Leaving the CLI Stone Age: Automating with APIs

Nick Russo
NETWORK ENGINEER

@nickrusso42518 www.njrusmc.net

Agenda

Why do we need APIs?

For each API in [NETCONF, RESTCONF]

- High level operations
- Contextual demos

Application Programming Interface

Set of operations built for standardized management of network devices. These operations are easily consumed by programmers and utilize structured data.

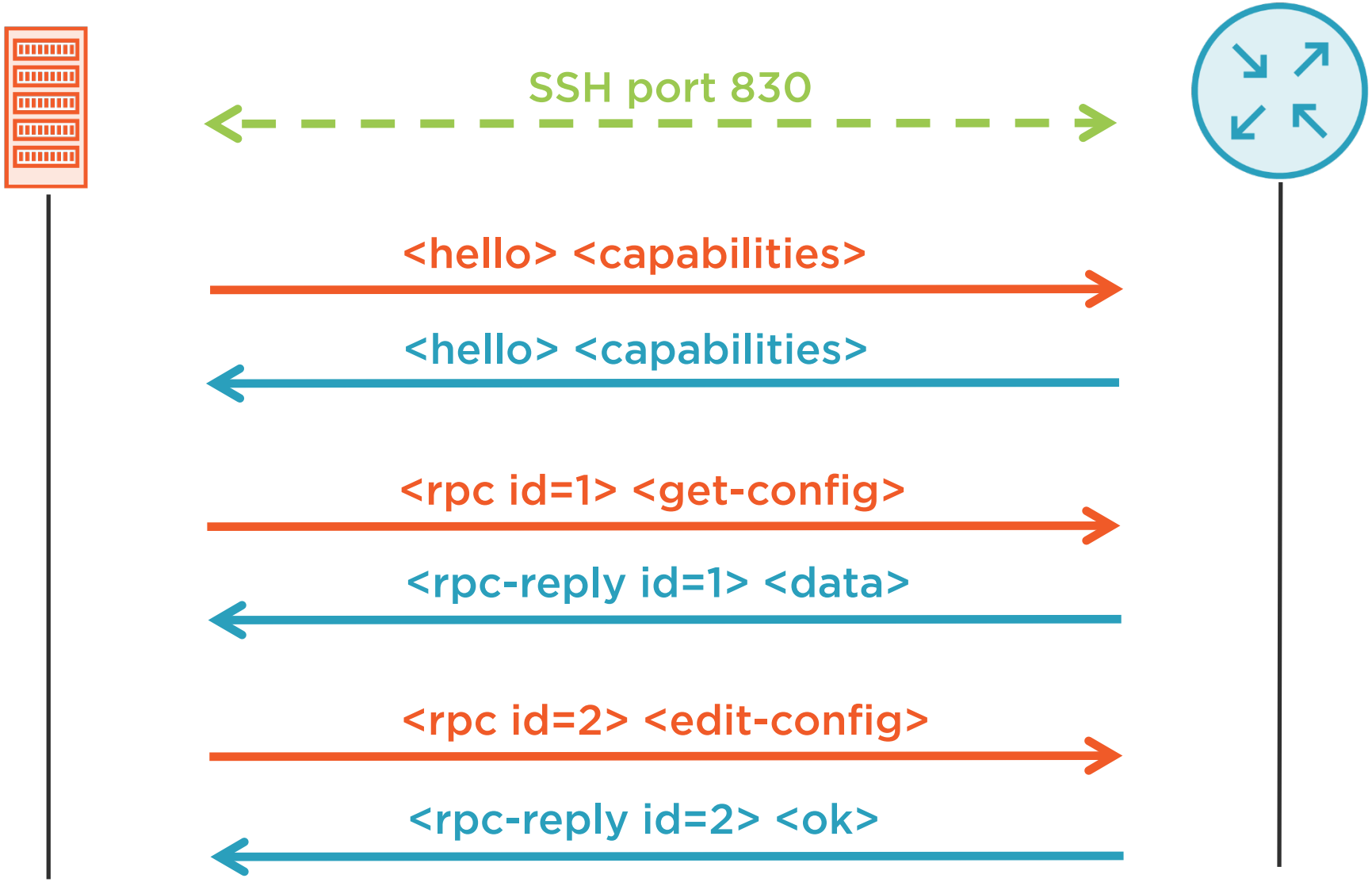
Who Cares?

Simplified implementation

Structured data

Standardized abstraction

Introducing NETCONF



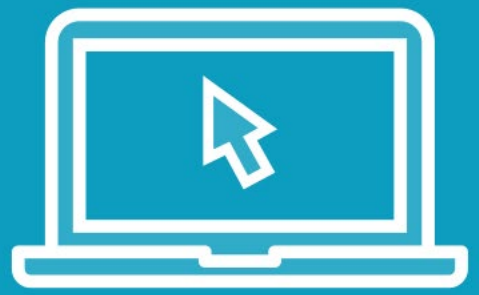
```
// ios-types
typedef asn-ip-type {
  type string {
    pattern "complex regex";
  }
}

// native VRF model
list export {
  description "lots of text";
  key "asn-ip";
  leaf asn-ip {
    type ios-types:asn-ip-type;
  }
}
```

- C-style modeling language
- What must an RT look like?
- We store export RTs in a list
- Each element has one key
- The value should conform to the type defined above

```
<export>
  <asn-ip>65000:1</asn-ip>
</export>
<import>
  <asn-ip>65000:1</asn-ip>
  <asn-ip>65000:2</asn-ip>
</import>
```

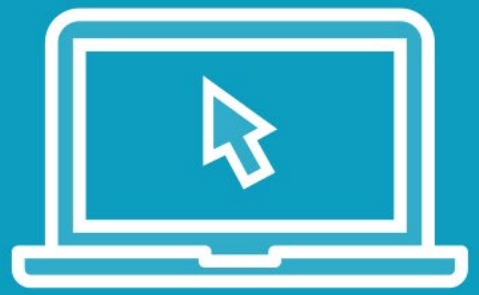
- List named "export"
- Elements named "asn-ip" with values conforming to typedef
- List named "import" with similar values

Demo



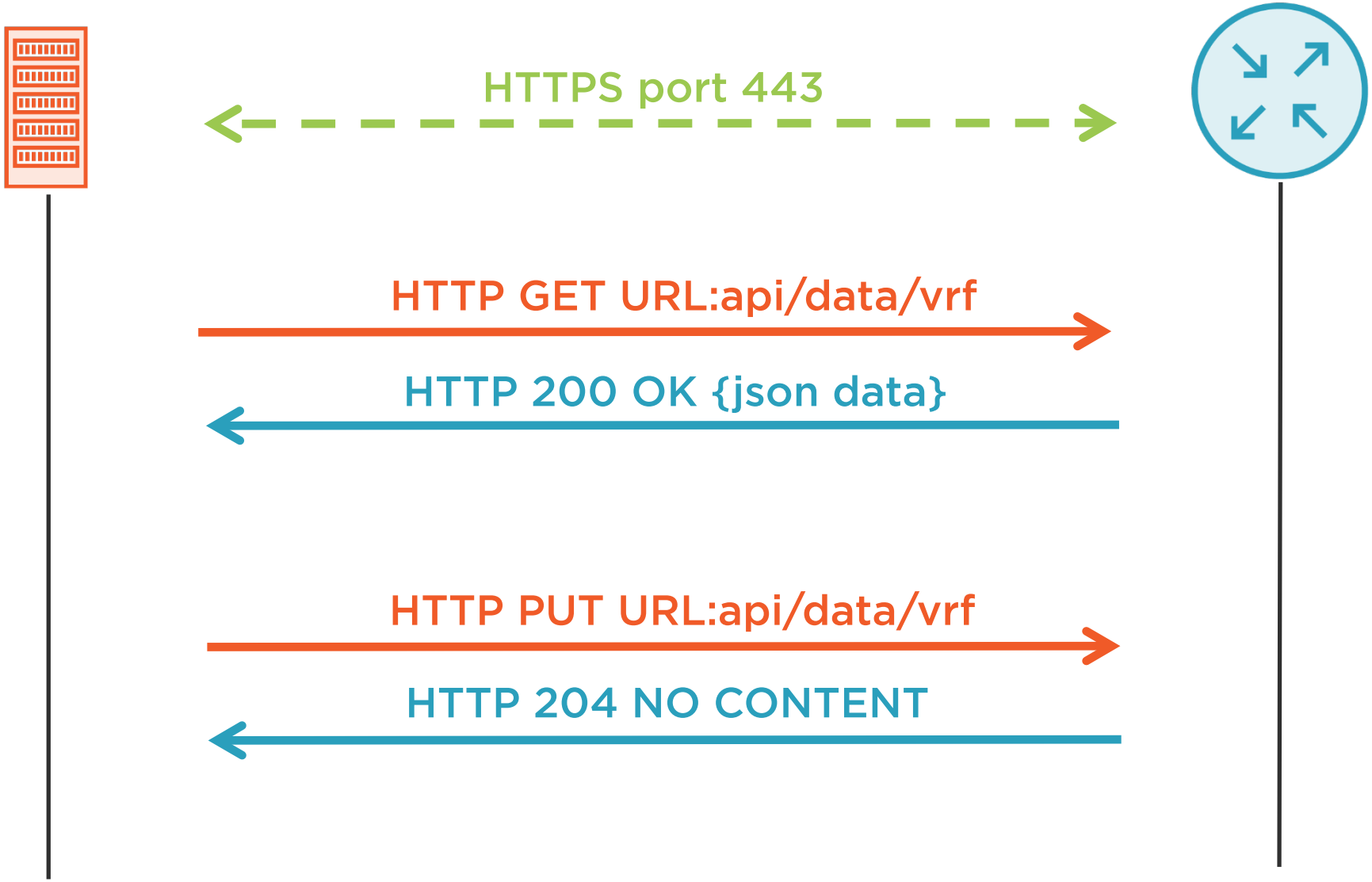
VRF "get-config" RPC with NETCONF

Demo



VRF "edit-config" RPC with NETCONF

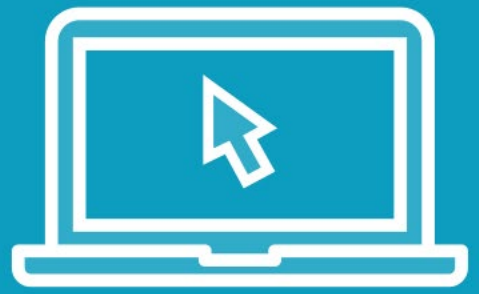
Introducing RESTCONF



```
"export": [
  {
    "asn-ip": "100:2"
  }
],
"import": [
  {
    "asn-ip": "100:1"
  }
]
```

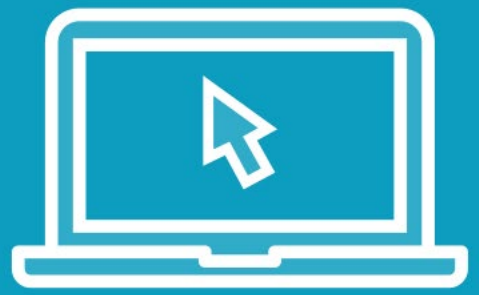
- List named "export"
- Elements named "asn-ip" with values conforming to typedef
- List named "import" with similar values

Demo



VRF "HTTP GET" with RESTCONF

Demo



VRF "HTTP PUT" with RESTCONF

Comparing NETCONF and RESTCONF

NETCONF	RESTCONF
SSH transport	HTTP transport
Custom RPCs	HTTP operations
XML only	XML or JSON
RFC 6241	RFC 8040