ALL IN

cisco Live!

DevNet Zone

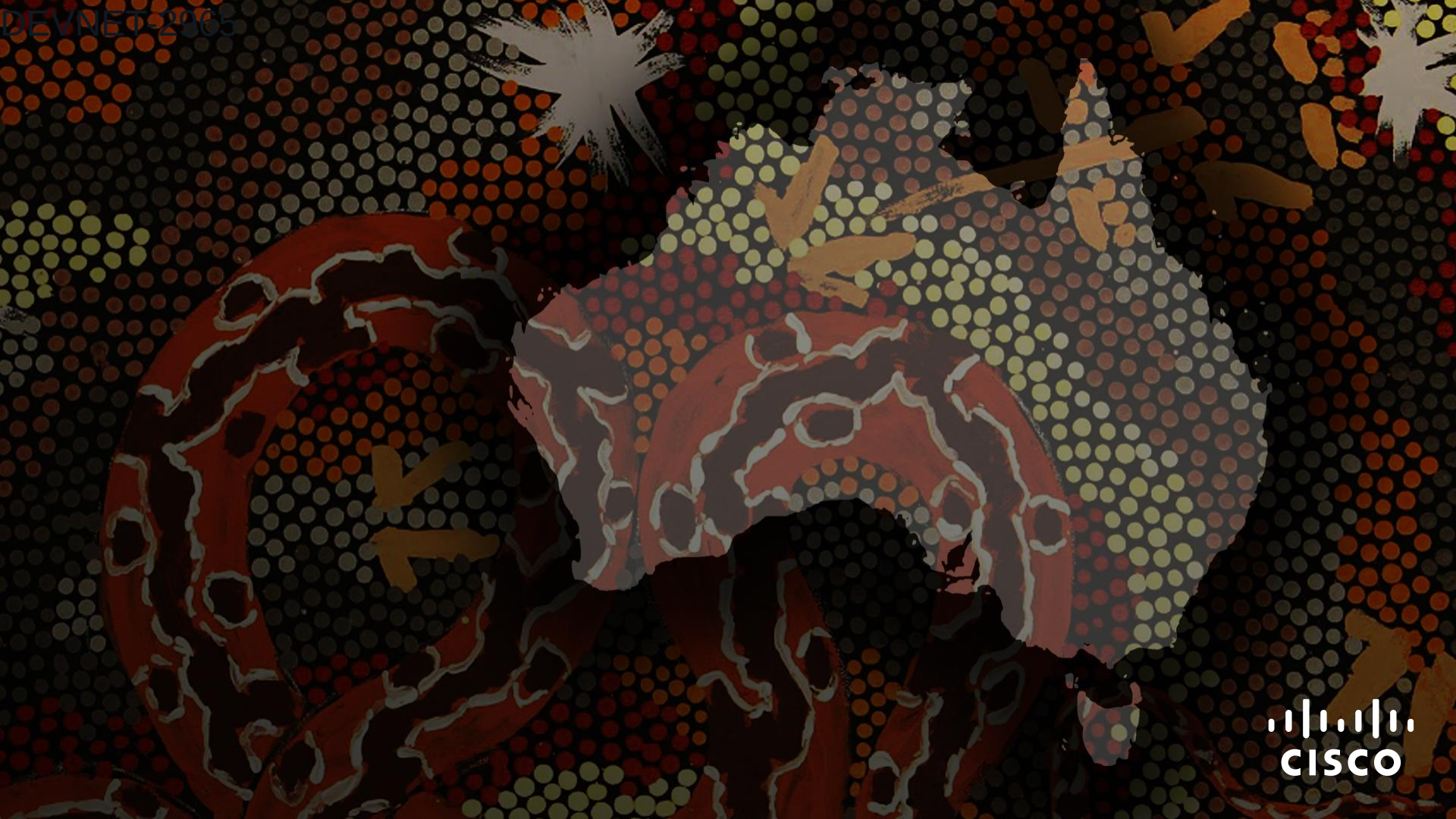
#CiscoLiveAPJC

The bridge to possible

Key learnings when creating a network device monitoring tool with Device APIs

Flo Pachinger, Developer Advocate
@flopachinger

DEVNET-2965



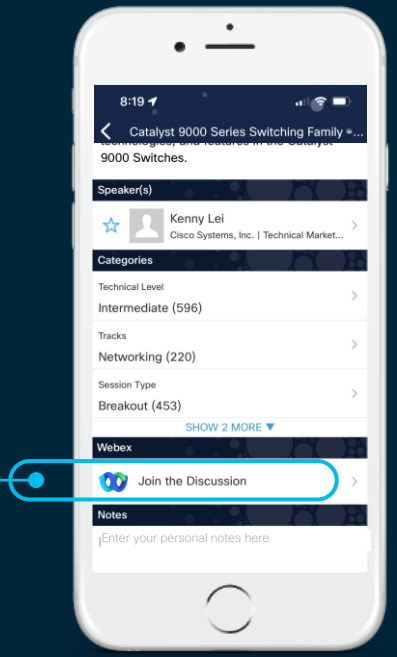
Cisco Webex App

Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space



https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-2695

Webex spaces will be moderated
by the speaker until Thursday 22 December, 2022.

Agenda

- WHY?
  - Introduction & Concept
- WHAT?
  - Showcasing network monitoring tool *johann*
  - Specific use-cases
- HOW? – Key Learnings
  - RESTCONF, NETCONF & REST-API
  - Django vs. Flask
  - xx

@flopachinger

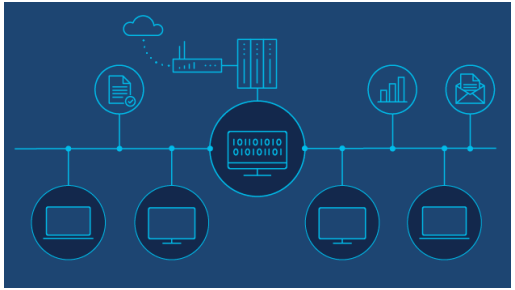
flopach

Flo Pachinger

Developer Advocate

- IoT, ML/Data Engineering
- Based in Vienna, Austria
- DevNet Projects:
  - Creator of johann (Network Monitoring Tool)
  - Po Robot Arm Challenge @ CiscoLive
  - Play Minecraft on Catalyst 9300 Switch
  - LoRaWAN Smart Parking (Cisco Office Frankfurt)

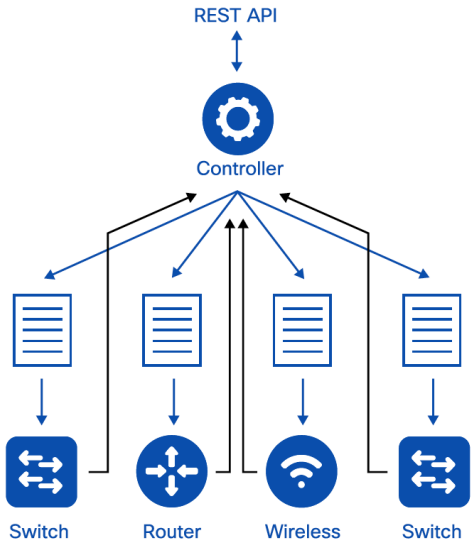
Why using a Network Monitoring Solution?



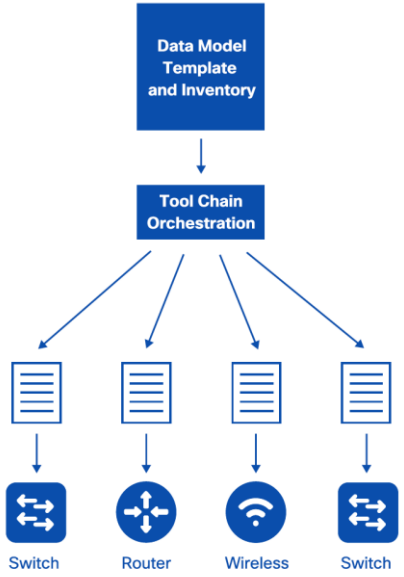
How to manage all these devices?

3 Operational Approaches

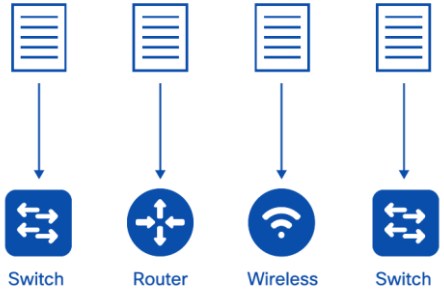
Controller-based (e.g. Cisco DNA Center)



Configuration Management Tool (e.g. Ansible)



Device-Level APIs (NETCONF/RESTCONF)



Example

Show status of all interfaces

```
Router# show ip interface brief

Interface              IP-Address      OK?    Method Status      Protocol
GigabitEthernet0/1     unassigned      YES    unset  up          up
GigabitEthernet0/2     192.168.190.235 YES    unset  up          up
GigabitEthernet0/3     unassigned      YES    unset  up          up
GigabitEthernet0/4     192.168.191.2   YES    unset  up          up
TenGigabitEthernet2/1  unassigned      YES    unset  up          up
TenGigabitEthernet2/2  unassigned      YES    unset  up          up
TenGigabitEthernet2/3  unassigned      YES    unset  up          up
TenGigabitEthernet2/4  unassigned      YES    unset  down        down
GigabitEthernet36/1    unassigned      YES    unset  down        down
GigabitEthernet36/2    unassigned      YES    unset  down        down
GigabitEthernet36/11   unassigned      YES    unset  down        down
GigabitEthernet36/25   unassigned      YES    unset  down        down
Te36/45                unassigned      YES    unset  down        down
Te36/46                unassigned      YES    unset  down        down
Te36/47                unassigned      YES    unset  down        down
Te36/48                unassigned      YES    unset  down        down
Virtual36              unassigned      YES    unset  up          up
```

Show status of all interfaces

The Output gives plain text

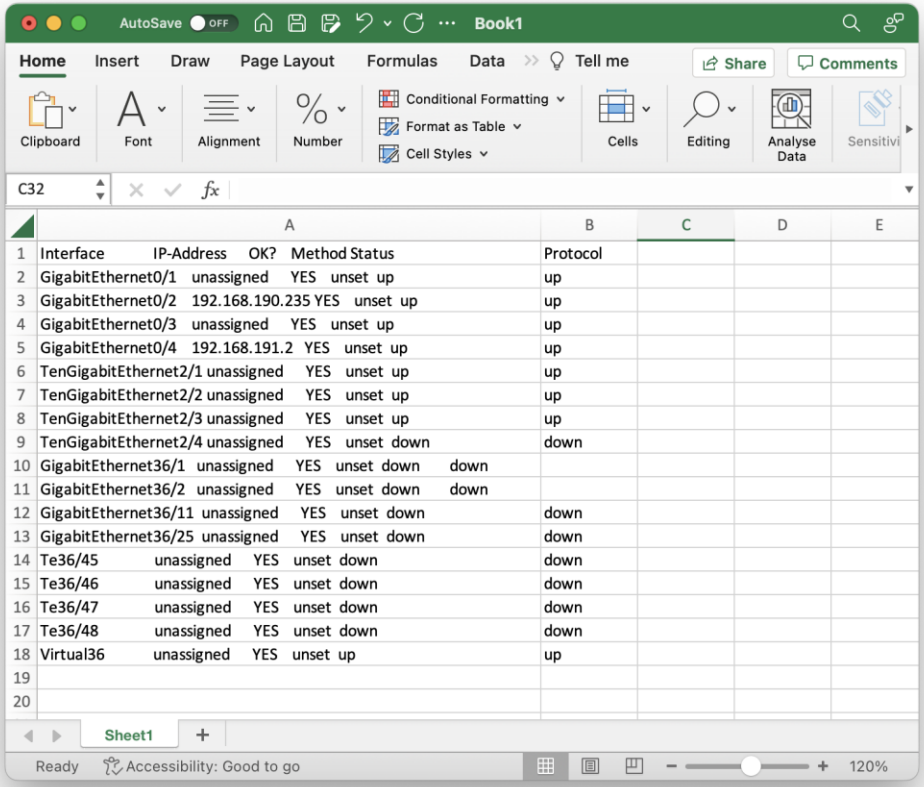
untitled

```
Interface              IP-Address      OK?   Method Status      Protocol
GigabitEthernet0/1     unassigned      YES   unset  up          up
GigabitEthernet0/2     192.168.190.235 YES   unset  up          up
GigabitEthernet0/3     unassigned      YES   unset  up          up
GigabitEthernet0/4     192.168.191.2   YES   unset  up          up
TenGigabitEthernet2/1  unassigned      YES   unset  up          up
TenGigabitEthernet2/2  unassigned      YES   unset  up          up
TenGigabitEthernet2/3  unassigned      YES   unset  up          up
TenGigabitEthernet2/4  unassigned      YES   unset  down        down
GigabitEthernet36/1    unassigned      YES   unset  down        down
GigabitEthernet36/2    unassigned      YES   unset  down        down
GigabitEthernet36/11   unassigned      YES   unset  down        down
GigabitEthernet36/25   unassigned      YES   unset  down        down
Te36/45                unassigned      YES   unset  down        down
Te36/46                unassigned      YES   unset  down        down
Te36/47                unassigned      YES   unset  down        down
Te36/48                unassigned      YES   unset  down        down
Virtual36              unassigned      YES   unset  up          up
```

untitled* 18:67 LF UTF-8 Plain Text GitHub Git (0)

Show status of all interfaces

Output copy/pasted in Excel



	A	B
1	Interface IP-Address OK? Method Status	Protocol
2	GigabitEthernet0/1 unassigned YES unset up	up
3	GigabitEthernet0/2 192.168.190.235 YES unset up	up
4	GigabitEthernet0/3 unassigned YES unset up	up
5	GigabitEthernet0/4 192.168.191.2 YES unset up	up
6	TenGigabitEthernet2/1 unassigned YES unset up	up
7	TenGigabitEthernet2/2 unassigned YES unset up	up
8	TenGigabitEthernet2/3 unassigned YES unset up	up
9	TenGigabitEthernet2/4 unassigned YES unset down	down
10	GigabitEthernet36/1 unassigned YES unset down down	
11	GigabitEthernet36/2 unassigned YES unset down down	
12	GigabitEthernet36/11 unassigned YES unset down	down
13	GigabitEthernet36/25 unassigned YES unset down	down
14	Te36/45 unassigned YES unset down	down
15	Te36/46 unassigned YES unset down	down
16	Te36/47 unassigned YES unset down	down
17	Te36/48 unassigned YES unset down	down
18	Virtual36 unassigned YES unset up	up
19		
20		

Show status of all interfaces

Standardizing & formatting the data

```
Router# show ip interface brief

Interface              IP-Address      OK?    Method Status      Protocol
GigabitEthernet0/1     unassigned      YES    unset  up          up
GigabitEthernet0/2     192.168.190.235 YES    unset  up          up
GigabitEthernet0/3     unassigned      YES    unset  up          up
GigabitEthernet0/4     192.168.191.2   YES    unset  up          up
TenGigabitEthernet2/1  unassigned      YES    unset  up          up
TenGigabitEthernet2/2  unassigned      YES    unset  up          up
TenGigabitEthernet2/3  unassigned      YES    unset  up          up
TenGigabitEthernet2/4  unassigned      YES    unset  down        down
GigabitEthernet36/1    unassigned      YES    unset  down        down
GigabitEthernet36/2    unassigned      YES    unset  down        down
GigabitEthernet36/11   unassigned      YES    unset  down        down
GigabitEthernet36/25   unassigned      YES    unset  down        down
Te36/45                unassigned      YES    unset  down        down
Te36/46                unassigned      YES    unset  down        down
Te36/47                unassigned      YES    unset  down        down
Te36/48                unassigned      YES    unset  down        down
Virtual36              unassigned      YES    unset  up          up
```

Table

Interface	IP	Status	…
GiEt0/1	n/a	UP	
GiEt0/2	192.168.190.235	UP	
GiEt0/3	n/a	UP	

Key-Value

```
"interfaces" : [
    { "name" : "GiEt0/1,
    "ip" : "na",
    "status" : "up"},
    { "name" : "GiEt0/2,
    "ip" : "192.168.190.235",
    "status" : "up"},
    { … }
]
```

NETCONF/ RESTCONF

is here to help

Device Level API: NETCONF/RESTCONF

Cisco
Switch

NETCONF/
RESTCONF

Client/
Application

IOS Configuration & YANG: Disable Interface

Device Configuration in XML, based on YANG model

```
<config>
 <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
  <interface>
   <name>GigabitEthernet1/10</name>
    <enabled>false</enabled>
  </interface>
 </interfaces>
</config>
```

Device Configuration via CLI

```
interface GigabitEthernet 1/10
shutdown
```

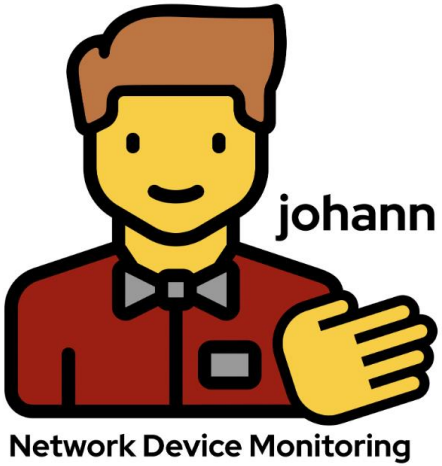
=

Device Configuration in JSON, based on YANG model

```
{
 "ietf-interfaces:interface": {
  "name": "GigabitEthernet1/10",
  "type": "iana-if-type:ethernetCsmacd",
  "enabled": false
 }
}
```

Showcasing *johann*

https://github.com/flopach/johann-network-device-monitoring



Network Device Monitoring

cisco DEVNET published

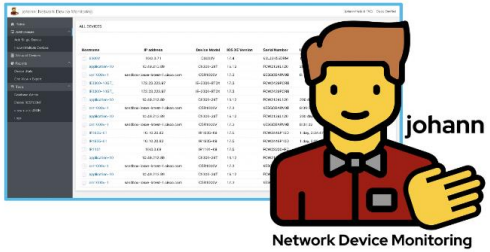
johann is a web-based network device monitoring tool for **Cisco IOS XE devices**. Collect configuration and operational data of your networking devices in a structured way in one single database!

Get an Overview in this YouTube Video!

IOS XE Devices

configurational + operational data

RESTCONF



johann: Architecture Overview

SCAN ME

python™

django

Dashboard

Database (SQLite)

RESTCONF Calls

Python Code

Docker Containers

HTTPS

IE 3400

(or any other IOS XE device!)

Why johann?

- **Get inspired**
  See johann as an inspiration of what you can do with the device APIs NETCONF/RESTCONF of networking devices
- **Use johann**
  If you need exactly a tool like that, just feel free to use it in your own lab or try it out with your IOS XE devices. It's read-only
- **Extend johann**
  If you like johann, feel free to add your code

Key Learnings

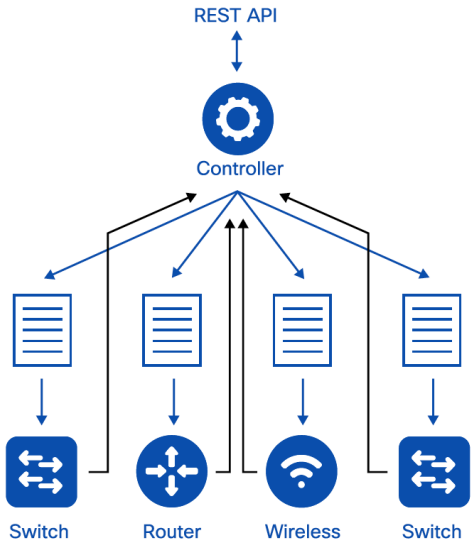
1. Define what data points you want to collect and check how you can do that in advance

Ask yourself:
What are you trying to solve?

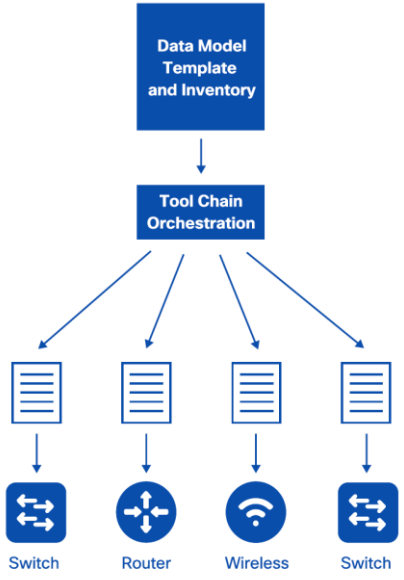
What data should be collected/changed for which use-case?

3 Operational Approaches

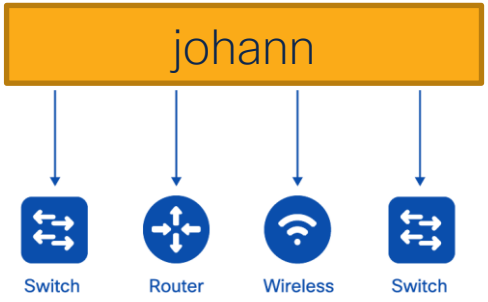
Controller-based (e.g. Cisco DNA Center)



Configuration Management Tool (e.g. Ansible)



Device-Level APIs (NETCONF/RESTCONF)

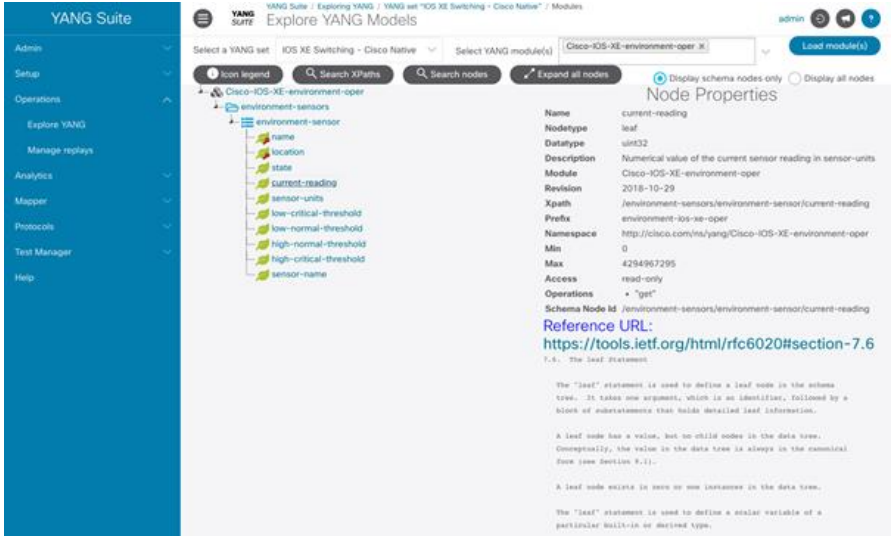


At device level:

- SSH Command Parsing
- NETCONF/RESTCONF
- 3rd Party or Open Source Library:
  - Python: paramiko, netmiko, NAPALM

NETCONF/RESTCONF

- Explore the YANG models
- Try it out on real devices
- Work with JSON/XML



2. Choose your programming language & application framework wisely

Why Python?

- Good NETCONF library: ncclient
- Personal experience, powerful and popular language
- Web-Frameworks available:
  - Django
  - Flask

Definition	high-level Python web framework	lightweight WSGI web application framework
Philosophy	rapid development and clean pragmatic design; monolithic	Minimalistic but extensible; modular
Time to create the first application	Takes a bit more time	Fast and easy to start with
Template engine	integrated	Expandable with Jinja2 template engine
Database	SQLite included per default	Not included per default
Database interaction	Integrated object-relation-mapping & data model support	Direct access only
Web project size	Mainly built for larger web projects	Mainly built for smaller to medium-sized web projects
Architecture approach	Full model-view-controller framework	Minimalistic, single-page application

3. Don’t underestimate the error handling and parsing work

Everything is plain text

untitled

```
Interface              IP-Address      OK?    Method Status      Protocol
GigabitEthernet0/1     unassigned      YES    unset  up          up
GigabitEthernet0/2     192.168.190.235 YES    unset  up          up
GigabitEthernet0/3     unassigned      YES    unset  up          up
GigabitEthernet0/4     192.168.191.2   YES    unset  up          up
TenGigabitEthernet2/1  unassigned      YES    unset  up          up
TenGigabitEthernet2/2  unassigned      YES    unset  up          up
TenGigabitEthernet2/3  unassigned      YES    unset  up          up
TenGigabitEthernet2/4  unassigned      YES    unset  down        down
GigabitEthernet36/1    unassigned      YES    unset  down        down
GigabitEthernet36/2    unassigned      YES    unset  down        down
GigabitEthernet36/11   unassigned      YES    unset  down        down
GigabitEthernet36/25   unassigned      YES    unset  down        down
Te36/45                unassigned      YES    unset  down        down
Te36/46                unassigned      YES    unset  down        down
Te36/47                unassigned      YES    unset  down        down
Te36/48                unassigned      YES    unset  down        down
Virtual36              unassigned      YES    unset  up          up
```

untitled* 18:67 LF UTF-8 Plain Text GitHub Git (0)

What if there is no NTP server configured?

```
try:
    ntp_server_list = []
    for ntp_server in r_content["ntp"]["Cisco-IOS-XE-ntp:server"]["server-list"]:
        ntp_server_list.append(ntp_server["ip-address"])
    self.new_device.ntp_server_list = ntp_server_list
except:
    pass
```

JSON Parsing all the way

Home Workspaces Reports Explore Search Postman Invite Upgrade

GET https://10.49.232.... No Environment

https://10.49.232.51:443/restconf/data/Cisco-IOS-XE-environment-oper:environment-sensors Save

GET https://10.49.232.51:443/restconf/data/Cisco-IOS-XE-environment-oper:environment-sensors Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Accept	application/yang-data+json	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 248 ms Size: 3.14 KB Save Response

Pretty Raw Preview Visualize JSON


```
1  {
2      "Cisco-IOS-XE-environment-oper:environment-sensors": {
3          "environment-sensor": [
4              {
5                  "name": "Inlet Temp Sensor",
6                  "location": "Switch 1",
7                  "state": "Norm",
8                  "current-reading": 20,
9                  "sensor-units": "celsius",
10                 "low-critical-threshold": 0,
11                 "low-normal-threshold": 0,
12                 "high-normal-threshold": 56,
13                 "high-critical-threshold": 0,
14                 "sensor-name": "temperature"
15             },
16             {
17                 "name": "Outlet Temp Sensor",
18                 "location": "Switch 1",
19                 "state": "Norm",
20                 "current-reading": 25,
21                 "sensor-units": "celsius",
22                 "low-critical-threshold": 0,
23                 "low-normal-threshold": 0,
```


Find and Replace Console Bootcamp Runner Trash

4. Use a database which fits best for your project

Relation or no relation?

Relational Database

- All data in “a table”
- Query via SQL

Interface	IP	Status	…
GiEt0/1	n/a	UP	
GiEt0/2	192.168.190.235	UP	
GiEt0/3	n/a	UP	

Non–relational Databases

- Save data in a key-value format

```
“interfaces” : [
    { ”name” : “GiEt0/1,
    “ip” : “na”,
    “status” : “up”},
    { ”name” : “GiEt0/2,
    “ip” : “192.168.190.235”,
    “status” : “up”},
    { … }
]
```

Django Database API

Create a data model

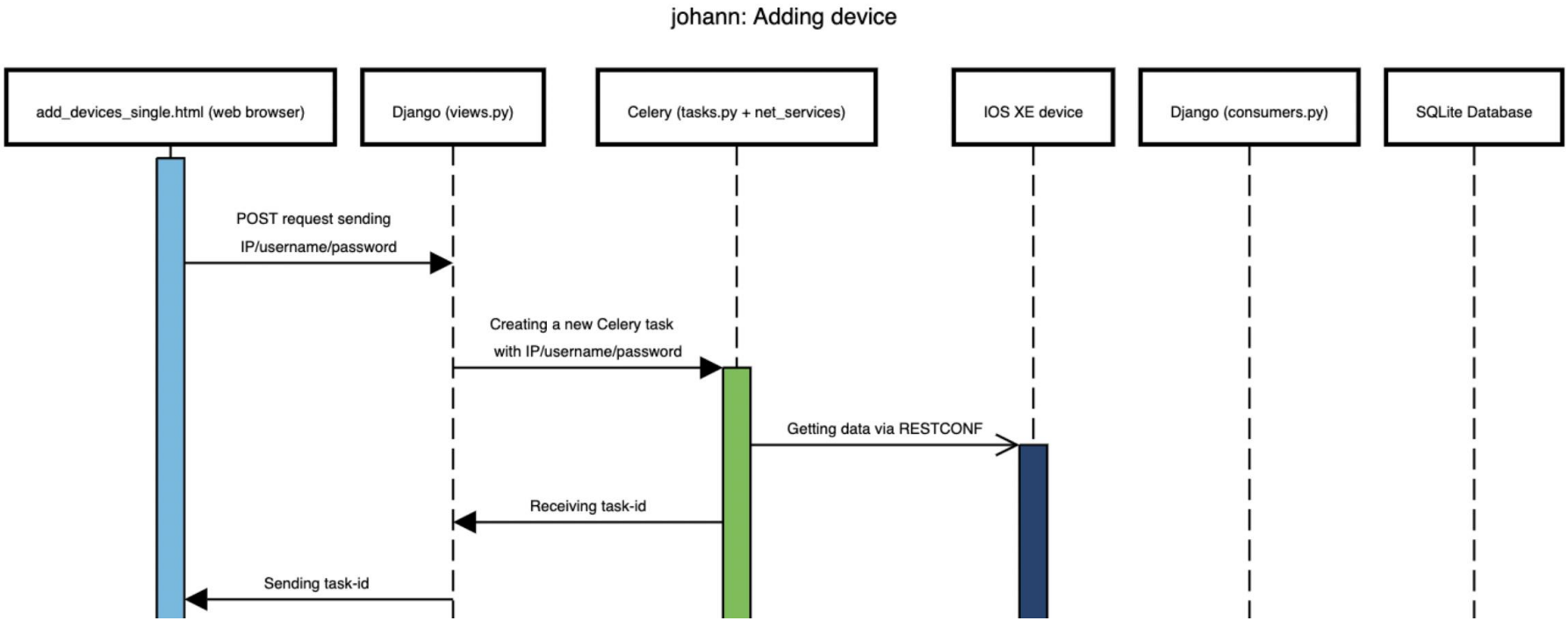
```
from django.db import models

class Blog(models.Model):
    name = models.CharField(max_length=100)
    tagline = models.TextField()

    def __str__(self):
        return self.name
```

Add data

```
>>> from blog.models import Blog
>>> b = Blog(name='Beatles Blog', tagline='All the latest Beatles news.')
>>> b.save()
```

6. Go web-based and use an async queue

Django & celery in detail



Conclusions

All Key Learnings

1. Define what data points you want to collect and check how you can do that in advance
2. Choose your programming language & application framework wisely
3. Don’t underestimate the error handling and parsing work
4. Use the a database which fits best for your project
5. Go web-based and use an async queue

3 things to remember

- APIs & Programmability…
  - …help you to create new innovations, products and services
  - …will save you and your customers time and costs
- RESTCONF/NETCONF and REST APIs are the standardized interfaces which makes the magic happen
- Cisco IOS XE based (industrial) devices support these awesome features – use them!

Next Steps for you? Start small, step by step.

Resources

- https://blogs.cisco.com/tag/johann
- https://github.com/flopach/johann-network-device-monitoring

Session Surveys

We would love to know your feedback on this session!

- Complete the session surveys in the Cisco Events mobile app. You'll earn some points in the Cisco Live Game and potentially win a prize.
- Complete a minimum of four session and the overall event surveys to claim a Cisco Live cable bag.

Continue your education

Visit the Cisco Showcase for related demos

Book your one-on-one Meet the Expert meeting

Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs

Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Cisco Learning and Certifications

From technology training and team development to Cisco certifications and learning plans, let us help you empower your business and career. www.cisco.com/go/certs

Pay for Learning with Cisco Learning Credits

(CLCs) are prepaid training vouchers redeemed directly with Cisco.

Learn

Cisco U.
IT learning hub that guides teams and learners toward their goals

Cisco Digital Learning
Subscription-based product, technology, and certification training

Cisco Modeling Labs
Network simulation platform for design, testing, and troubleshooting

Cisco Learning Network
Resource community portal for certifications and learning

Train

Cisco Training Bootcamps
Intensive team & individual automation and technology training programs

Cisco Learning Partner Program
Authorized training partners supporting Cisco technology and career certifications

Cisco Instructor-led and Virtual Instructor-led training
Accelerated curriculum of product, technology, and certification courses

Certify

Cisco Certifications and Specialist Certifications
Award-winning certification program empowers students and IT Professionals to advance their technical careers

Cisco Guided Study Groups
180-day certification prep program with learning and support

Cisco Continuing Education Program
Recertification training options for Cisco certified individuals

Thank you

cisco Live!

ALL IN

#CiscoLiveAPJC