aruba

a Hewlett Packard Enterprise company

python™

Campus Automation

Joe Neville – Aruba Consulting System Engineer - @joeneville_

Agenda

- Network Automation – What is it and why is it?
- Aruba, Python and APIs
- Fingers crossed (demo time) 
- Start Small - Building an ecosystem

DISCLAIMER

You don’t need to know any of this stuff*

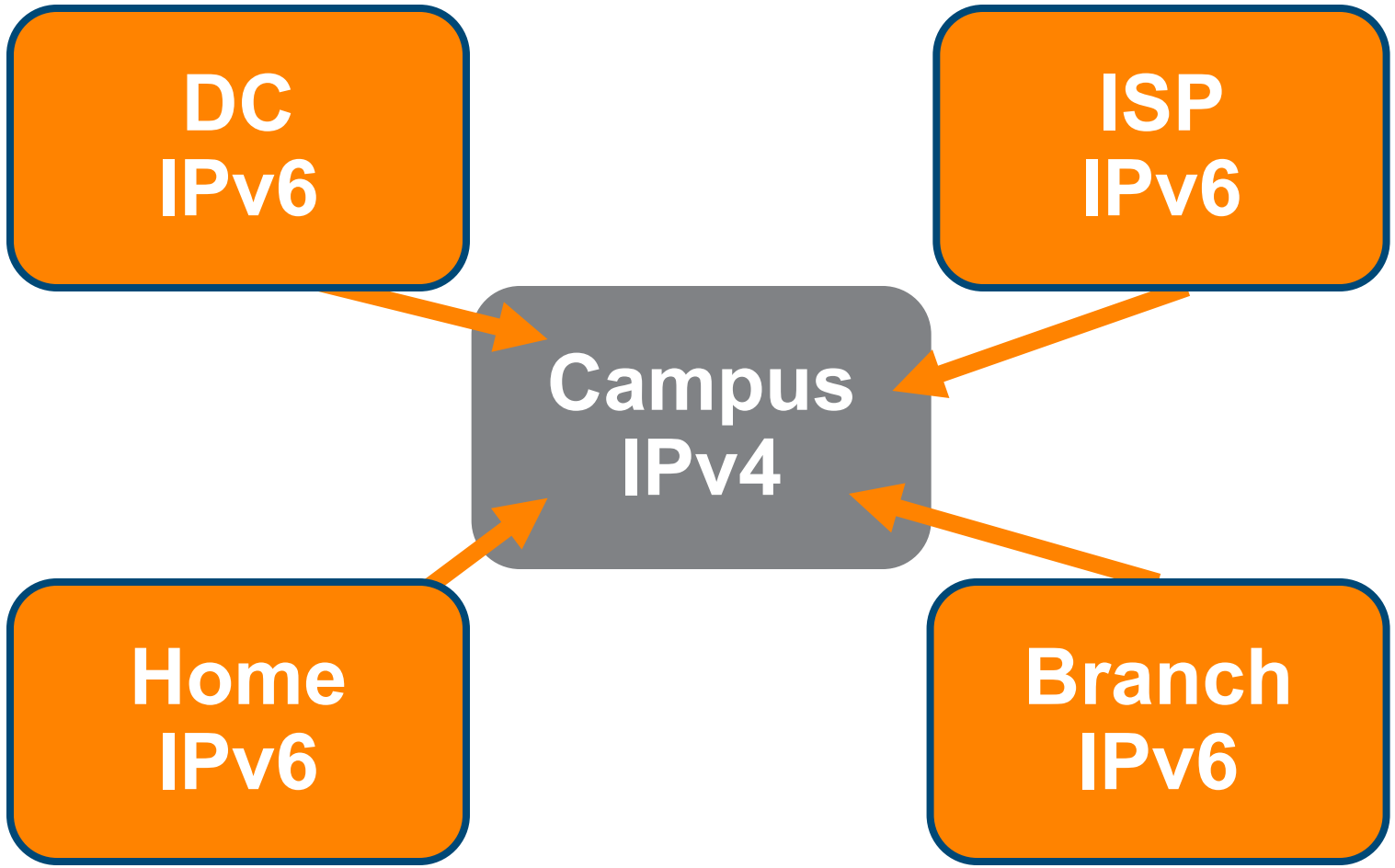
ArubaOS-Switch & CX = full CLI

**But you’ll be glad that you do*

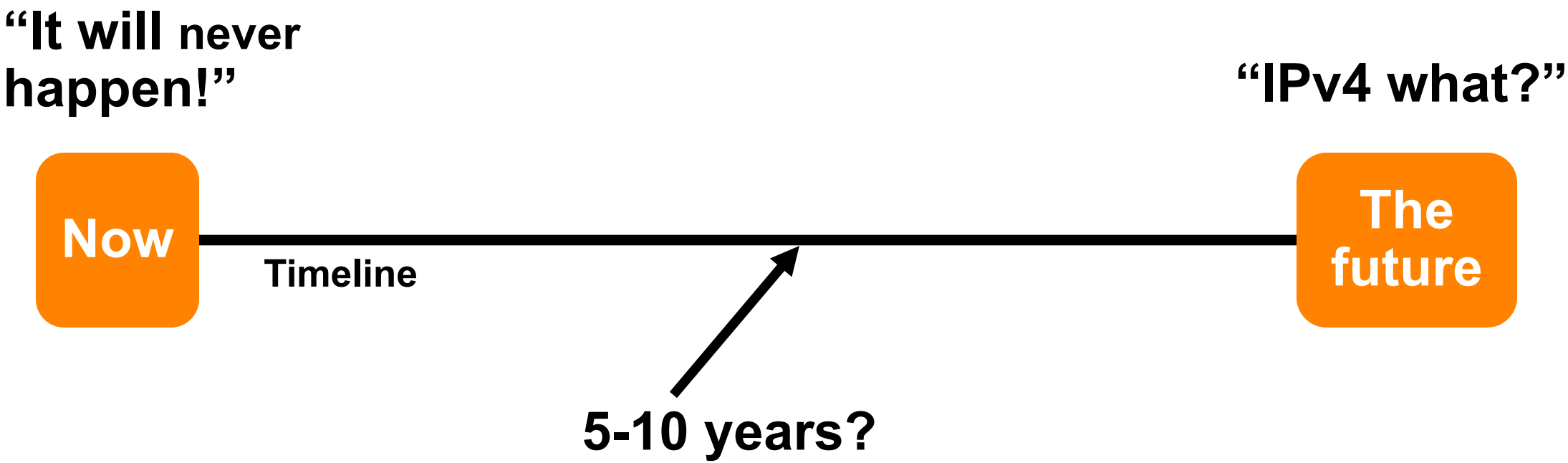
IPv6 in the Campus

“It will never happen!”

The v4 Island?



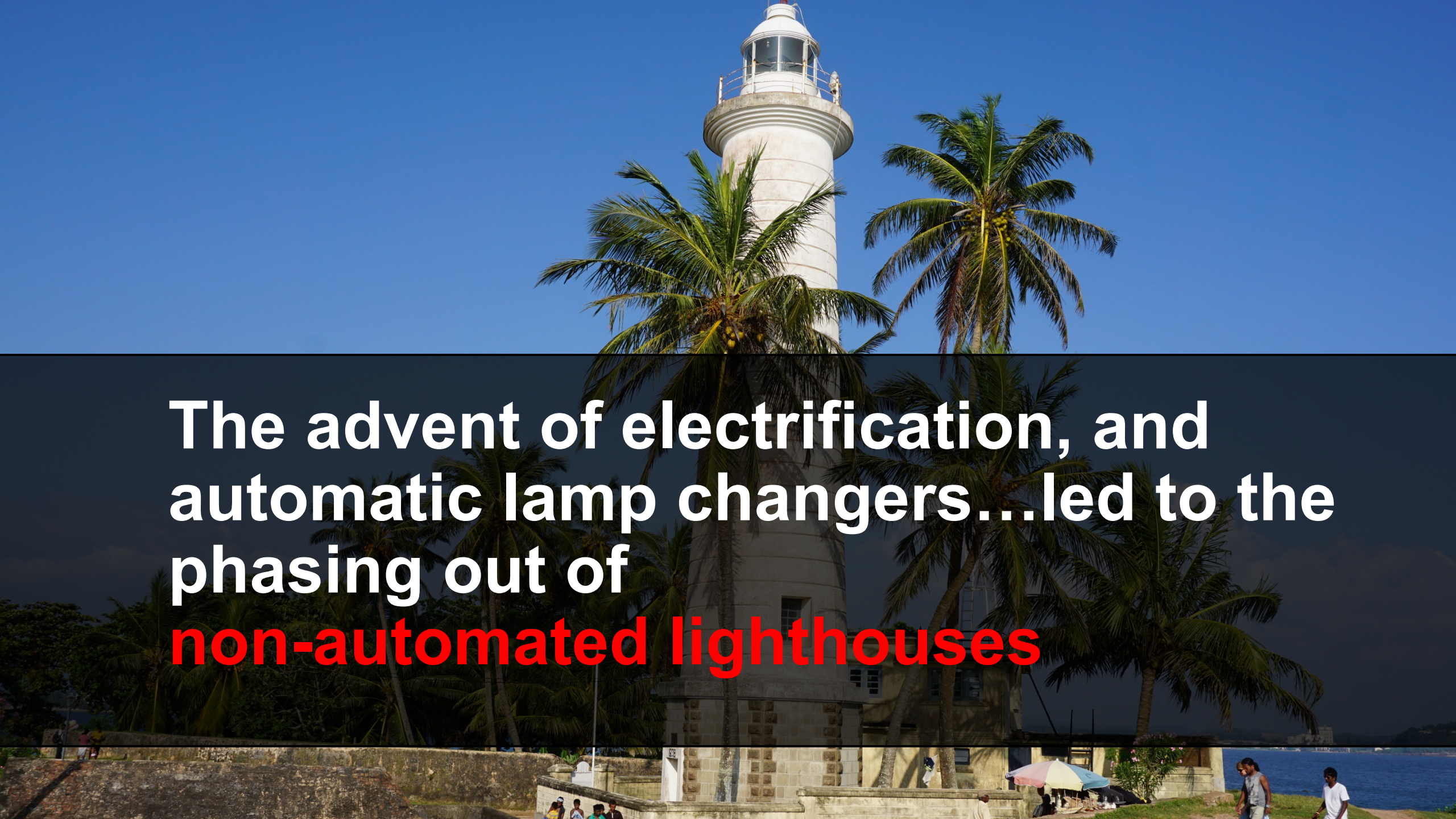
IPv6 in the Campus



Campus Automation

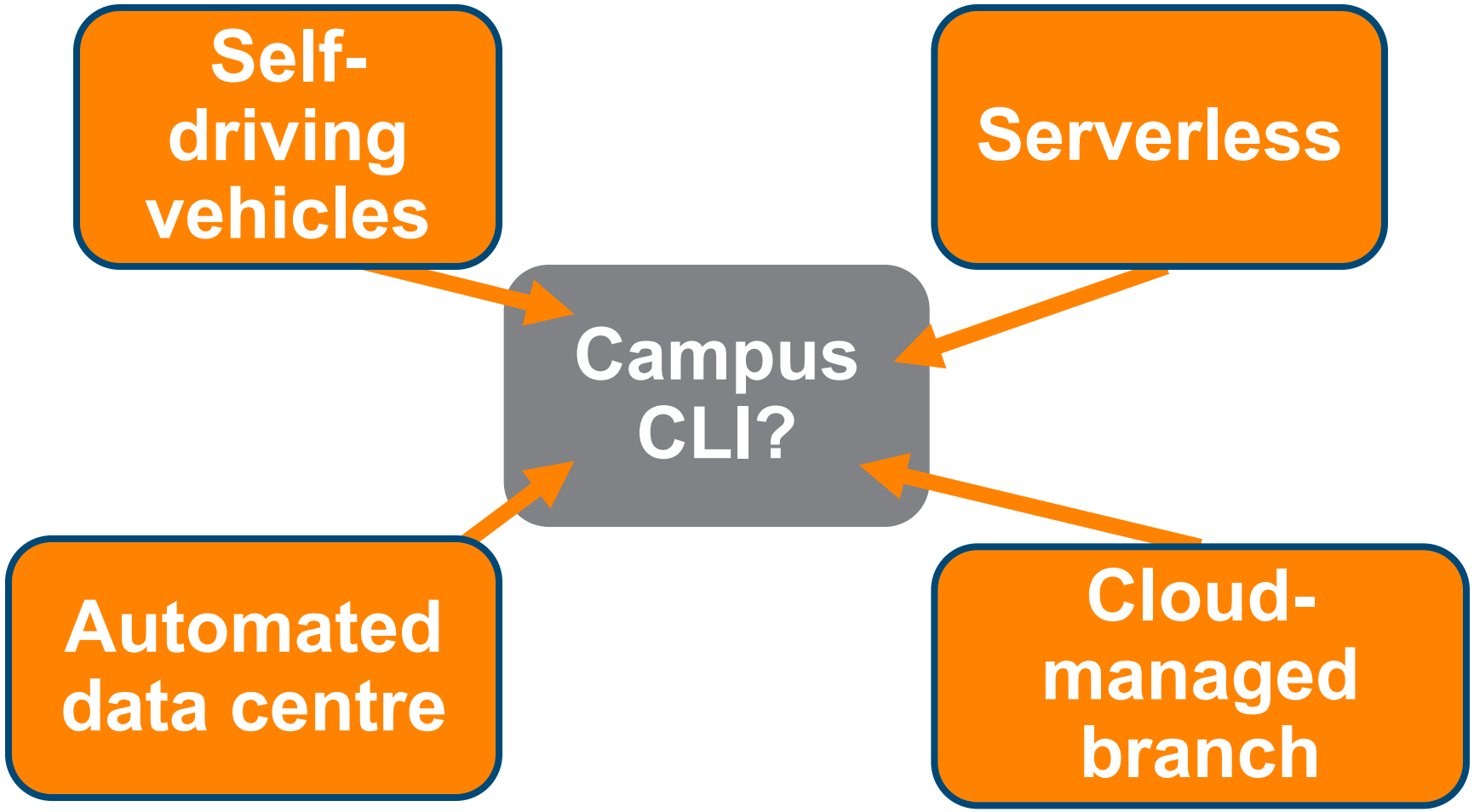
“It will never happen!”

- Network Engineers use CLI & SNMP.
- They are not programmers!
- Heard it all before with SDN.



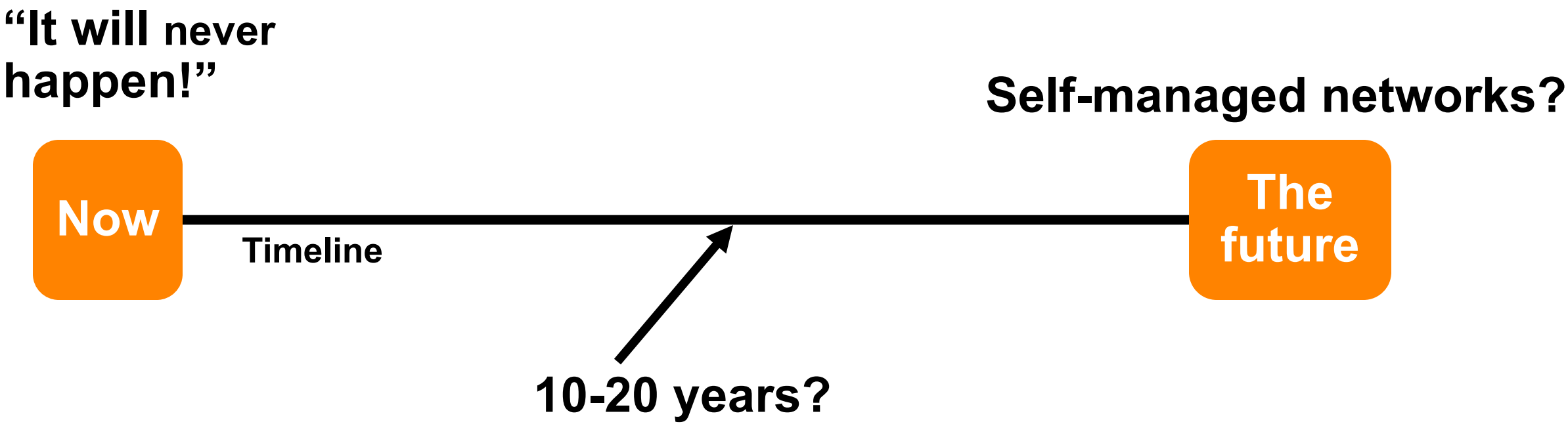
The advent of electrification, and automatic lamp changers…led to the phasing out of non-automated lighthouses

The Campus CLI Island?



aruba
a Hewlett Packard Enterprise company

Automation in the Campus



Network Automation – Define our terms

Static

- Static / manual configuration
- Primarily using CLI
- Large majority of campus customers use these practises.

Automation

- Combine multiple processes into workflows.
- Tasks can pass through multiple phases without manual intervention.
- Programmatic. “If this is true, do that”

Network Automation – Define our terms

Automation

- Most networks already use an automated process for IP address assignment (DHCP)
- Lots of others use Zero Touch Provisioning.
- ZTP uses a process flow to get a switch up and running with minimal intervention from NetOps.
- We're now exploring other process flows, at different stages of device lifecycle (e.g. day to day ops / MACs)

So what's wrong with CLI?

- Humanly readable
  - Slow & cumbersome (forcing a machine to speak English)
- Slow
  - Interacting via CLI, even if scripted, is slow (SSH in, input commands, parse output)
- Response is unstructured
- Networks getting bigger (IoT)
- **Customer looking for new ways to automate (like they did with servers)**

Network Automation

- **So CLI is dead?** 
- No despite some clickbait out there, CLI is a tool in the toolbox and a very valuable one.
- CLI is the tool for humans.
- But we want to speed-up and lessen human interaction
- It is time for some new tools
- Like what….

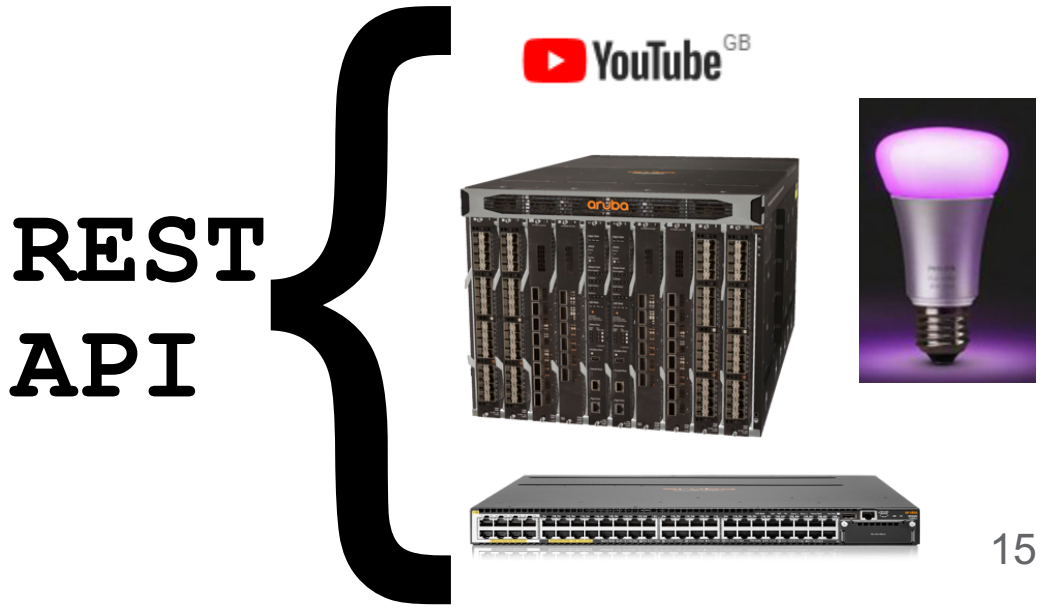
Enter the API

- **Application Programming Interface:**

  “a set of functions and procedures that allow the creation of applications which access the features or data of an operating system, application, or other service.”

  – google dictionary

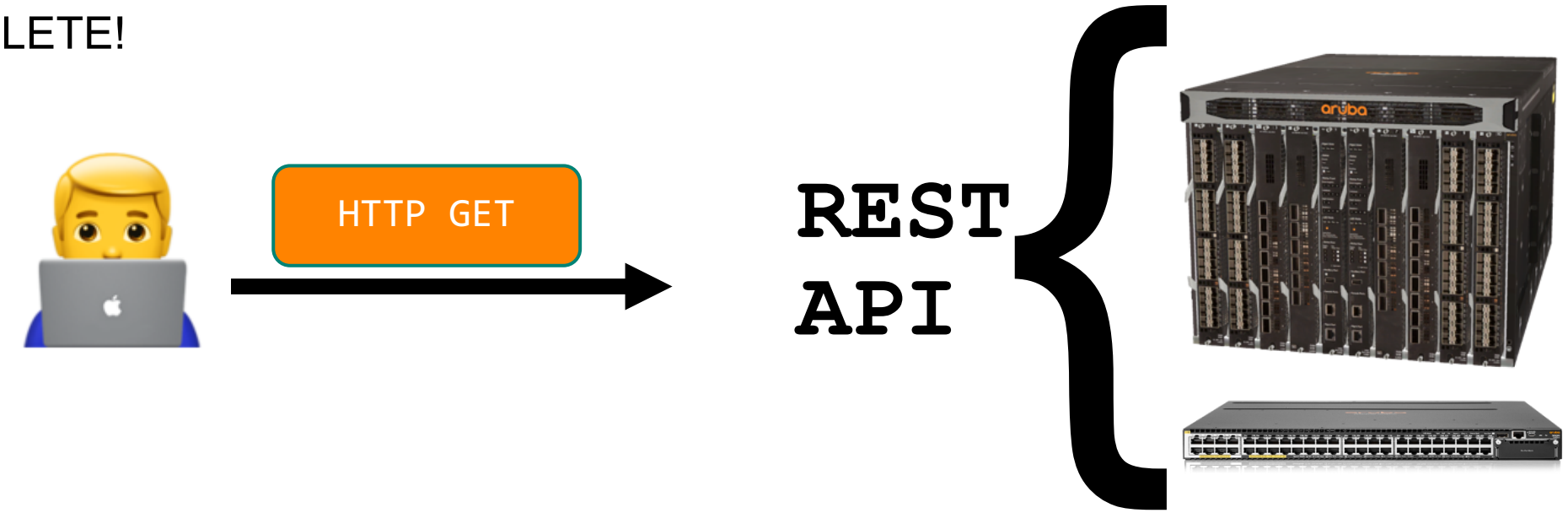
- **Modern approach: REST API = Representational state transfer**
  - Stateless call and response method to interact with an OS, application and now networking devices
  - Popular: fridges, youtube, ArubaOS-Switches and ArubaOS-CX have REST APIs.



Enter the API

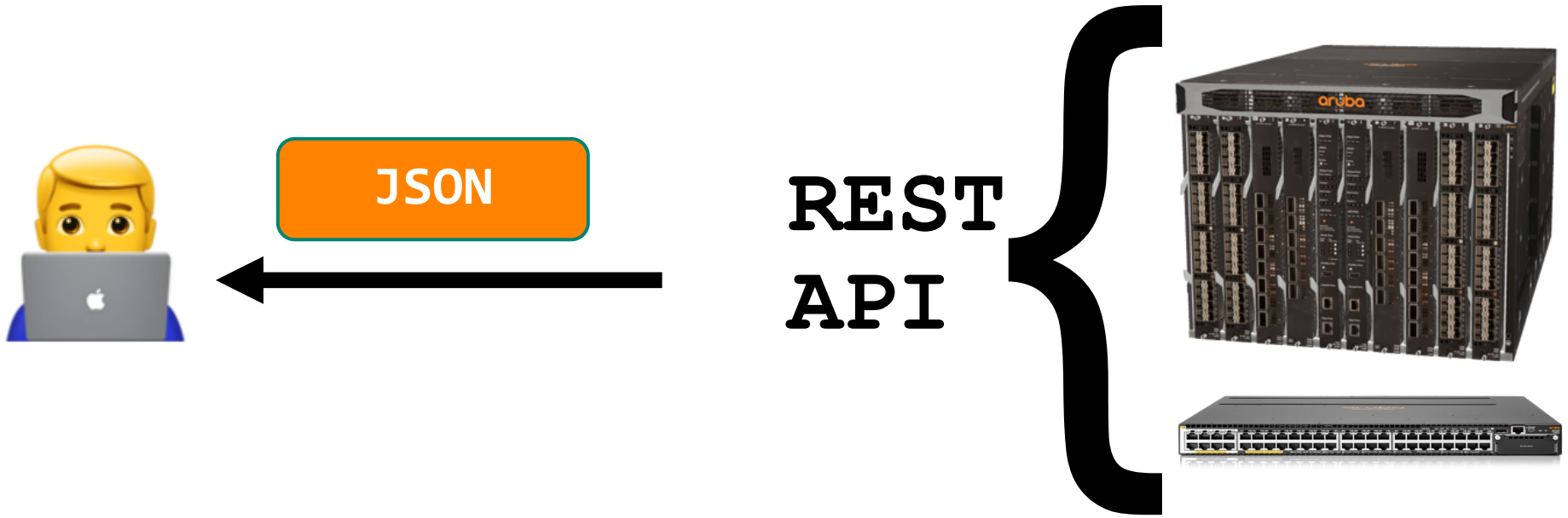
REST concept will be familiar to most:

- Based on HTTP methods to perform an action on a device:GET, POST, PUT, DELETE
- Fire HTTP at a device, get a response. No open channel between Sender and Receiver hence stateless
- show command = HTTP GET to URI
- New config = POST, amend config = PUT
- Delete is DELETE!



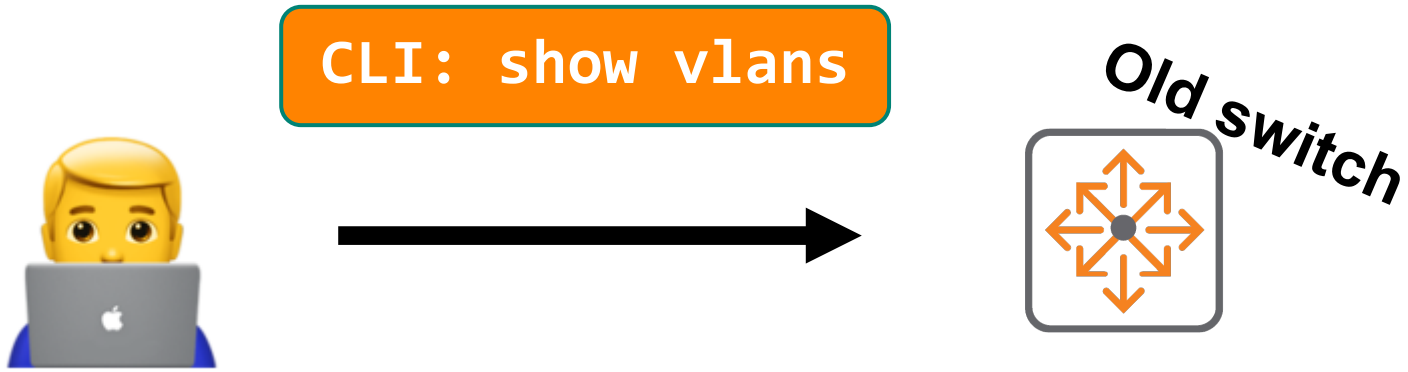
Enter the API

- Data is returned as JavaScript Object Notation – structured data
- THIS IS KEY
- We can quickly mine the data we need



Enter the API

- Without an API – previous scripting (PERL)
- Slow, log in, issue CLI command, log out….THEN

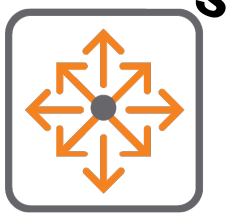


Enter the API

- Without an API – previous scripting (PERL)
- Slow, log in, issue CLI command, log out….THEN
- CLI OUTPUT = semi-structured (whitespace, column headers) to make it readable to humans = superfluous information
- Must be parsed to get the salient data = work / regex
- API structured data makes our lives easier

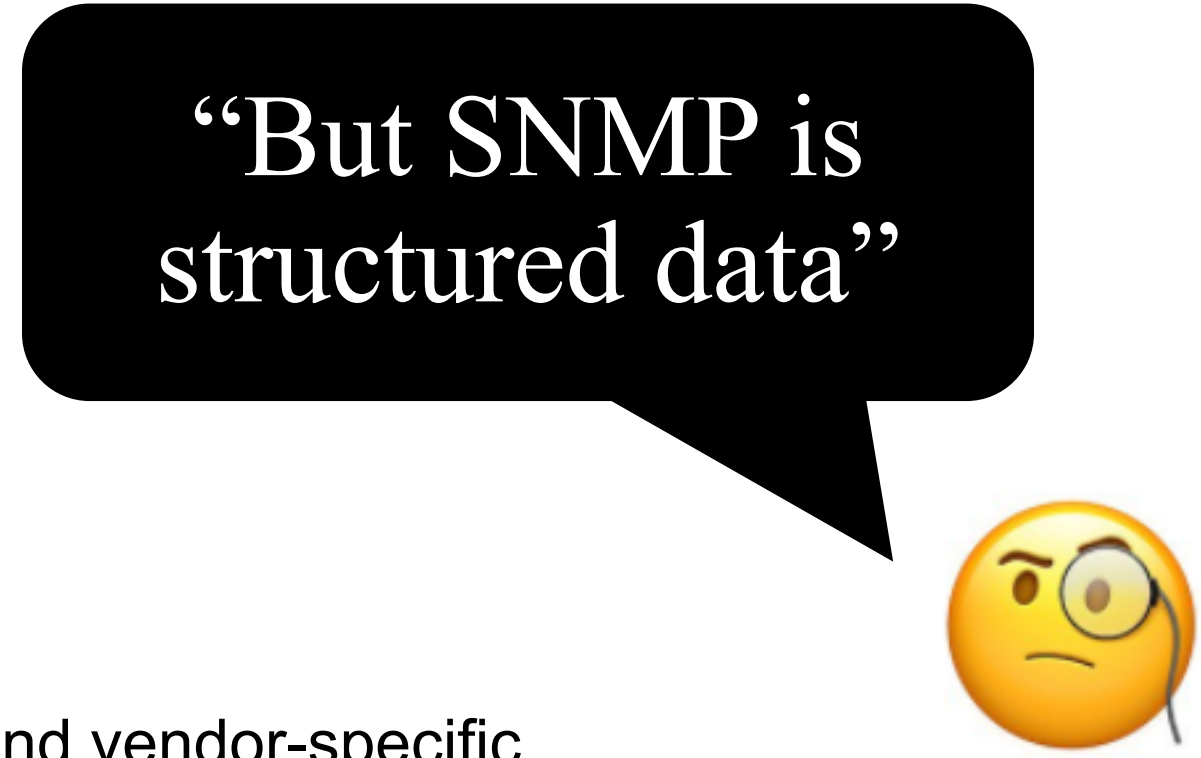
```
sw1# sh vlan

-------------------------------------------------------------------------------------------
VLAN  Name                                Status  Reason               Type      Interfaces
-------------------------------------------------------------------------------------------
1     DEFAULT_VLAN_1                      down    no_member_forwarding  default   1/1/2
2     VLAN_2                              up                            default   1/1/5
3     VLAN_3                              up                            default   1/1/6
4     VOICE                               up                            default   1/1/7
5     VIDEO                               up                            default   1/1/8
```

CLI:



Old switch

Cue this response:



- Sure, but SNMP…
- Complex, specialized and vendor-specific.
- Reality: CLI & SNMP hasn’t got us where we want to be – lots of manual changes

REST API != SNMP / CLI

- Not only faster and quicker to mine data
- REST API not a networking specific feature. They are on all your home automation kit and fave social media sites
- Barrier to entry much lower – technical types in all areas of IT touch them / kids out of college can hack on them

Aruba ❤️ APIs

ArubaOS-CX

Auto-generated from database
100% day one

8400

8320

Aruba 🧡 APIs

ArubaOS-Switch

Started in 16.02
Now on v4 as rollout continues

Where does Python fit in?

- Use code to build REST API calls and handle the JSON
- Python is a good candidate language

Why Python?

- Popular – lots of examples online and tutorials
- Widely supported – default on linux distros, easy Wins install
- Third-party library support – especially ‘requests’ i.e. someone has already written the code we need
- Quite readable…dare I say fun?

Basic operations with Python

1. Login – send HTTP POST to https://<ipaddr>/rest/v1/login

Basic operations with Python

1. Login – send HTTP POST to https://<ipaddr>/rest/v1/login
2. Response from CX

Basic operations with Python

1. Login – send HTTP POST to https://<ipaddr>/rest/v1/login
2. Response from CX
3. Get VLAN table

Basic operations with Python

1. Login – send HTTP POST to https://<ipaddr>/rest/v1/login
2. Response from CX
3. Get VLAN table
4. JSON returned

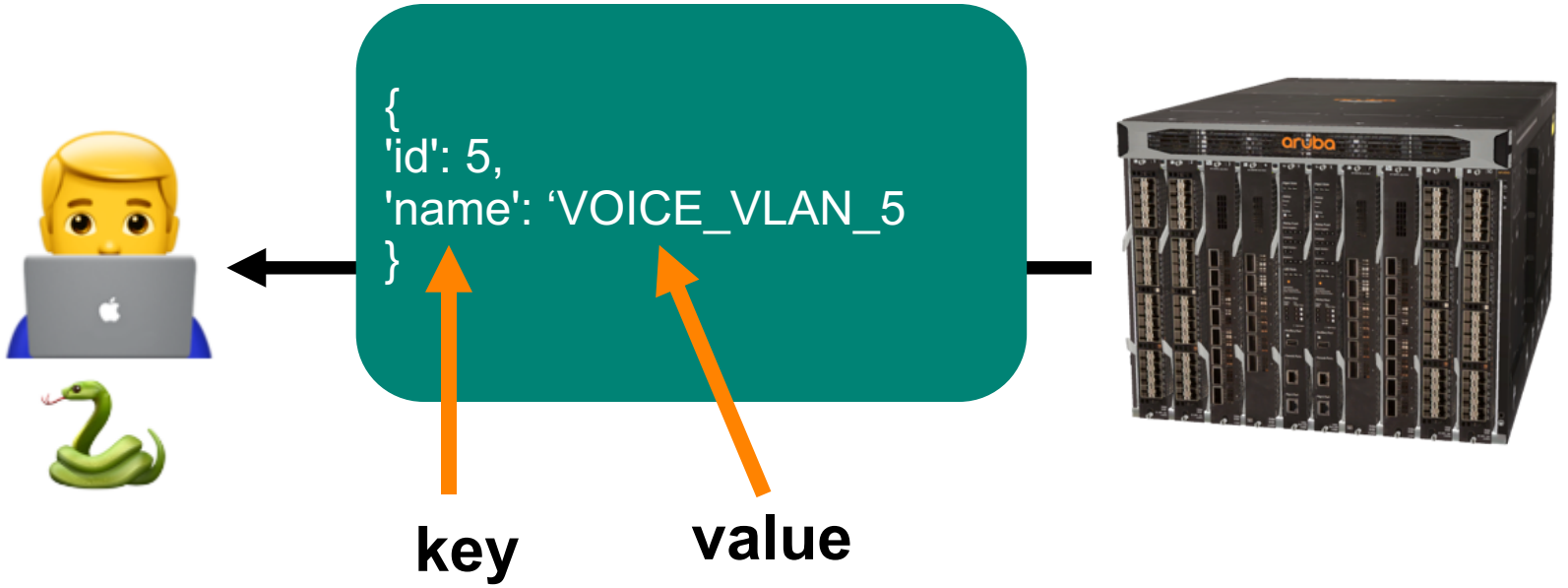

```
[{'aclv4_in_statistics': {},
  'aclv4_in_status': {},
  'aclv4_out_statistics': {},
  'aclv4_out_status': {},
  'aclv6_in_statistics': {},
  'aclv6_in_status': {},
  'aclv6_out_statistics': {},
  'aclv6_out_status': {},
  'admin': 'up',
  'flood_enabled_subsystems': ['/rest/v1/system/subsystems/system/base'],
  'id': 1,
  'internal_usage': {},
  'mgmd_counters': {'igmp_fast_leave_counter': 0,
                    'igmp_forced_fast_leave_counter': 0,
                    'igmp_membership_timeout_counter': 0,
                    'igmp_rx_bad_checksum_counter': 0,
                    'igmp_rx_malformed_pkt_counter': 0,
                    'igmp_rx_unknown_msg_type_counter': 0,
                    'igmp_rx_v1_all_host_query_counter': 0,
                    'igmp_rx_v1_member_report_counter': 0,
                    'igmp_rx_v2_all_host_query_counter': 0,
                    'igmp_rx_v2_group_specific_query_counter': 0,
                    'igmp_rx_v2_member_leave_counter': 0,
                    'igmp_rx_v2_member_report_counter': 0,
                    'igmp_rx_v3_all_host_query_counter': 0,
                    'igmp_rx_v3_group_specific_query_counter': 0,
                    'igmp_rx_v3_gssq_counter': 0,
                    'igmp_rx_v3_member_report_counter': 0,
                    'igmp_rx_wrong_version_query_counter': 0,
                    'igmp_tx_flood_on_vlan': 0,
                    'igmp_tx_forward_to_routers': 0,
                    'igmp_tx_v2_group_specific_query_counter': 0,
                    'igmp_tx_v3_group_specific_query_counter': 0},
  'mgmd_drop_unknown_status': {},
  'mgmd_dynamic_group_count': {'igmp_exclude_mode': 0, 'igmp_include_mode': 0},
  'mgmd_enable': {},
  'mgmd_enable_status': {},
  'mgmd_igmp_block_ports': [],
  'mgmd_igmp_fastleave_ports': [],
  'mgmd_igmp_forcedfastleave_ports': [],
  'mgmd_igmp_forward_ports': [],
  'mgmd_igmp_router_port_time_expiry': {},
  'mgmd_igmp_static_groups': [],
  'mgmd_mld_block_ports': [],
  'mgmd_mld_fastleave_ports': [],
  'mgmd_mld_forcedfastleave_ports': [],
  'mgmd_mld_forward_ports': [],
  'mgmd_mld_router_port_time_expiry': {},
  'mgmd_mld_static_groups': [],
  'mgmd_oper_version': {},
  'mgmd_querier_ip': {},
  'mgmd_querier_port': {},
  'mgmd_querier_timer_info': {},
  'name': 'DEFAULT_VLAN_1',
  'oper_state': 'down',
  'oper_state_reason': 'no_member_port',
  'replication_group': '/rest/v1/system/replication_groups/5ad50a0f-0fd5-47dd-81b8-f40b4e107e6d',
  'type': 'default'}]
```


That's a lot of info…but it is structured!

aruba
a Hewlett Packard Enterprise company

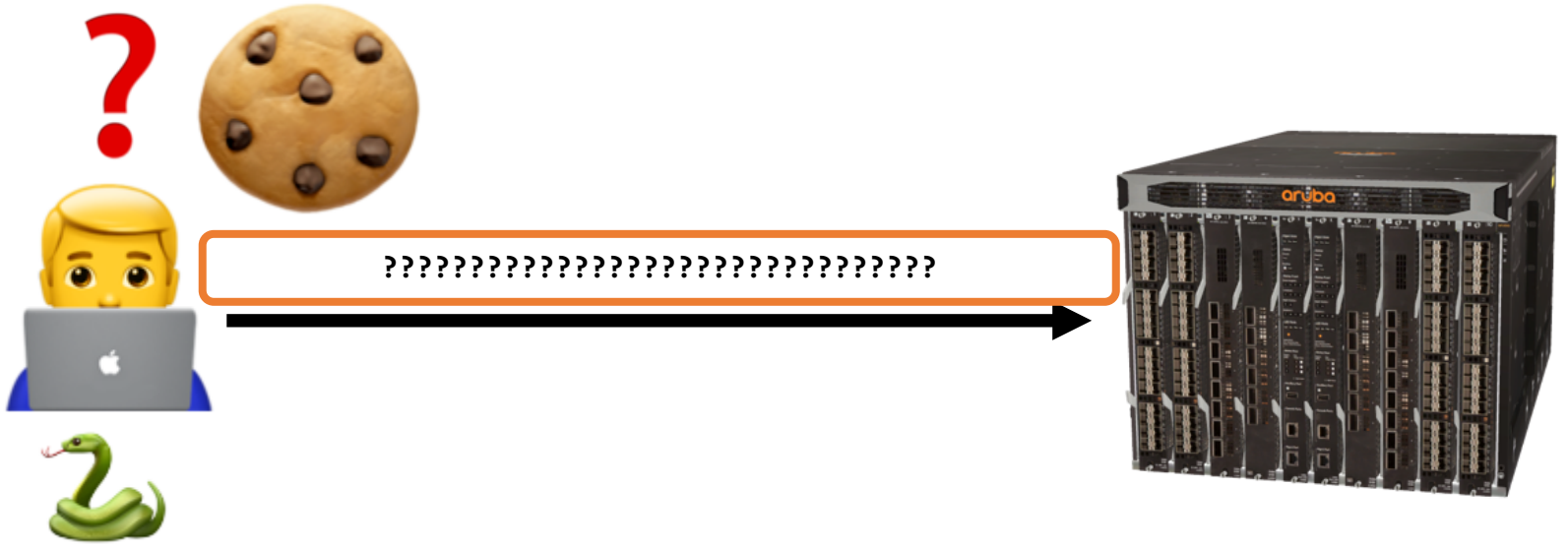
Basic operations with Python

5. Extract required info from **Python dictionary (key-value pair)**
6. Consume (print, if/else logic)
7. Logout



Hang on! How do we know the URLs?

- ArubaOS-Switch – Schema document
- ArubaOS-CX – Swagger UI



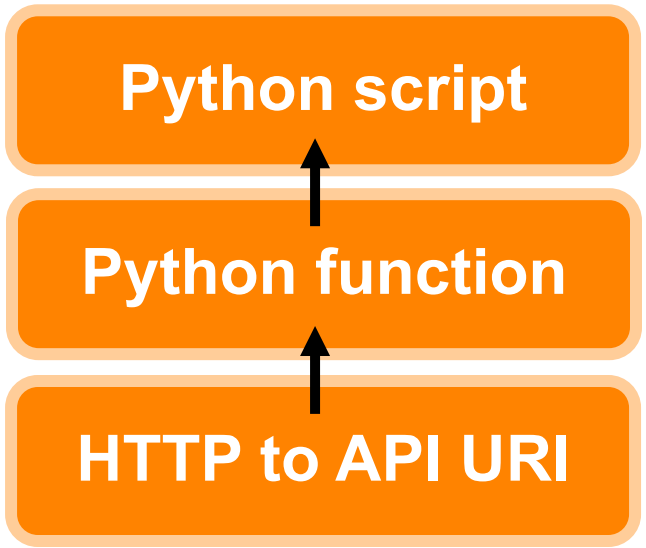
Hang on! How do we know the URLs?

- ArubaOS-Switch – Schema document
- ArubaOS-CX – Swagger UI

DEMO!

Python Functions

- API call operations (login, get x, configure y) lend themselves to python functions.



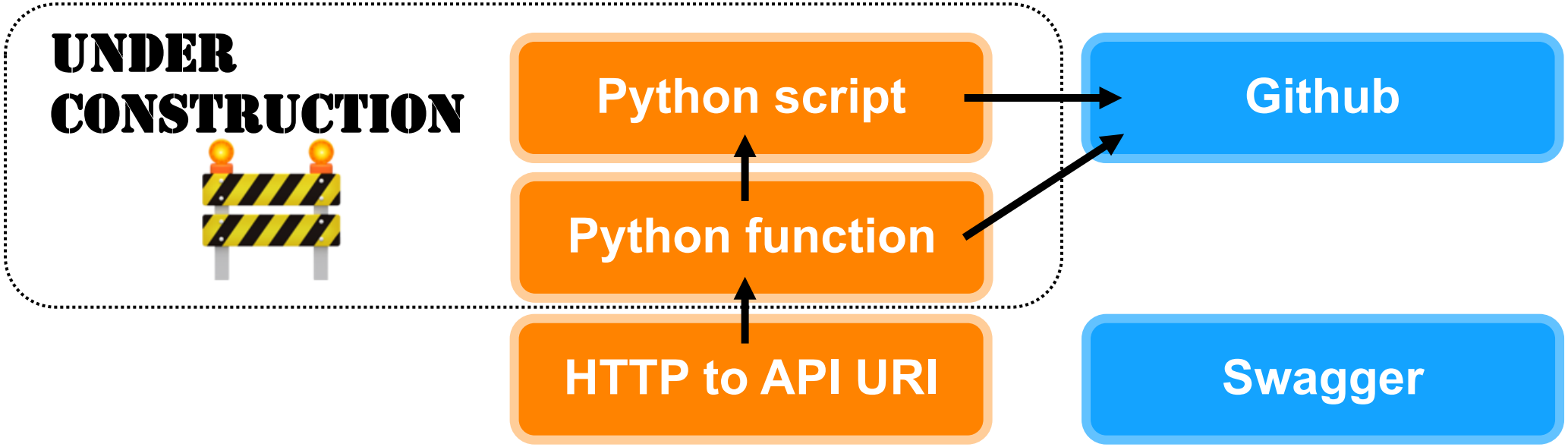
Python Functions

- API call operations (login, get x, configure y) lend themselves to python functions.

Task	Skills required
Build python script from functions	Networking, basic python
Write python function to send call and mine JSON	Intermediate python
Use swagger to get HTTP URI	Networking, Swagger XP

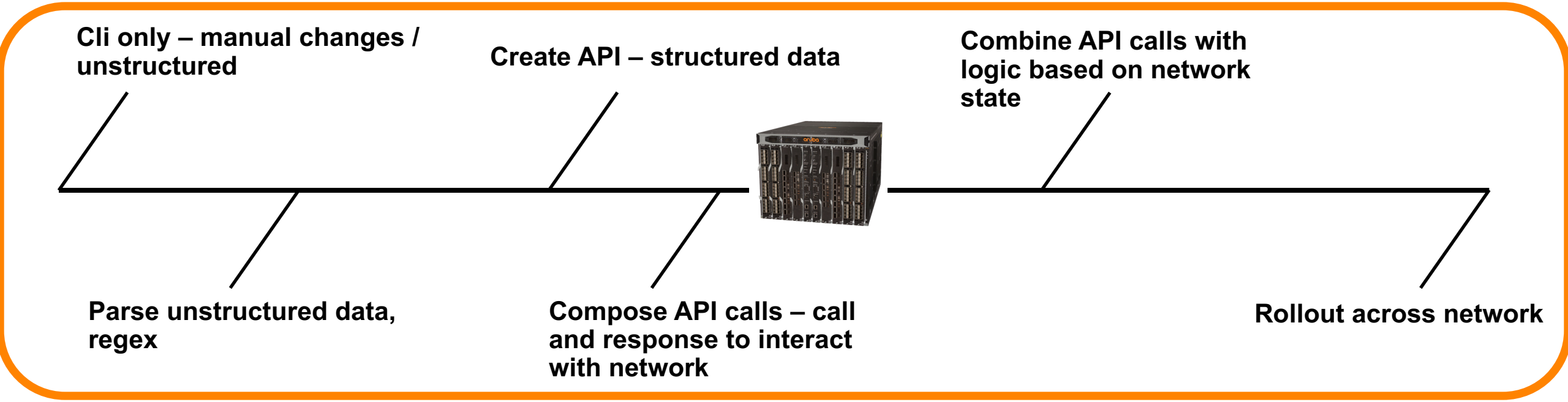
Python Functions

- Software = collaboration (no one writes from scratch)
- Git clone / copy&paste from Github
- Aruba are building an ecosystem - switchingbot



Network Automation Timeline

- Steps towards automating a network



First rule of ~~Fight Club~~ Network Automation:

“Start small”

My experience when mentioning Network Automation

Dreams of fully-orchestrated, self-healing, self-aware networks.

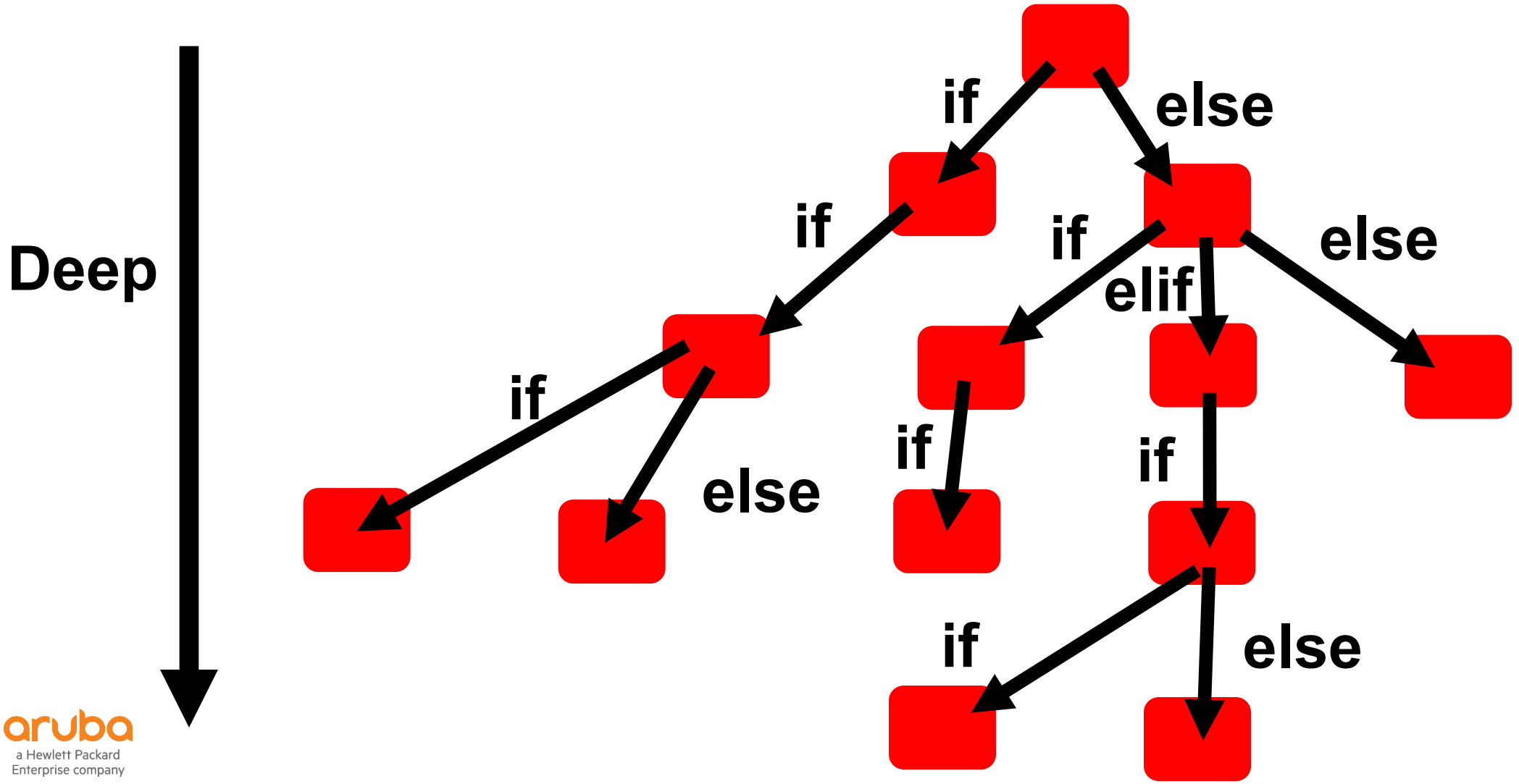
First rule of ~~Fight Club~~ Network Automation:

“Start small”

- Better to go for small, repetitive tasks the take up NetOps time
- Start with info-only.
- Harvest and process data from network.
- Use the existing processes of network engineers. AKA scraps of paper and post-its stuck to screen.

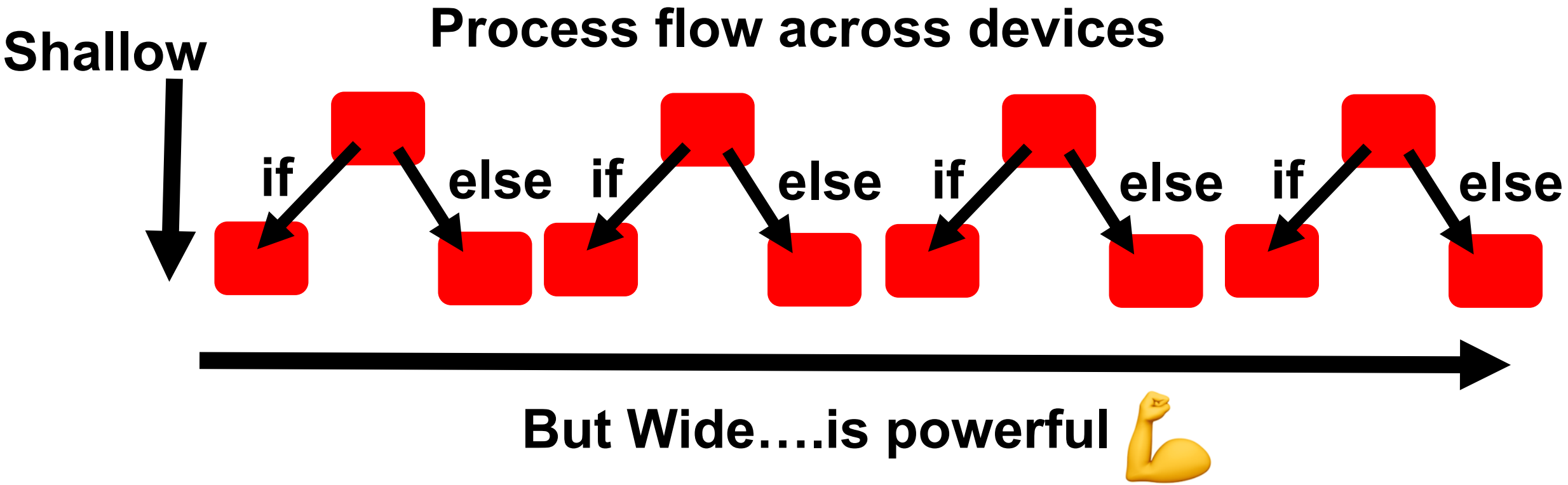
- Create workflows, automate.

Don’t Start Here!



aruba
a Hewlett Packard
Enterprise company

Better!


Shallow
Process flow across devices
if
else
if
else
if
else
if
else
But Wide….is powerful

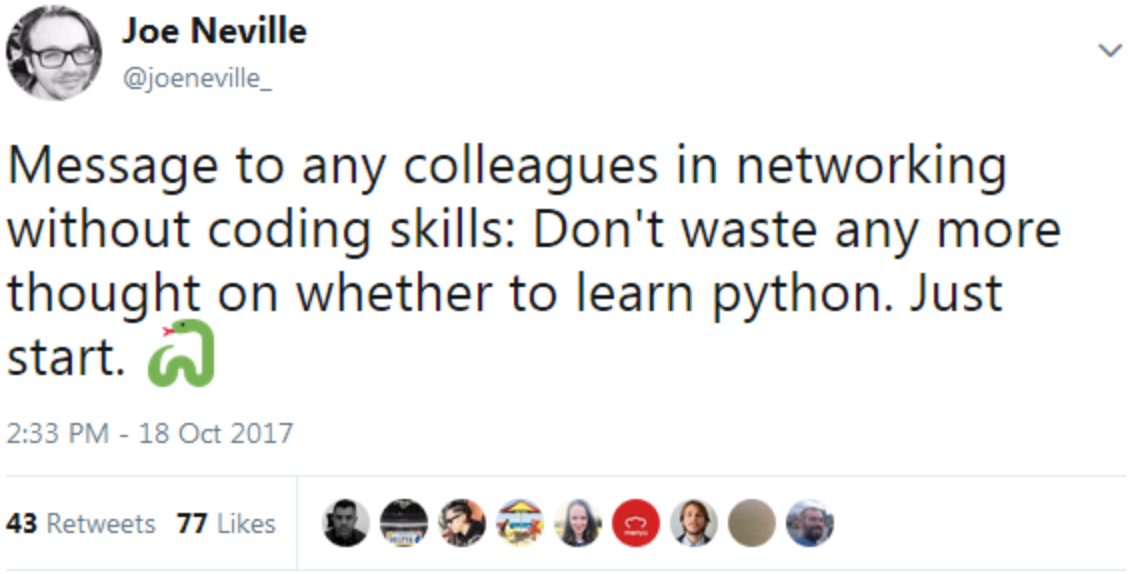
But WHY?????

- Do more with less planned downtime / staff
- Less human-error = less unplanned downtime
- Manage the explosion of networked devices (IoT).
- Save money…of course.

Network automation final thoughts:

- Customer driven
- Don’t panic – start of a long journey
- API is key - this unlocks new potential

Network automation final thoughts:



Thank You!