



Projekt JetConf

REST API pro vzdálenou správu

Ladislav Lhotka • lhotka@nic.cz • 24. listopadu 2017

Osnova

- motivace, historie
- standardy: RESTCONF a YANG
- JetConf: implementace RESTCONF serveru
- backendy: Knot DNS a Turrís
- klienti: skripty, univerzální a specifický



Rozhraní pro konfiguraci a správu

Síťová zařízení a software mají základní rozhraní pro konfiguraci a monitoring přizpůsobena individuálnímu uživateli (člověku).

- konfigurační soubory
- řádkové rozhraní (CLI)
- grafické/webové rozhraní (GUI)

Jsou často významným faktorem diferenciací výrobců/produktů.



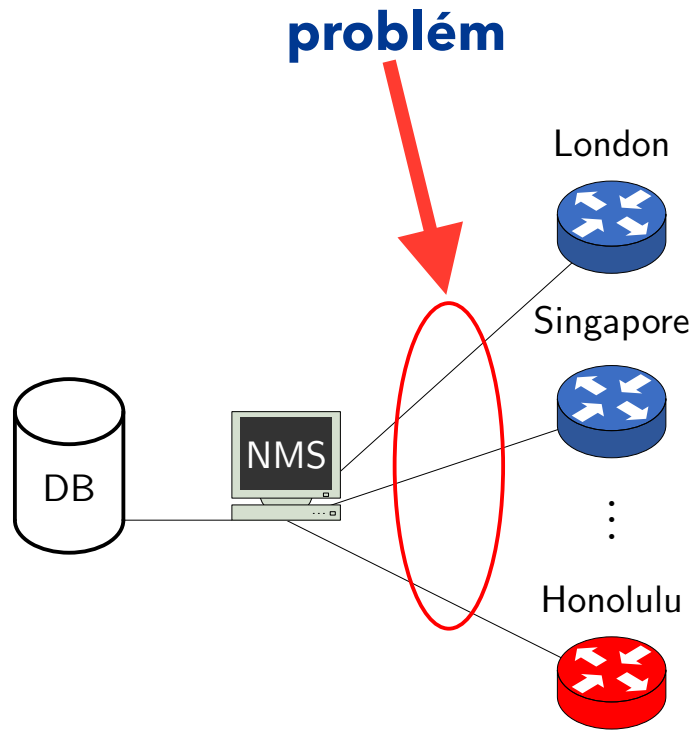
Pohled operátorů

Databáze:

- topologie sítě
- směrovací a bezpečnostní pravidla
- SLS

Požadavky:

- automatizace
- programovatelnost (API)
- standardizace
- bezpečnost, přístupová práva



Přístup #1: simulace uživatele CLI

- Expect skripty
- Puppet
- Ansible

Pracné a křehké řešení (screenscraping).



Přístup #2: protokoly pro management

- CMIP, CMOT (1987)
- SNMP (1988)
- NETCONF (2005)
- SDN (2011)
- RESTCONF (2016)



Protokol RESTCONF

Definován v RFC 8040 (leden 2017).

Vlastnosti:

- využívá principy REST a metody HTTP
- povinné TLS
- konfigurační a stavová data, RPC operace, asynchronní notifikace
- více možných reprezentací dat (JSON a XML)



YANG - jazyk pro modelování dat

Verze 1.1 definována v RFC 7950, JSON reprezentace dat v RFC 7951.

- inspirovaný schémovými jazyky XML (W3C XML Schema, Schematron)
- popis hierarchie dat, datových typů a sémantických pravidel
- vstupní a výstupní parametry RPC operací
- obsah asynchronních notifikací
- možnost formální validace dat proti datovému modelu
- podrobná dokumentace dat



JetConf - implementace serveru RESTCONF

Repo: <https://github.com/CZ-NIC/jetconf>

- napsán v Pythonu 3, používá HTTP/2 & TLS
- autentizace pomocí klientských certifikátů
- JSON reprezentace dat
- všechny povinné funkce podle RFC 8040
- konfigurace přístupových práv - NACM [RFC 6536]



Rozšíření proti RFC 8040

- Transakce, *staging* úložiště (per user)
- API pro programování backendů
- RPC operace potřebné pro klienty





Backend pro Knot DNS

Repo: <https://github.com/CZ-NIC/knot-jetconf>

Dvě části datového modelu:

- konfigurace serveru - strom konfiguračních dat
- konfigurace zón - editace zdrojových záznamů (RR) pouze RPC operacemi

Standardní datové modely zatím bohužel nejsou.

Backend k interakci s démonem využívá Unix socket.





Backend pro Turris

Repo: <https://gitlab.labs.nic.cz/jetconf/jetconf-turris>

Zatím dílčí datový model, v maximální možné míře využívající standardní modely IETF a IEEE:

- konfigurace switche
- konfigurace rozhraní a VLAN
- konfigurace IPv4/v6
- rainbow - konfigurace LED

Backend používá nízkoúrovňové funkce (Netlink socket), obchází UCI.



Klientské skripty

Vhodné pro automatizované provádění hromadných konfiguračních akcí, sběr dat apod.

Lze využít např. k implementaci modulů pro Ansible.

Nástroje: *curl*, klientské HTTP knihovny.



Univerzální klient

Funguje s každým serverem/datovým modelem bez nutnosti úprav.

Možné funkce: navigace v datovém stromu, čtení a editace dat, RPC operace, ...

Sestavení datového modelu:

- přímo z YANG modulů inzerovaných serverem
- pomocí operace `get-schema-digest` (pouze JetConf)

Nevýhoda: one size fits all, UI není přizpůsobeno konkrétnímu použití.



Specializovaný klient

Zobrazovaná data a workflow aplikace jsou přizpůsobeny konkrétnímu zařízení/sloužbě a cílové skupině uživatelů.

Využívá REST API, za běhu nutně nepotřebuje datový model.

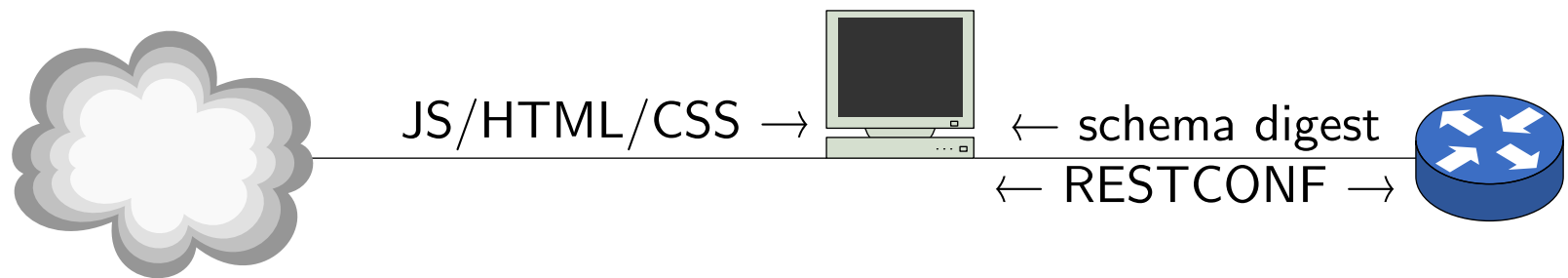
Nevýhoda: naprogramovaný ad hoc pro jeden datový model



JetScreen: univerzální webový klient

Repo: <https://gitlab.labs.nic.cz/jetconf/jetscreen>

Client-side webová aplikace



Zatím funguje jen s JetConfem (operace `get-schema-digest`).

Použitý framework: *Angular 2*, <https://angular.io/>.



https://[redacted]:8443

Commit

Reset

▼ /

▷ cznic-ethernet-switch:ports

config non-config

Send

Discard

▷ cznic-ethernet-switch:switch

▷ cznic-ethernet-switch:vllans

▼ cznic-rainbow-common:turris-leds

▷ **leds**

▷ ietf-interfaces:interfaces

▷ ietf-interfaces:interfaces-state

▷ ietf-yang-library:modules-state

Name	Type	Value
<i>name</i>	identityref	cznic-rainbow-common:pwr
color	led-color	<input type="text" value="FF3333"/>
status	enumeration	<input type="text" value="auto ▼"/>



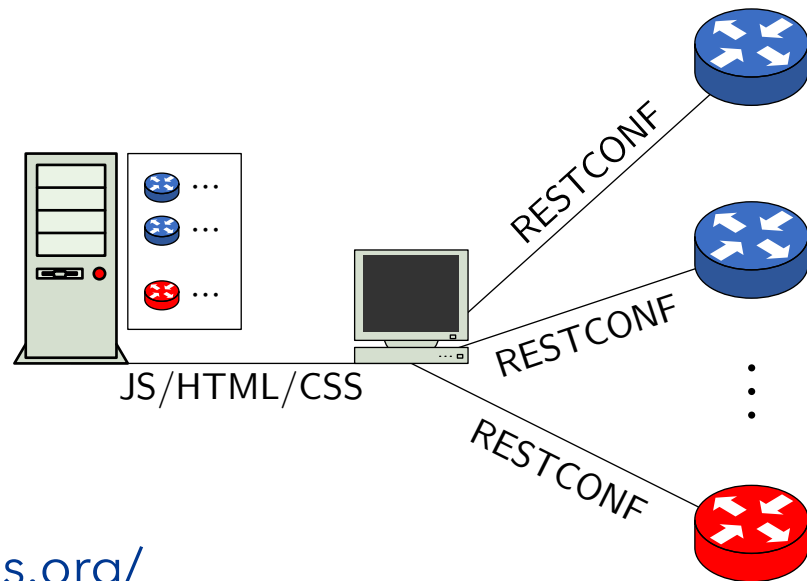
Webový klient pro sítě s Turrisy

Repo: <https://gitlab.labs.nic.cz/jetconf/jetconf-turris-ui>

Také client-side aplikace, struktura a formuláře přizpůsobeny pro Turris.

Úvodní spojení na centrální server, z něho (po autentizaci) kód aplikace a seznam dostupných routerů. Další autentizace zvlášť na každém routeru.

Použitý framework: React JS, <https://reactjs.org/>



Router list

Manage routers

Import config

Export config

Docs

Turris – záložní konektivita

Wifi jen 5GHz

██████████:8443
Plzeň, horní patro

Turris 1

██████████:8443
České Budějovice

Turris 2

██████████:8443
České Budějovice



Turris 1 (██████████:8443)

⊘ RESET

☁ COMMIT

Interfaces

Switch

VLANs

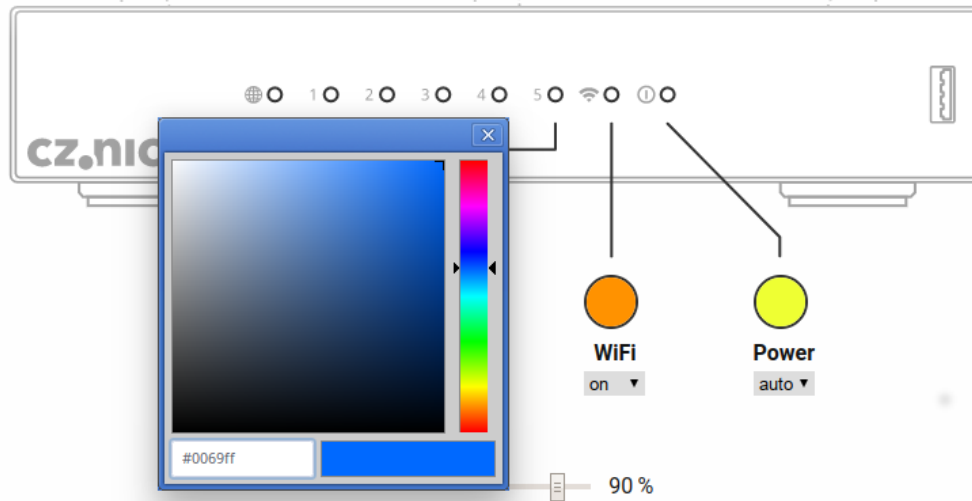
Interface state

LEDs

Ports

? Docs

← Back to router list





Děkuji za pozornost.

Ladislav Lhotka • lhotka@nic.cz • <https://labs.nic.cz>