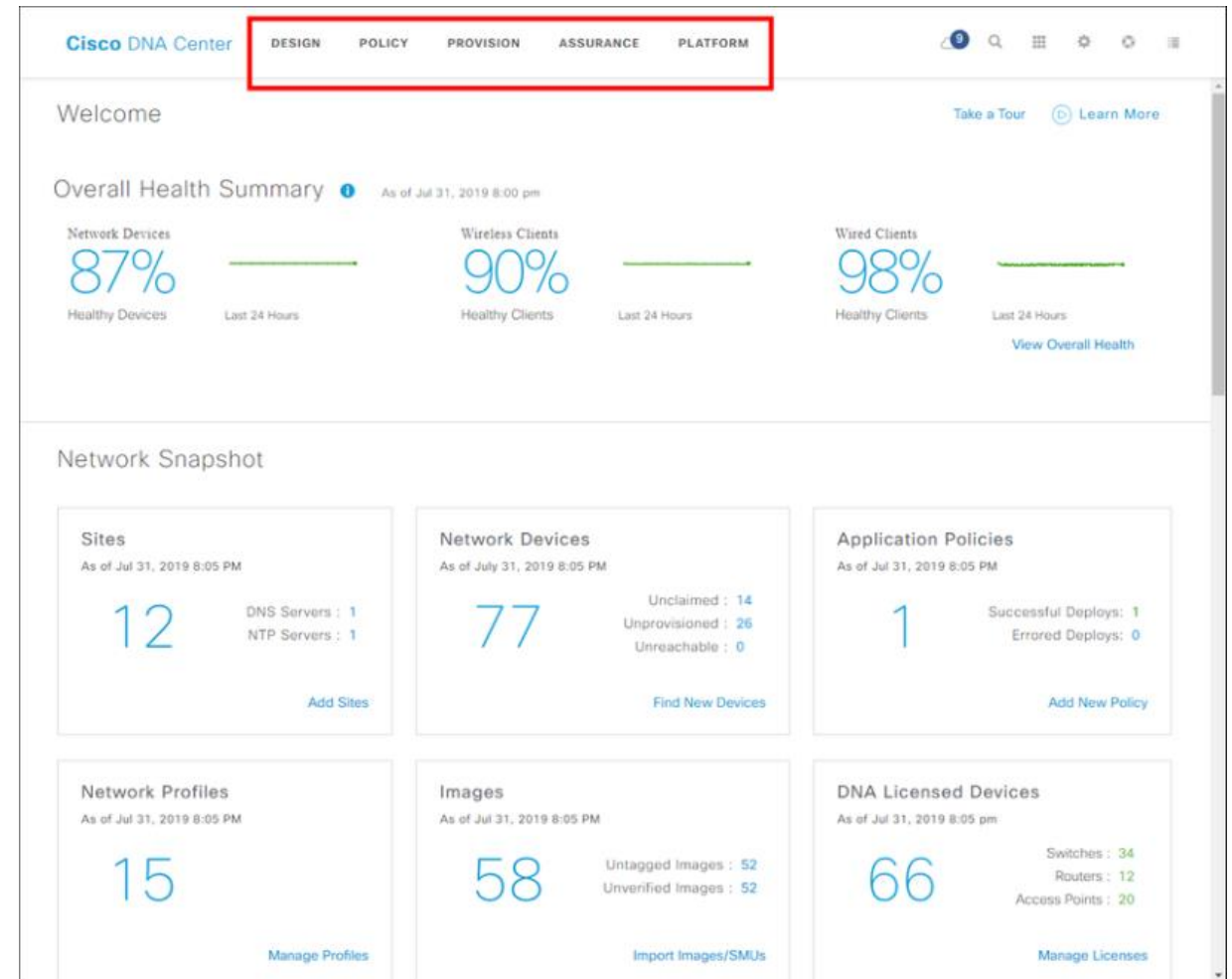


Prvky Cisco DNA

- SD-Access
 - Prvé IBN Cisco riešenie
 - Vytvára jednotnú zabezpečenú štruktúru nad LAN a WLAN
 - Oddeluje používateľov, zariadenia a aplikačnú prevádzku
 - Automatizuje politiky na prístup používateľov ku zdrojom
 - Zabezpečuje jednoduchý prístup pre používateľov z rôznych zariadení bez zníženia bezpečnosti
- SD-WAN
 - Zabezpečená centrálna správa WAN spojení z cloudu
 - Jednoduché sprevádzkovanie zabezpečeného spojenia pobočiek, zamestnancov...
- Cisco DNA Assurance
 - Zjednodušenie riešenia problémov, navrhuje riešenia
 - Prehľadný monitoring, doplňujúce informácie až na konkrétneho používateľa
 - Zvýšenie produktivity IT zamestnancov
- Cisco DNA Security
 - Poskytuje viditeľnosť v sieti, zjednodušuje riešenie zabezpečenia
 - Granulárna kontrola nad aplikovaním bezpečnostných politík

Cisco DNA Center

- Srdce Cisco DNA riešenia (kontrolér)
- Pomocou GUI rozhrania umožňuje formulovať požiadavky
- Základná automatizácia, úvodné nastavenie zariadení
- Monitorovanie siete, pomoc pri riešení problémov
- Hlavné menu:
 - Design
 - Vymodelovanie celej siete, od lokalít a budov až po zariadenia a prepoje medzi nimi
 - Policy
 - Použitie politik na automatizáciu a zjednodušenie správy siete
 - Provision
 - Nastavenie nových služieb pre používateľov v celej sieti
 - Assurance
 - Monitoring a dohľad na sieťou, nástroje pre lepšiu viditeľnosť v sieti
 - Platform
 - API na integráciu s firemnými aplikáciami





UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

Ďakujem za pozornosť



Vytvorené v rámci projektu KEGA 026TUKE-4/2021