The bridge to possible

Catalyst Center Automation

Use Cases in Cisco IT

Jonathan Cuthbert – Site Reliability Engineer – Technical Leader
Akanxa Padhi – Senior Software Engineer

BRKCOC-2041

Cisco Webex App

https://ciscolive.ciscoevents.com/ciscolivebot/#BRKCOC-2041

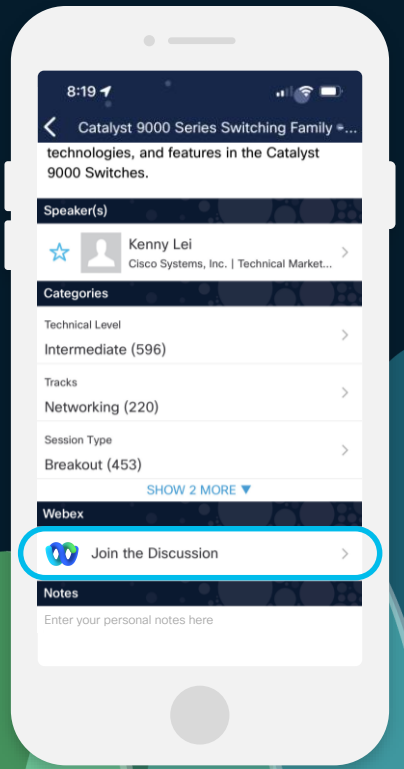
Questions?

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

Webex spaces will be moderated by the speaker until June 7, 2024.



Agenda

- Introduction
- Software Image Management
- Configuration Management
- Conclusion

Agenda

- Introduction
- About Our Network
- Why Controllers
- SWIM
  - Via GUI Workflows
  - (Fully-Automated) Atomic SWIM
- Config Standardization using Day-0 and Day-N Templates
- Config Backup
- Config Drift
- Conclusion

About This Session

- Cisco IT has multiple use-cases for Catalyst Center.
- Two things we do all the time as part of our network.
  - We upgrade devices all of the time.
  - We modify device configuration all of the time.
- There are two primary use-case in this session:
  - Device Software Image Management (SWIM)
  - Device Configuration Management

Session Structure

Jonathan (Software Image Management)

- SWIM
- Atomic SWIM
- Configuration Management Teaser

Akanxa (Config Management)

- Config Standardization
  - C9800 PNP
  - C9800 Day-N
- Config Backup
- Config Drift

Two Parts of the Same Whole

Software Image Management

- ☑ Painful
- ☑ Time-consuming
- ☑ Repetitive
- ☑ (Boring)*

Configuration Management

- ☑ Painful
- ☑ Time-consuming
- ☑ Repetitive
- ☑ (Boring)*

* Tedious
* Not Intellectually Interesting

The Virtues of Boring*

*Boring

- Production Rollouts are rock-solid, rapid, and reliable.
- They happen without unnecessary suspense, mystery, or puzzles.

System Stability Versus System Agility

Stability

Freeze the network as it is now

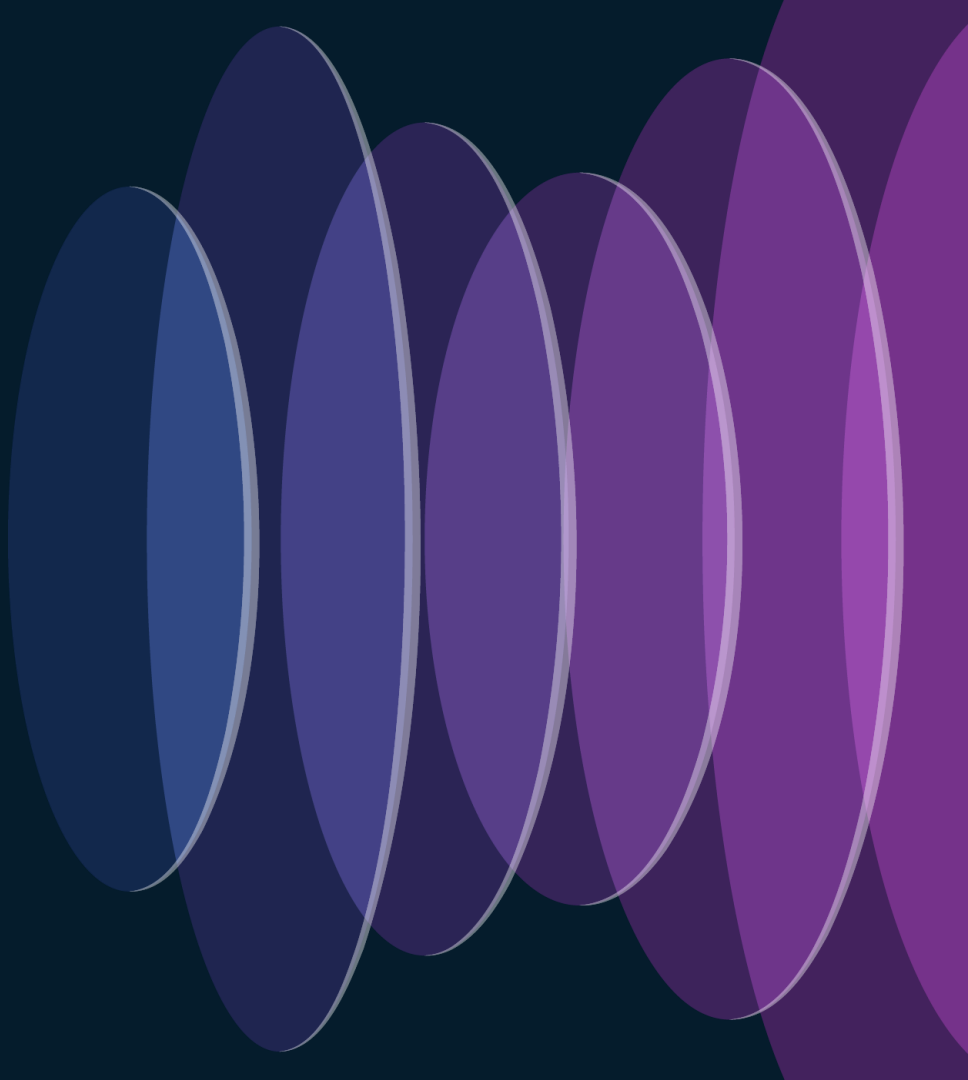
- No additional users or devices
- No incidents or issues
- No new features or capabilities
- Licensing fees regardless 

Agility

Introduce change into the network

- Bugs
- Incidents | Outages
- Growth | Scale | New Capabilities
- Licensing fees regardless 

Our Network



System Scale

- Theaters – 3 
- Regions – 12 
- Countries – 73 
- Cities – 183 
- Buildings – 277 
- Devices – 15,687
  - Routers – 359 
  - Switches – 2,617 
  - WLCs – 289 
  - Access Points – 12,422 

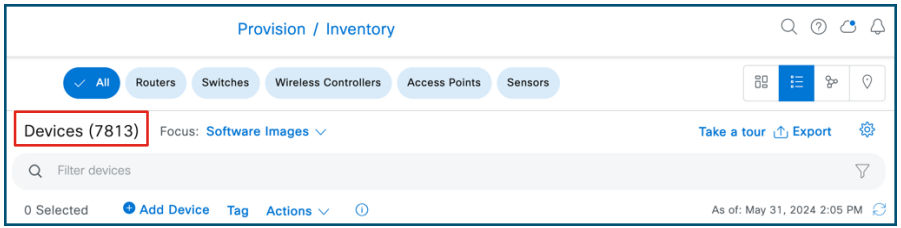
Our Controller Footprint

Production

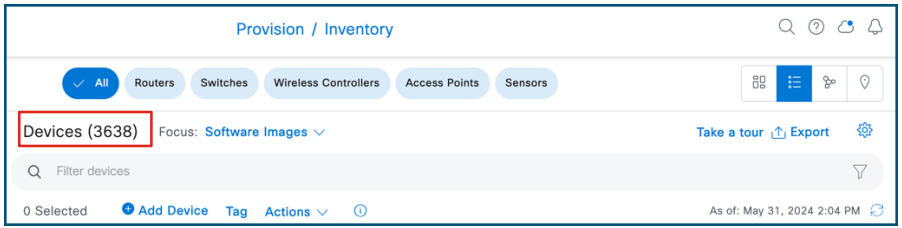
Non-Production

Controller Inventory Snapshot

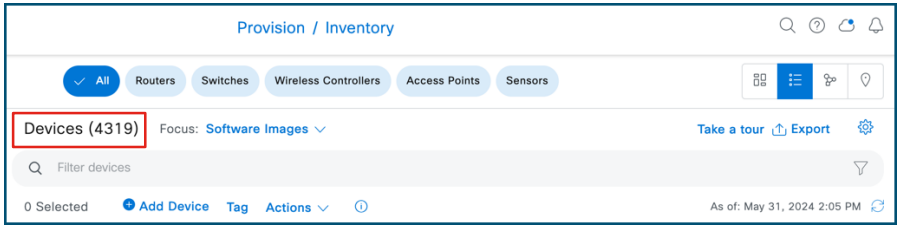
Americas



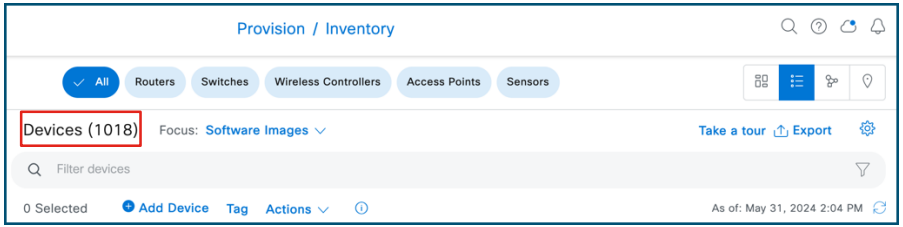
EMEAR



APJC

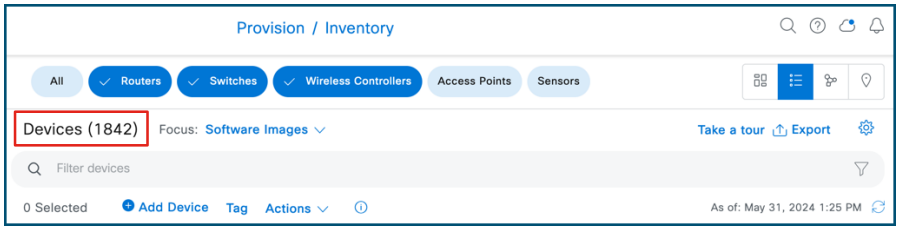


SD-Access

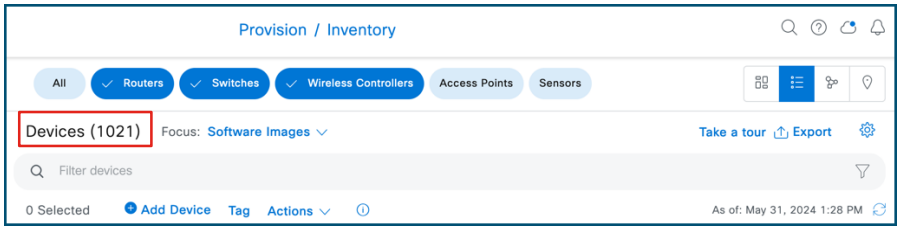


Controller Inventory Snapshot

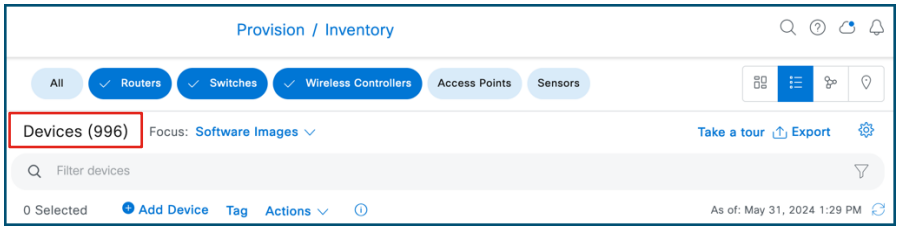
Americas



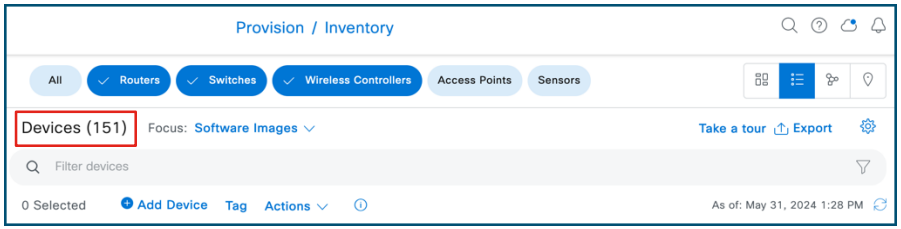
EMEAR



APJC

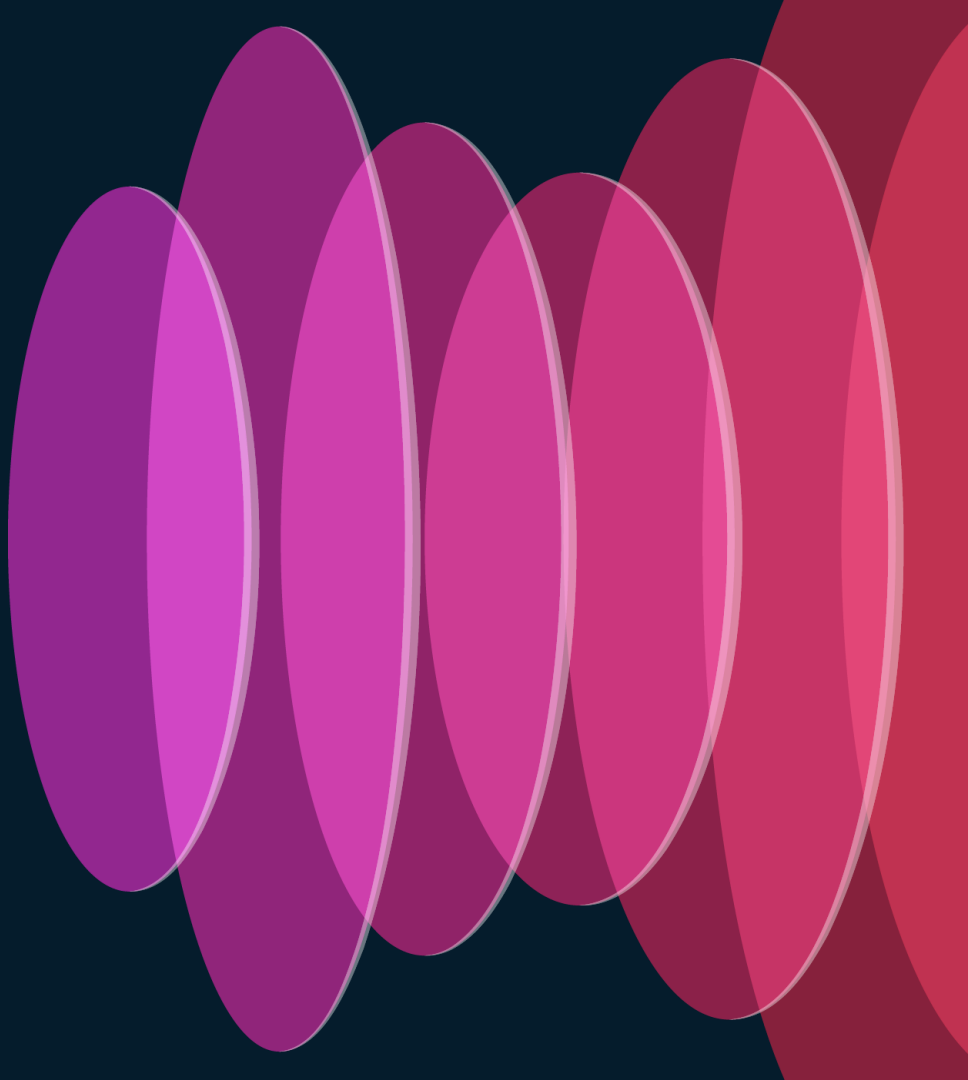


SD-Access



Controller Based Automation

The Why



Controller-Based Automation

What is our Motivation | What Drives Us to These Solutions

- Number of Devices
- Geographic Spread
- Use what you sell

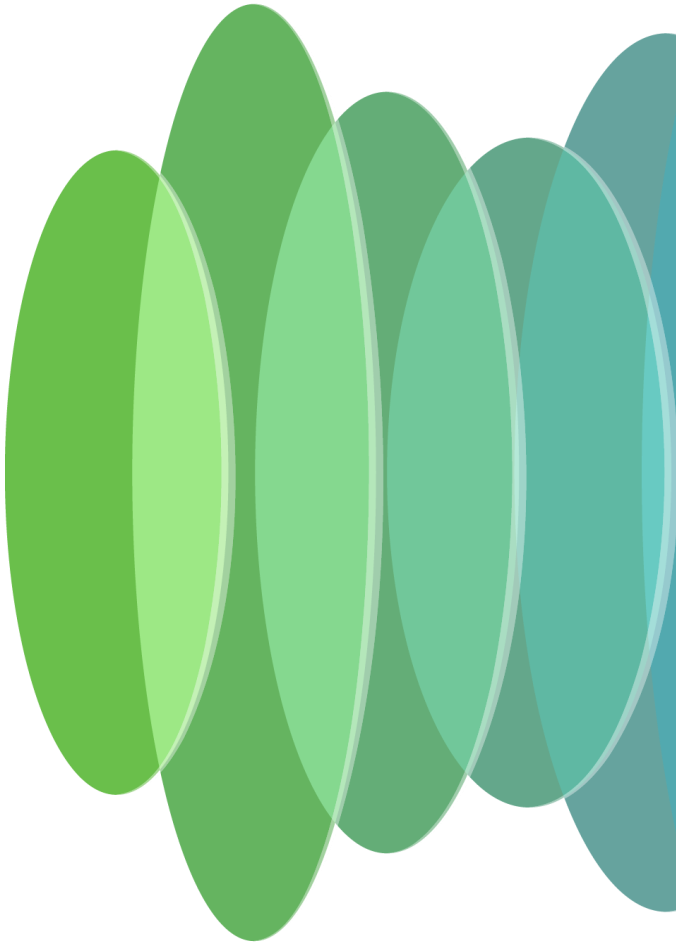
'Use What You Sell'

Understanding This Without The Cognitive Disconnect

- If done manually, we would never finish the upgrades before the next upgrade cycle started.
  - Ongoing ~~Breaks~~
  - Continuous ~~Reprieve~~
  - Never-Ending Job ~~Quarter~~

Start where you are.
Use what you have.
Do what you can.

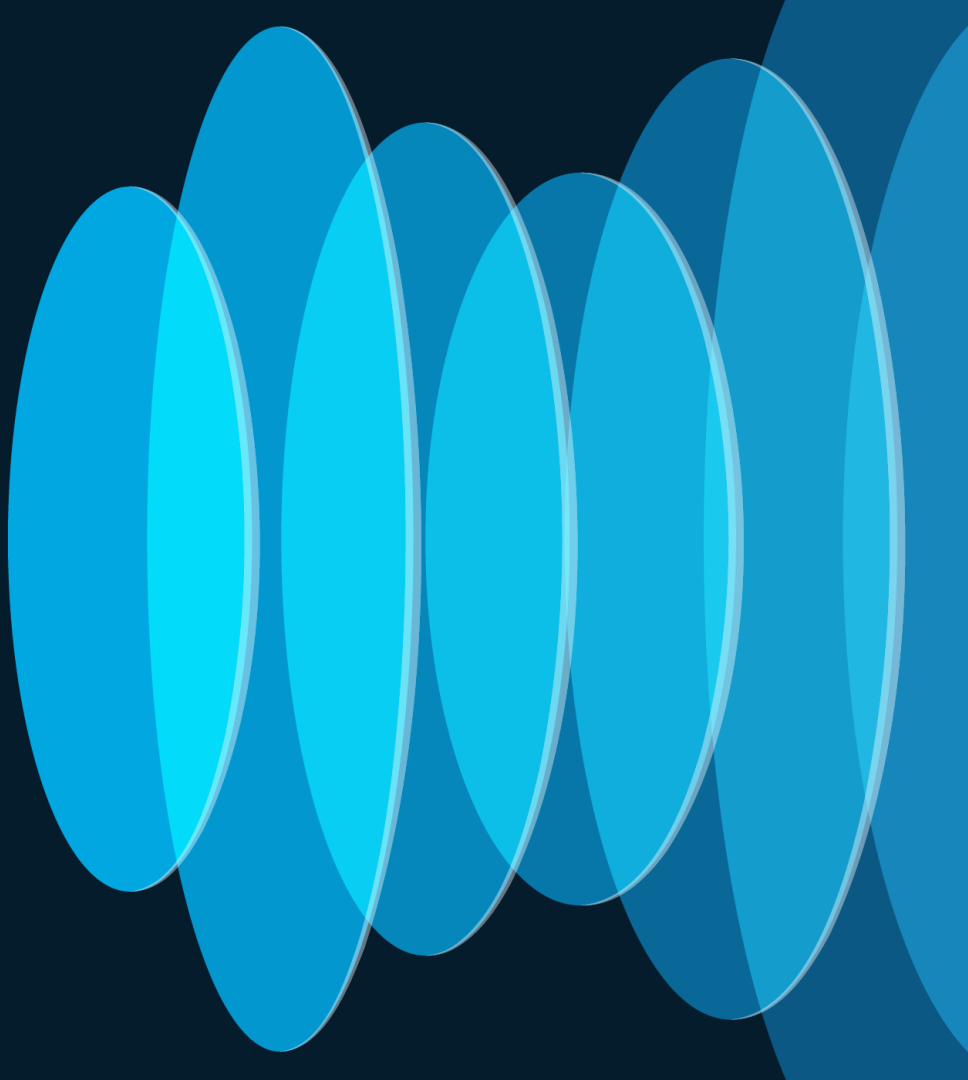
Theodore Roosevelt | Bill Widener | Arthur Ashe



Controller-Based Automation

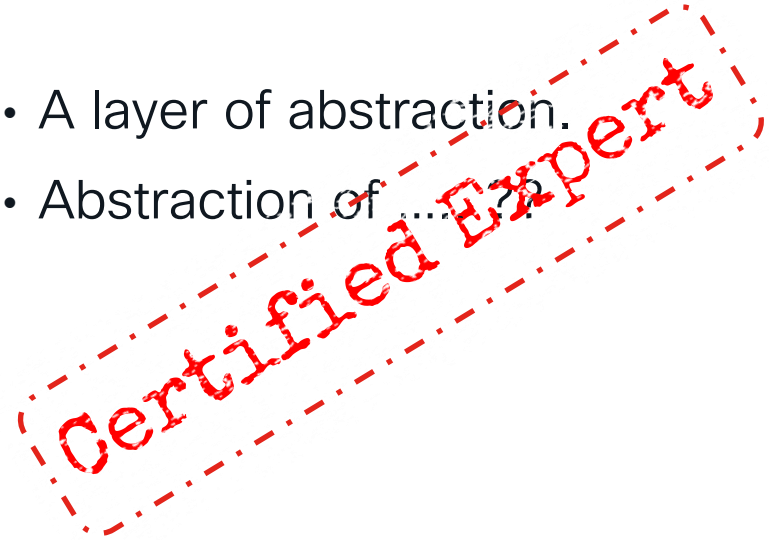
- Learn the capabilities
- Understand how they work.
- Use those capabilities.
- Understand the results.
- Understand the impact on the network.
- Learn how to abstract (automate) the capabilities further.

What is Automation?



Automation – A Definition

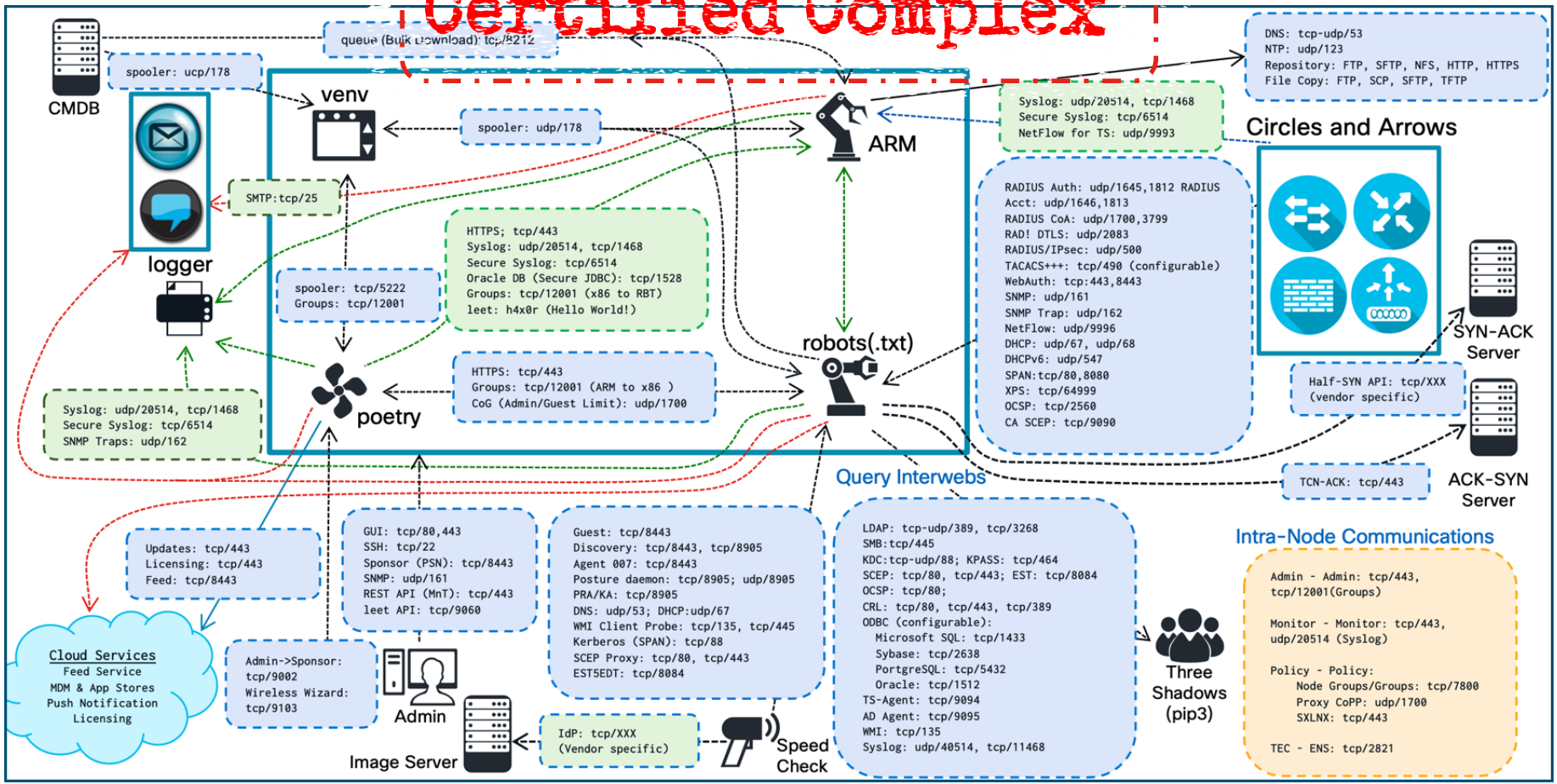
- A layer of abstraction.
- Abstraction of ….??



Automation – A Definition

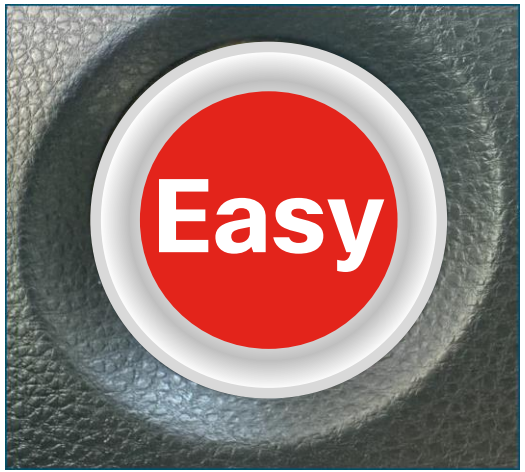
- A layer of abstraction.
- Abstraction of ..... ??
  - Complexity

Certified Complex



Automation – A Definition

- A layer of abstraction that reduces complexity.
- The easy button. ☺



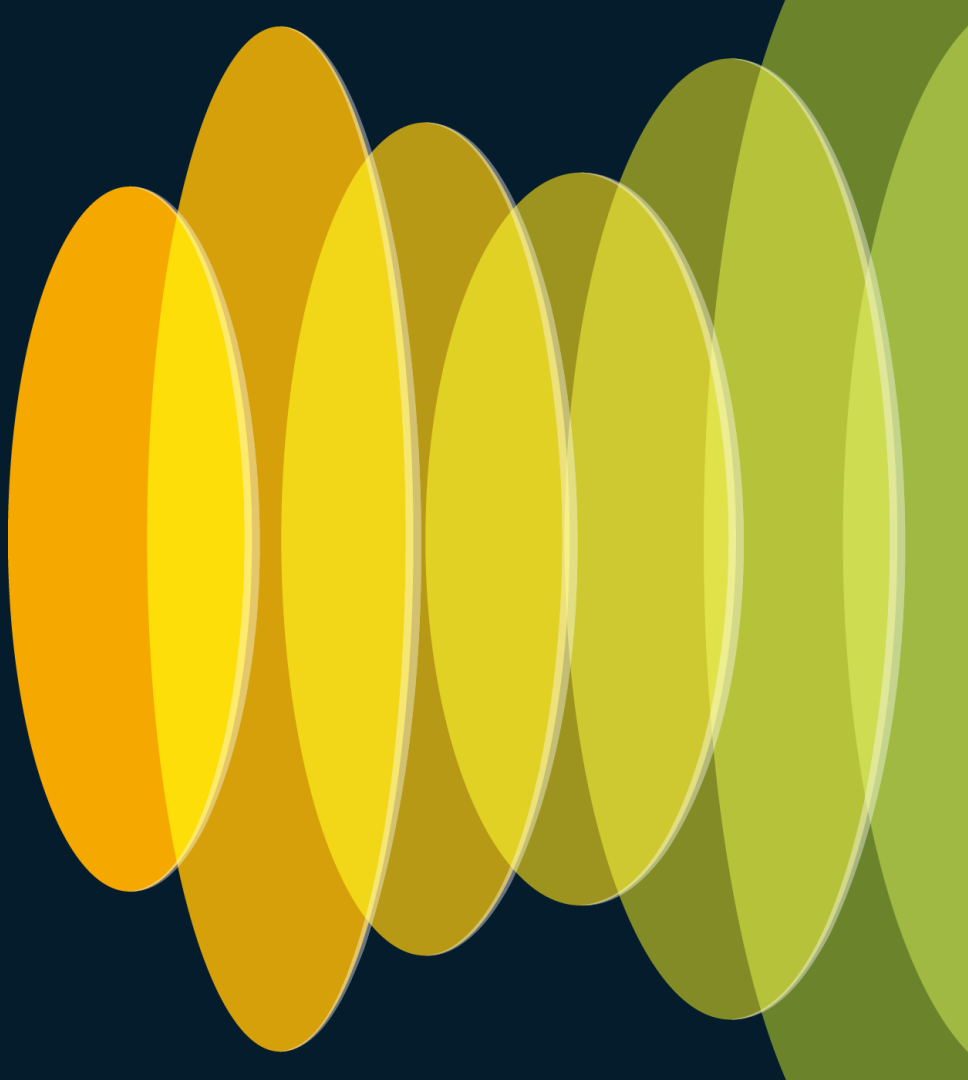
Levels of Abstraction

Getting to the Easy Button 

- Terminal
- Multi-tabbed Terminal
- GUI
- Click, Click, Click in the GUI
- GUI Workflows
- Scripting

Software Image Management

(SWIM)



Catalyst Center SWIM

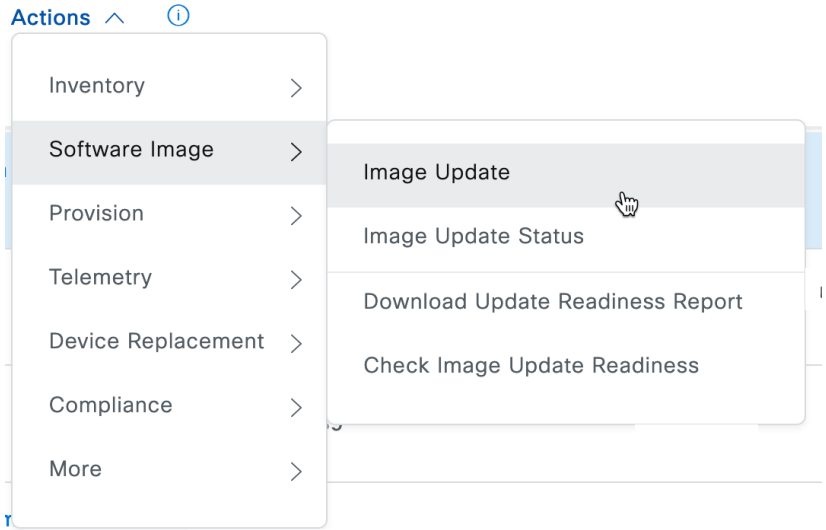
The Why

- Control consistency of software image version across the network
- Reduction of time necessary to perform image upgrades
- Reduction in human errors
- Detailed status and feedback

Catalyst Center SWIM

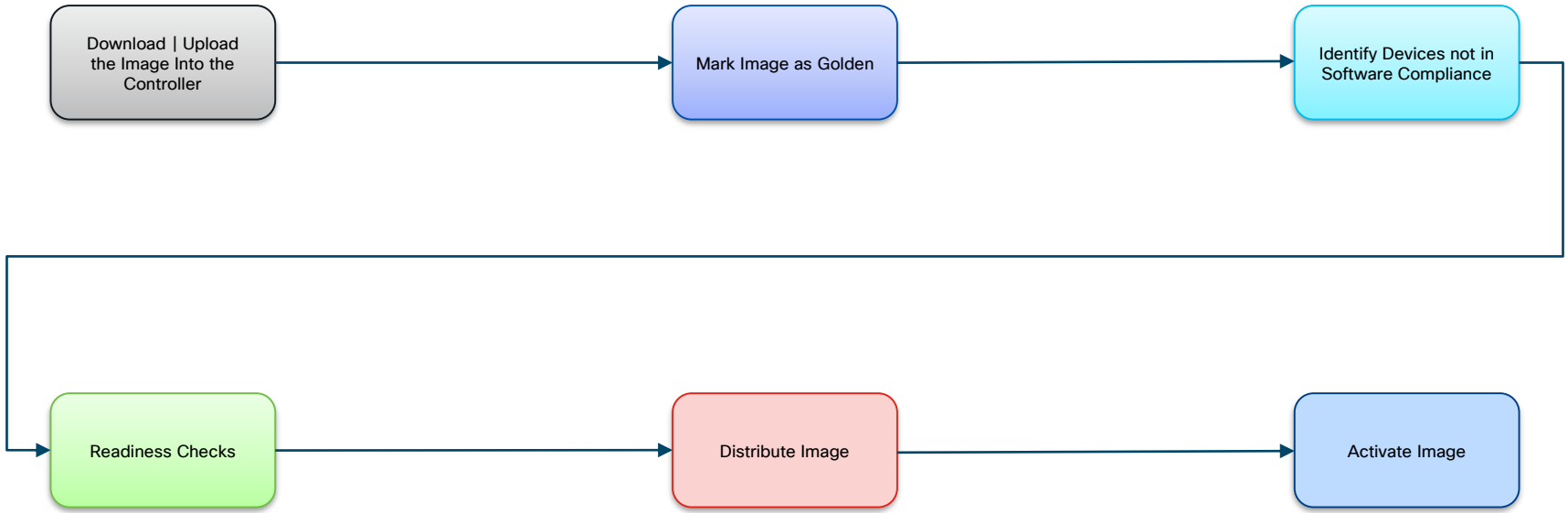
Cisco IT's Approach

- 'Off-the-shelf' solution.
- Using the native workflow in the user interface.



SWIM Via the GUI

High-Level Flow



Catalyst Center SWIM

SWIM Upgrades in the Last Year:

- 17.6.4 **x 3265 upgrades**
- 17.9.3 **x 3265 upgrades**
- 17.12.2 **x 3265 upgrades**

Device Count:

- Routers – 359
- Switches – 2,617
- WLCs – 289
- Total = 3,265

SWIM Metrics

- 3,265 upgrades
- Devices are generally upgrades in batches of around 25.
- Each batch takes around 3 hours.

- ~130 change requests
- 390 hours of upgrading
- 16 days of upgrading
  - If done back-to-back-to-back without stopping

* Tedious

SWIM Metrics + Maths

- ~130 change requests
- 390 hours of upgrading
- 16 days of upgrading
  - If done back-to-back-to-back without stopping

- ~ 450 people in the room
- Each person upgrades 8 devices.
- With precise, intense, and quick actions .... that upgrade still would take an hour.

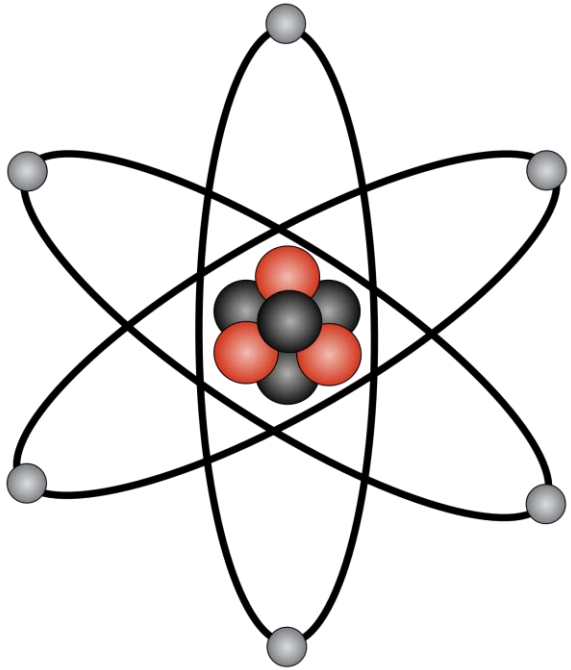
~ 7 minutes per devices
(when done in batches)

* Tedious

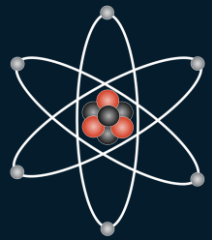
1. Why
2. How
3. Results
4. Lessons Learned
5. Roadmap

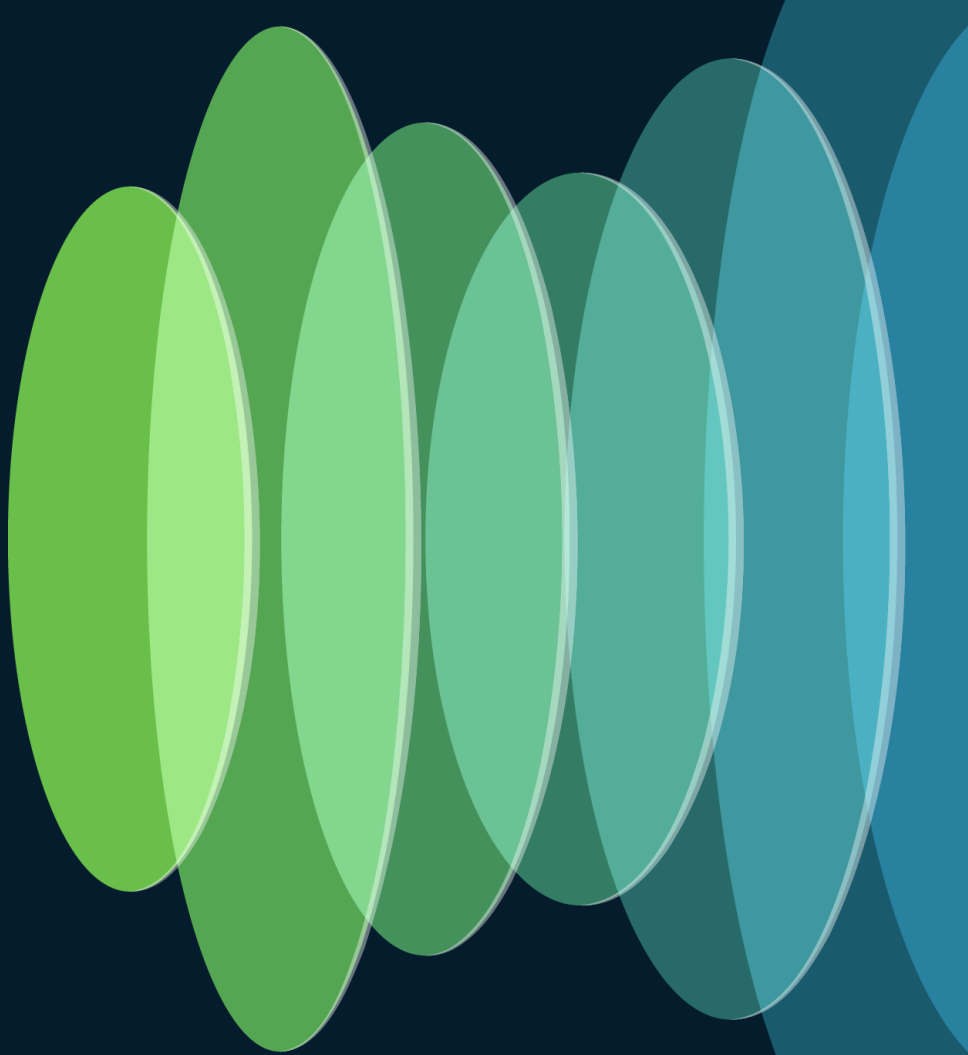
Catalyst Center SWIM

Where are We Going?



Atomic SWIM





Atomic – A Definition

☑ Automatic
☑ Atomic
☑ SWIM

Comes from the programming term *Atomic Operation*

- In concurrent programming, it is an operation that is run completely independently of other processes and operations.

- In terms of Atomic SWIM – each distribution and activation is run completely independently.

- An Alternative Term – Automatic Atomic SWIM

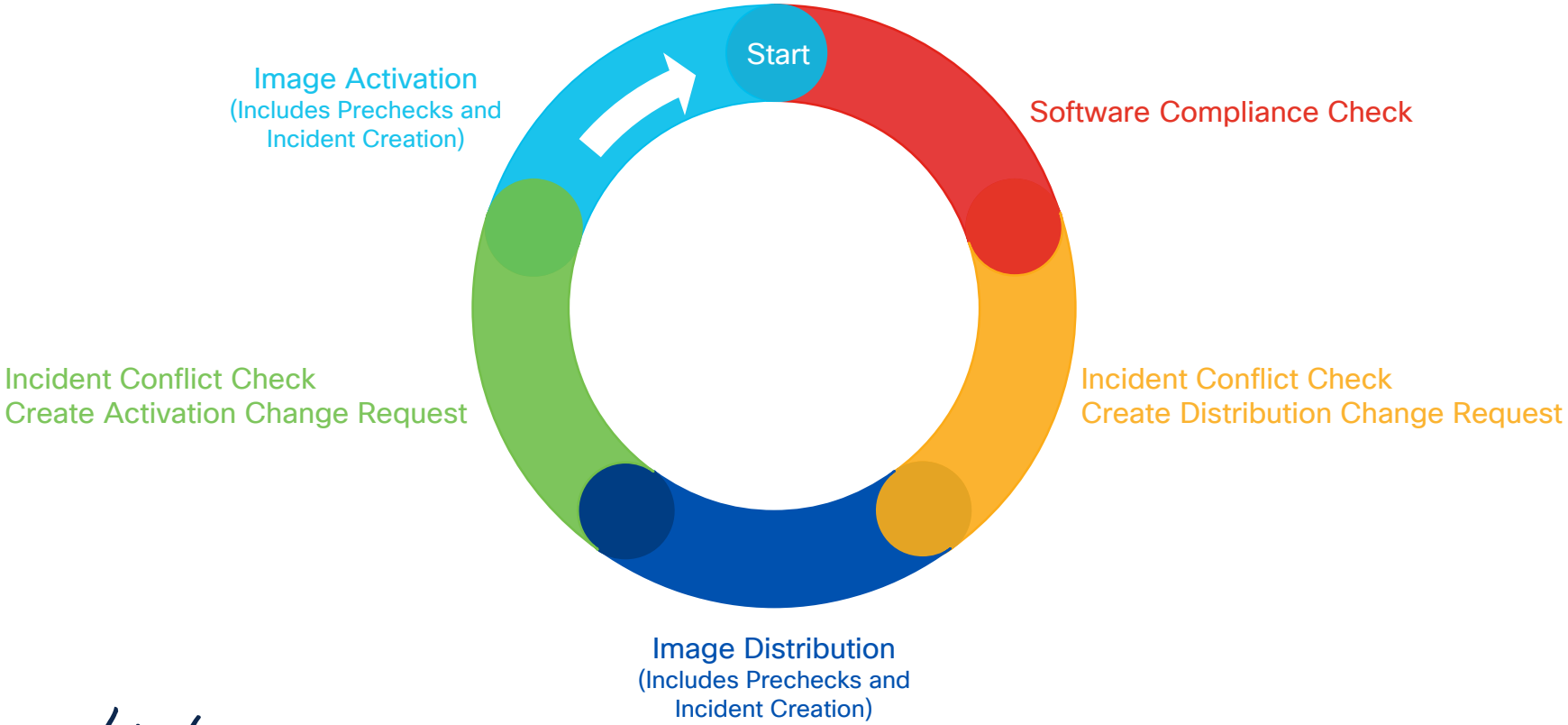
Automatic Atomic SWIM

What if you never had to worry about software upgrades again?

- Imagine the release in operational overhead.
- Consider all the time that could be saved.
- Imagine the freedom to work on other projects and innovations.

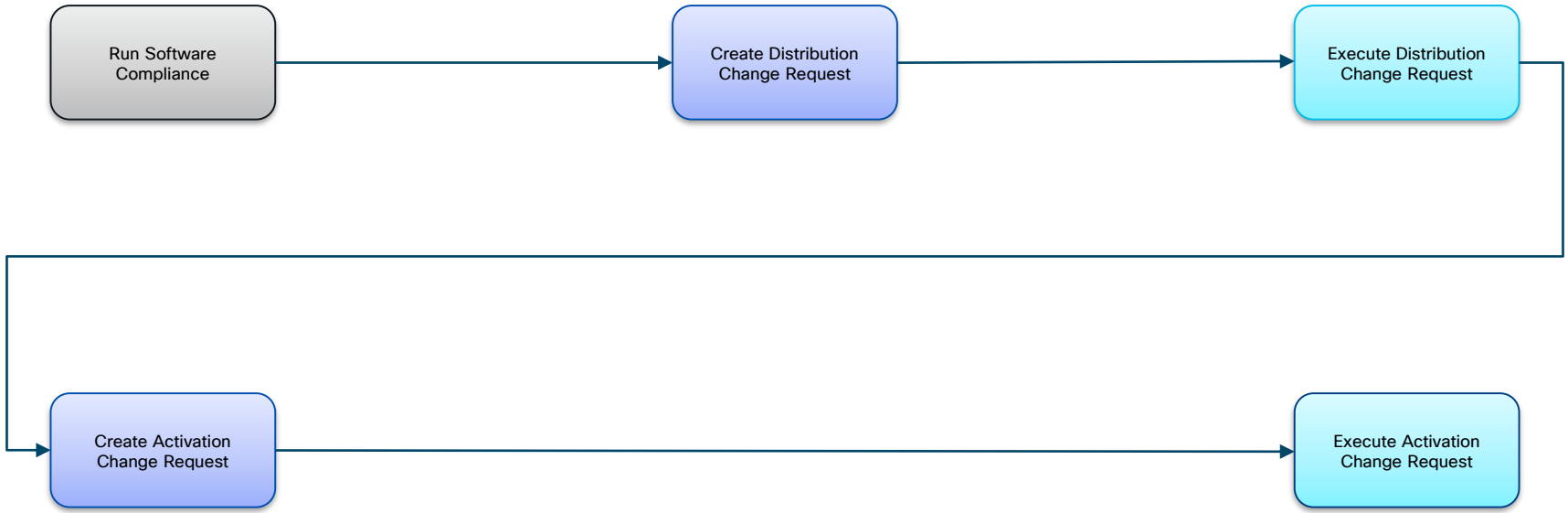
Automatic Atomic SWIM

Fully Orchestrated Workflow



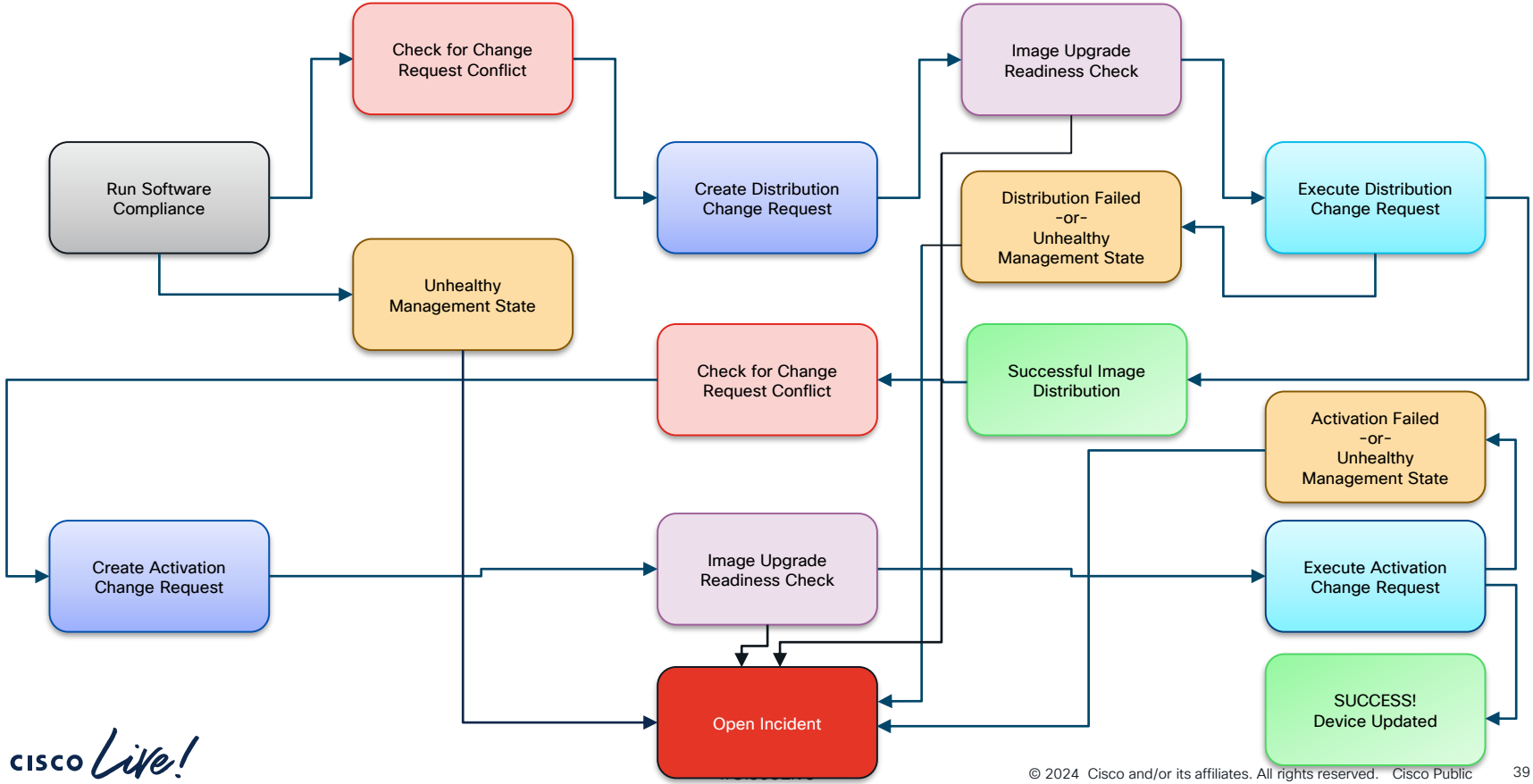
Automatic Atomic SWIM

High-Level Flow

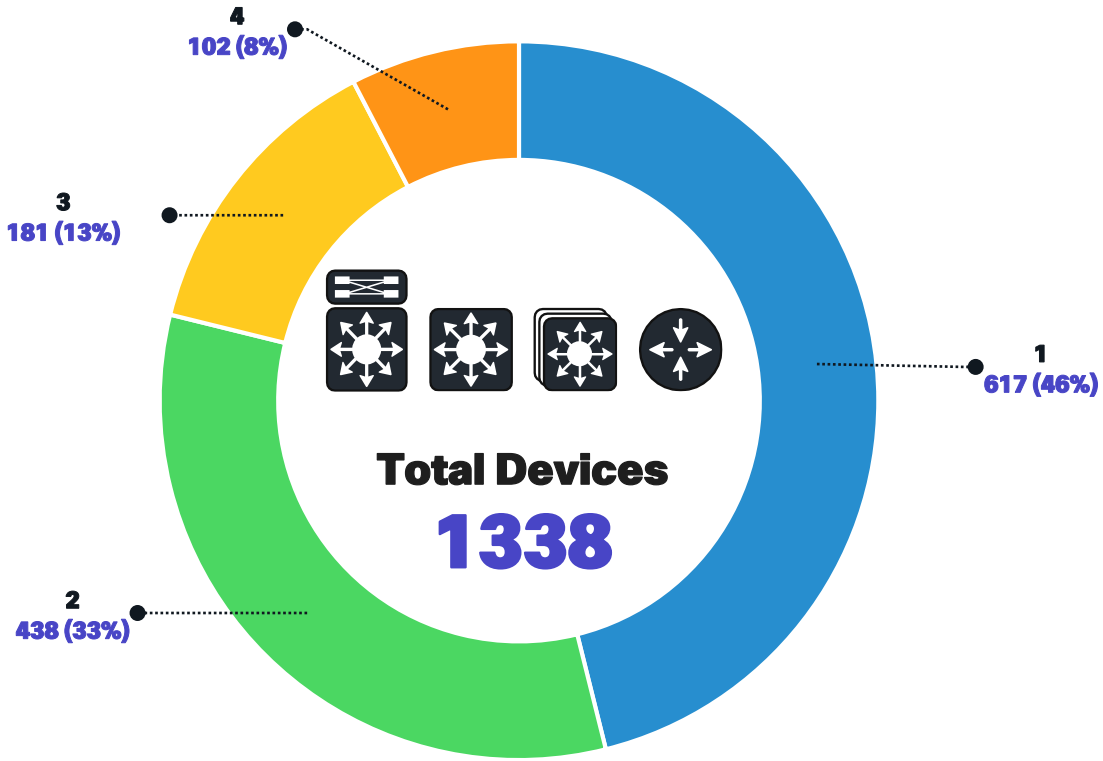


Automatic Atomic SWIM

Low-Level Flow



Atomic SWIM Activations by Device Model



Atomic SWIM Averages

Distributions

- 14 minutes 2 seconds

```
[
    {'time_taken' : '02:50:06'},
    {'time_taken' : '01:59:21'},
    {'time_taken' : '01:47:58'},
    {'time_taken' : '01:47:58'},
    {'time_taken' : '01:47:15'},
    {'time_taken' : '01:46:55'},
    {'time_taken' : '01:45:50'},
    {'time_taken' : '01:45:15'},
    {'time_taken' : '01:44:39'},
    {'time_taken' : '01:43:39'},
    {'time_taken' : '01:43:38'},
    {'time_taken' : '01:43:12'},
    {'time_taken' : '01:43:11'},
    {'time_taken' : '01:43:10'},
    {'time_taken' : '01:42:50'},
    {'time_taken' : '01:41:08'},
    {'time_taken' : '01:39:39'},
    {'time_taken' : '01:39:23'},
    {'time_taken' : '01:39:22'},
```

Activations

- 13 minutes 12 seconds

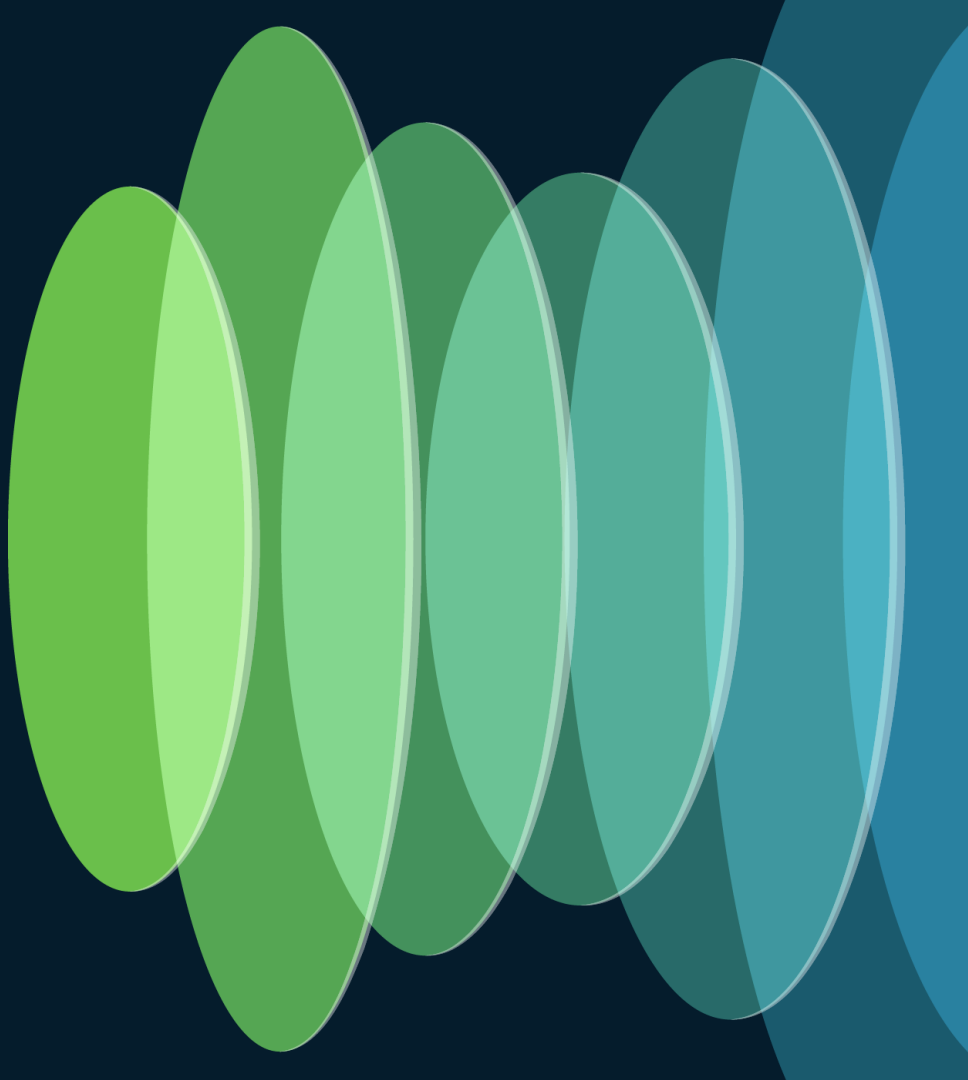
```
[
    {'time_taken' : '01:10:47'},
    {'time_taken' : '01:10:38'},
    {'time_taken' : '01:10:37'},
    {'time_taken' : '01:08:41'},
    {'time_taken' : '01:08:38'},
    {'time_taken' : '01:08:37'},
    {'time_taken' : '01:08:36'},
    {'time_taken' : '01:05:38'},
    {'time_taken' : '00:53:36'},
    {'time_taken' : '00:48:41'},
    {'time_taken' : '00:48:40'},
    {'time_taken' : '00:48:39'},
    {'time_taken' : '00:48:36'},
    {'time_taken' : '00:47:31'},
    {'time_taken' : '00:47:30'},
    {'time_taken' : '00:47:27'},
    {'time_taken' : '00:46:38'},
    {'time_taken' : '00:45:52'},
    {'time_taken' : '00:44:30'},
```

Monster Math

- Actual Code

```
import json

filename = 'atomics_swim_distributions.json'

with open(filename) as f:
    data = json.load(f)

time_taken = []

for element in data:
    for words, time in element.items():
        time_taken.append(time)

counter = 0
total_time = 0

for element in time_taken:
    split = element.split(':')
    hours = int(split[0])
    minutes = int(split[1])
    seconds = int(split[2])
    hours_to_secs = hours * 3600
    minutes_to_secs = minutes * 60
    total_seconds = hours_to_secs + minutes_to_secs + seconds
    total_time = total_time + total_seconds
    counter += 1

average = total_time // counter

MINUTES, SECONDS = divmod(average, 60)
print(f'{MINUTES} min {SECONDS} secs')
```

Spoiler Alert



Using Automation to Address a Security Vulnerability

1. Use Catalyst Center Templates to close the vulnerability

```
ip http active-session-modules none
ip http secure-active-session-modules none
```

2. Use Command Runner via API to check for locally configured users.

```
show running-config | include username
```

3. Use Python with Regular Expression to identify locally configured users.

```
match_device_hostname = re.search("^.+?(?=#)", device)
match_local_usernames = re.findall('.*privilege [0-9][0-9]', device)
```

4. Use Templates to remove locally configured users from the device.

```
! @start-ignore-compliance
#INTERACTIVE
no username admin privilege 15<IQ>[confirm]<R>y
#ENDS_INTERACTIVE
! @end-ignore-compliance
```

Metrics on Security Vulnerability Mitigation

Americas – SDA DNAC

- 151 Devices
- 115 minutes to provision

APJC – DNAC

- 204 Devices
- 67 minutes to provision

Americas – Central DNAC

- 23 Devices
- 48 minutes to provision

Americas – West DNAC

- 15 Devices
- 4 minutes to provision

EMEAR – DNAC

- 34 Devices
- 15 minutes to provision

Totals

- 427 Devices
- 249 minutes to provision

~ 35 seconds per device

What Is Included in that 35 Seconds?

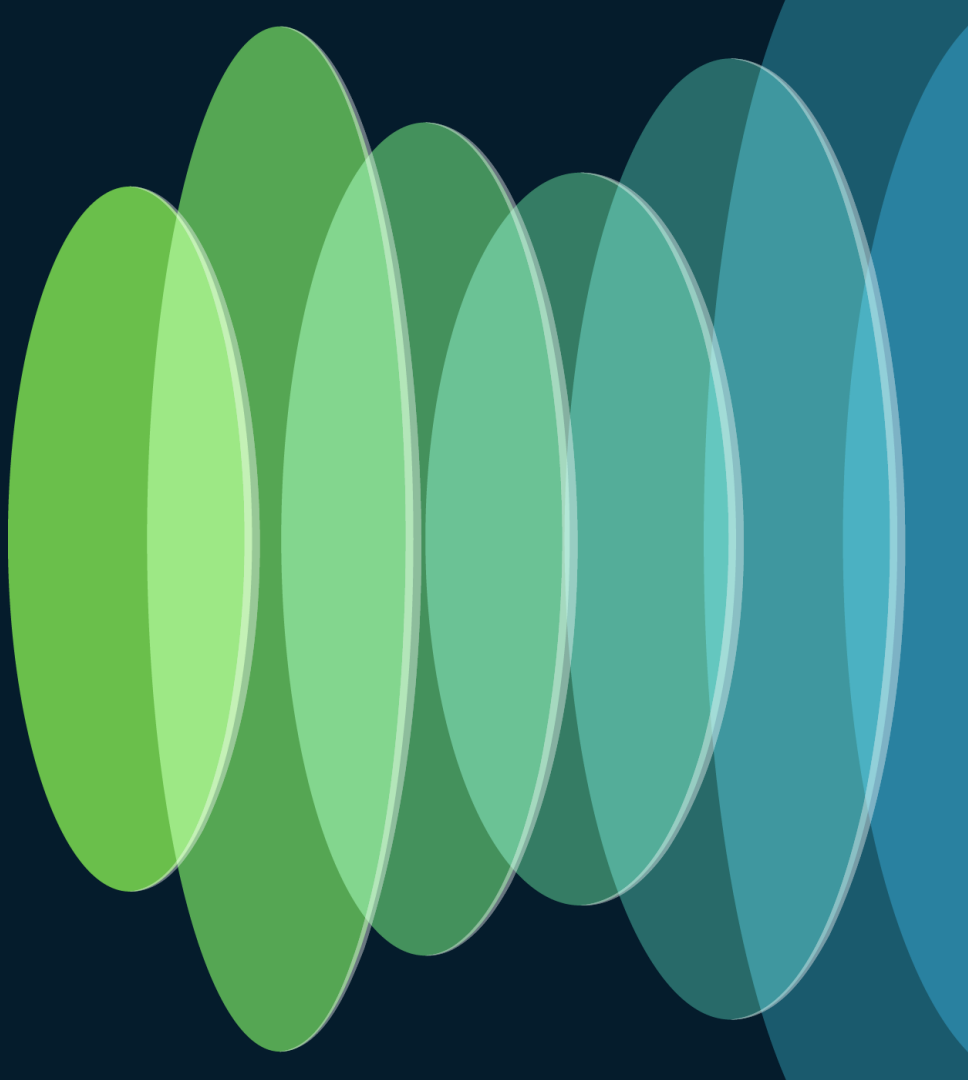
1. Create the Template
2. Initial Commit the Template
3. Update Template Contents
4. Final Commit Template
5. Log Template Contents
6. Provision Template to the Device

Actual Logs


```
24-Oct-2023 04:31:26 PM --> [remove_local_credentials_with_dynamic_templates.py:732] Project: REMOVE_LOCAL_CREDENTIALS already exists on dnac_
24-Oct-2023 04:31:26 PM --> [remove_local_credentials_with_dynamic_templates.py:733] Next Action: Create Templates for Project: REMOVE_LOCAL_CREDENTIALS.

24-Oct-2023 04:31:28 PM --> [remove_local_credentials_with_dynamic_templates.py:880] Step 1 of 6: Create Template
24-Oct-2023 04:31:28 PM --> [remove_local_credentials_with_dynamic_templates.py:881] Creating Template: _24_Oct_2023_04_30_03 on dnac_
24-Oct-2023 04:31:28 PM --> [remove_local_credentials_with_dynamic_templates.py:882] Remove Local Credentials Creation - Time in Task: 1.19 seconds.

24-Oct-2023 04:31:31 PM --> [remove_local_credentials_with_dynamic_templates.py:909] Step 2 of 6: Create Commit
24-Oct-2023 04:31:31 PM --> [remove_local_credentials_with_dynamic_templates.py:910] Submitting Initial Commit of Template: _24_Oct_2023_04_30_03 on dnac_
24-Oct-2023 04:31:31 PM --> [remove_local_credentials_with_dynamic_templates.py:911] Remove Local Credentials Initial Commit - Time in Task: 2.123 seconds.

24-Oct-2023 04:31:35 PM --> [remove_local_credentials_with_dynamic_templates.py:977] Step 3 of 6: Update Template Contents
24-Oct-2023 04:31:35 PM --> [remove_local_credentials_with_dynamic_templates.py:978] Writing template contents for Template: _24_Oct_2023_04_30_03 on dnac_
24-Oct-2023 04:31:35 PM --> [remove_local_credentials_with_dynamic_templates.py:979] Remove Local Credentials Update - Time in Task: 2.178 seconds.

24-Oct-2023 04:31:38 PM --> [remove_local_credentials_with_dynamic_templates.py:1007] Step 4 of 6: Template Final Commit
24-Oct-2023 04:31:38 PM --> [remove_local_credentials_with_dynamic_templates.py:1008] Submitting Final Commit of Template: _24_Oct_2023_04_30_03 on dnac_
24-Oct-2023 04:31:38 PM --> [remove_local_credentials_with_dynamic_templates.py:1009] Remove Local Credentials Final Commit - Time in Task: 2.029 seconds.

24-Oct-2023 04:31:39 PM --> [remove_local_credentials_with_dynamic_templates.py:1022] Step 5 of 6: View Template Contents
24-Oct-2023 04:31:39 PM --> [remove_local_credentials_with_dynamic_templates.py:1023]

! @start-ignore-compliance
#INTERACTIVE
no username admin privilege 15<IQ>[confirm]<R>y
no username cisco privilege 15<IQ>[confirm]<R>y
#ENDS_INTERACTIVE
! @end-ignore-compliance

24-Oct-2023 04:31:39 PM --> [remove_local_credentials_with_dynamic_templates.py:1231] Step 6 of 6: Provision Template
24-Oct-2023 04:31:39 PM --> [remove_local_credentials_with_dynamic_templates.py:1232] Deploy Template Task ID: efd31b69-be7e-4584-895f-8e7973b2055b
24-Oct-2023 04:31:41 PM --> [remove_local_credentials_with_dynamic_templates.py:1241] Template Deployment ID: 47e9781a-a549-4d9c-9e76-e31d0ccaaaa4

24-Oct-2023 04:31:52 PM --> [remove_local_credentials_with_dynamic_templates.py:1207] For Device: | Template Deployment Status: SUCCESS
24-Oct-2023 04:31:52 PM --> [remove_local_credentials_with_dynamic_templates.py:1208] Provisioning success for template _24_Oct_2023_04_30_03.
24-Oct-2023 04:31:52 PM --> [remove_local_credentials_with_dynamic_templates.py:1209] Time in Task: 0 minutes 11 seconds
```

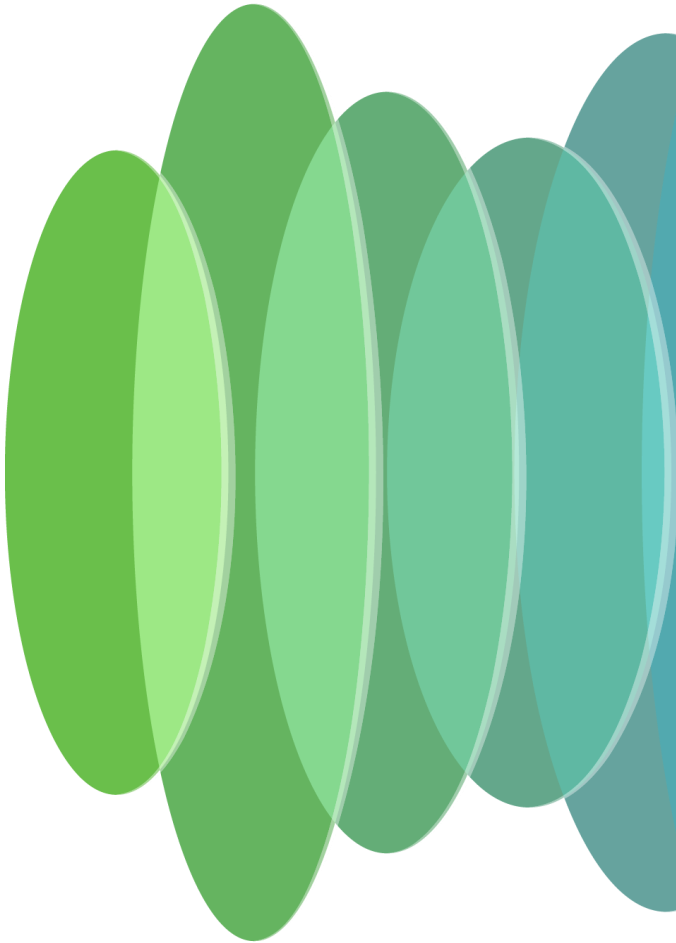
API Calls: 18.5 Seconds
Total Time: 26 Seconds

Key Takeaways



To Err is Human.

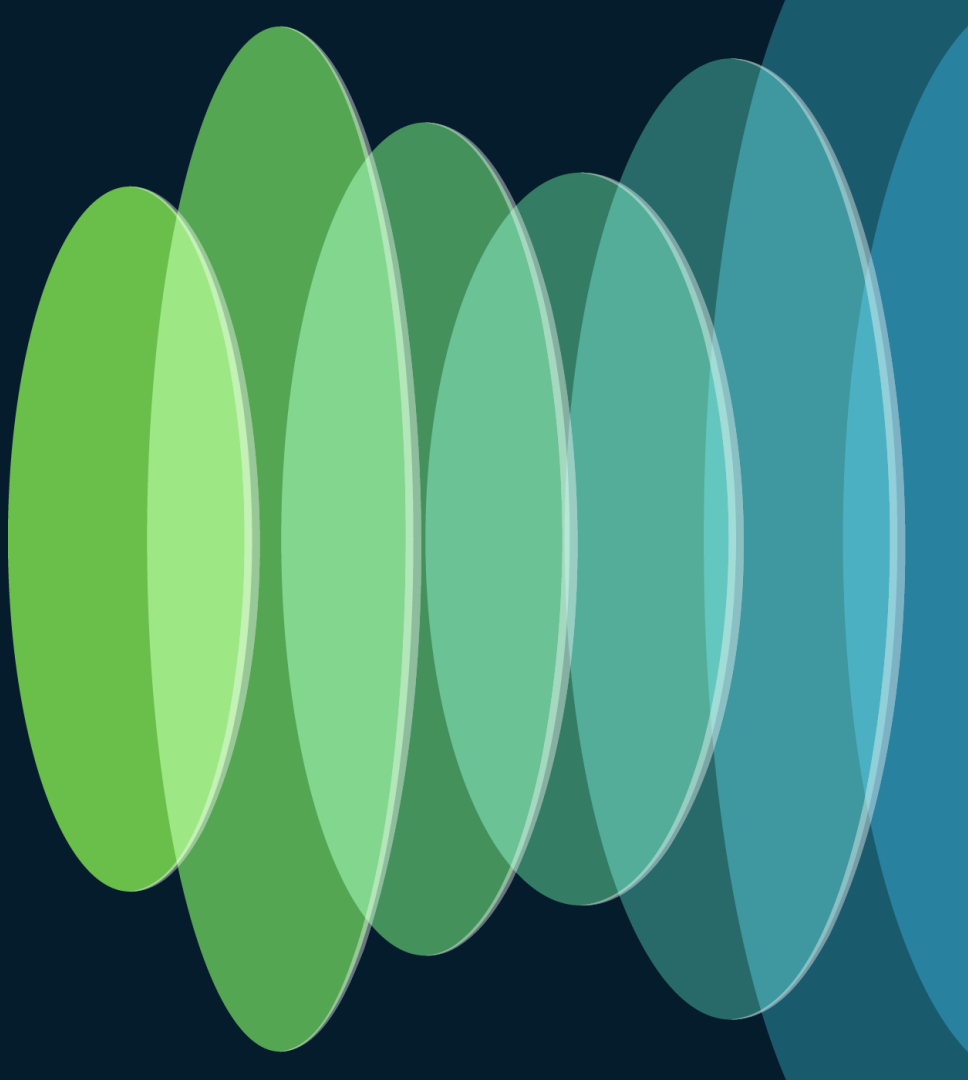
To Err catastrophically at speed and scale is...
(only possible through)
...automation.



Controller-Lead Automation

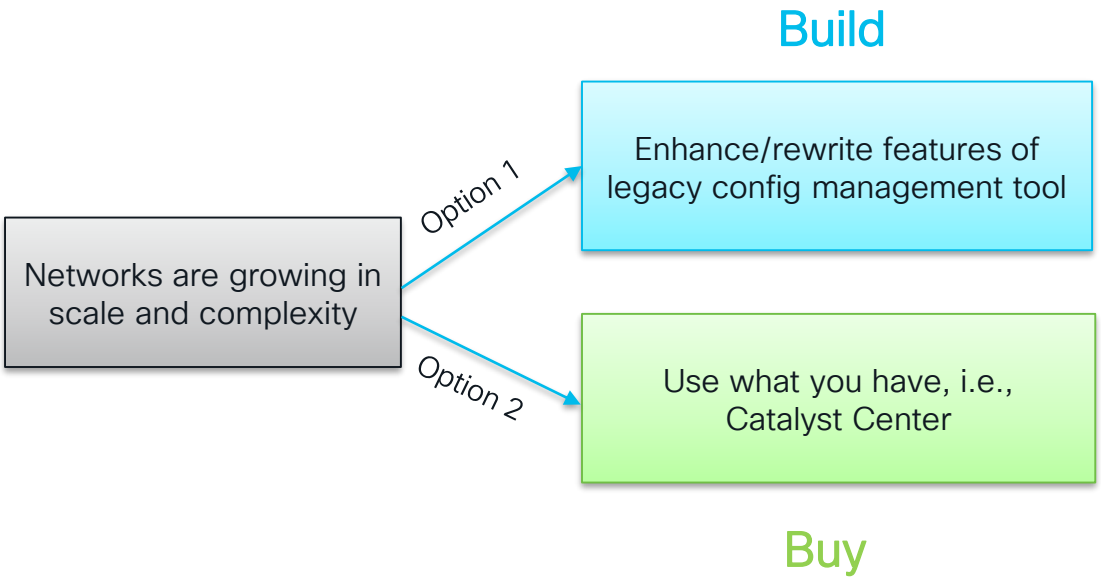
- Learn the capabilities
- Understand how they work.
- Use those capabilities.
- Understand the results.
- Understand the impact on the network.
- Learn how to abstract (automate) the capabilities further.

Configuration Management



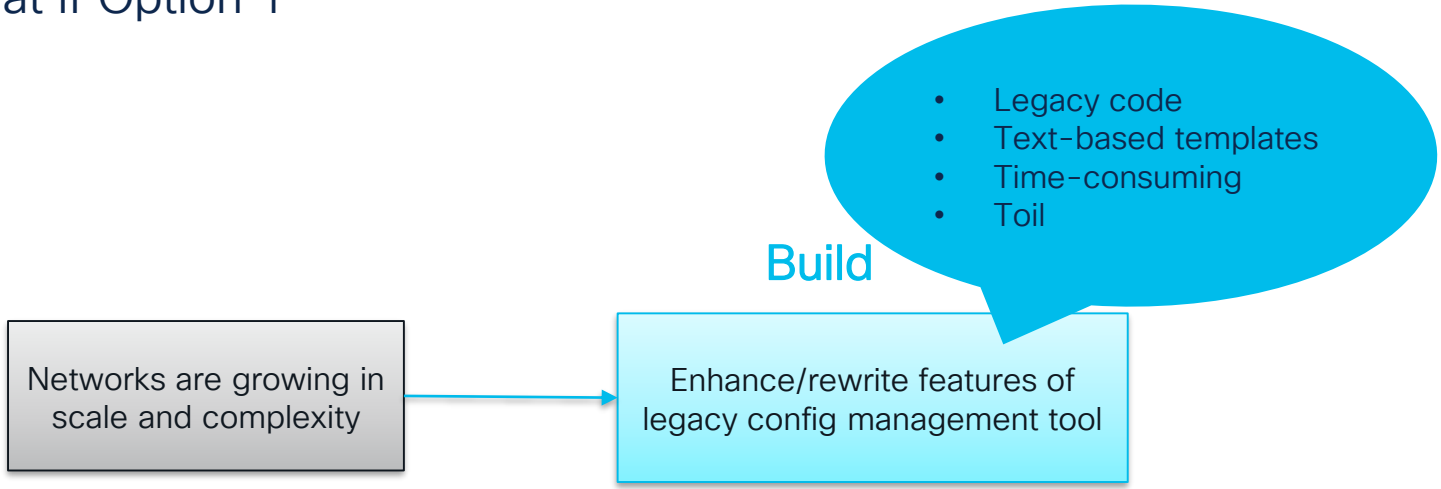
Catalyst Center Configuration Management

The Why



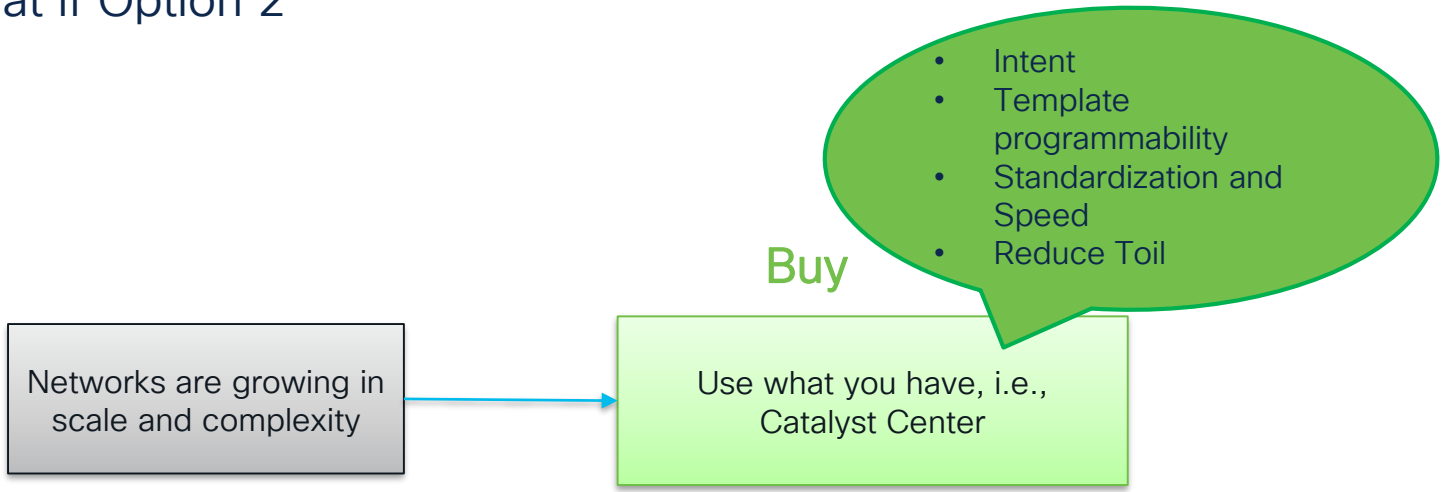
Catalyst Center Configuration Management

What if Option 1



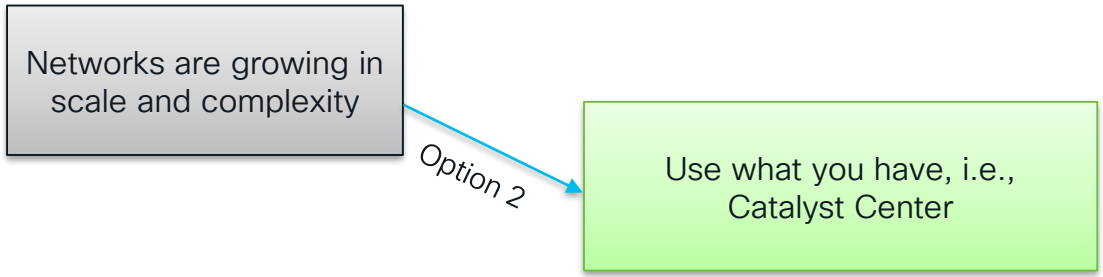
Catalyst Center Configuration management

What if Option 2



Catalyst Center Configuration Management

The Winner



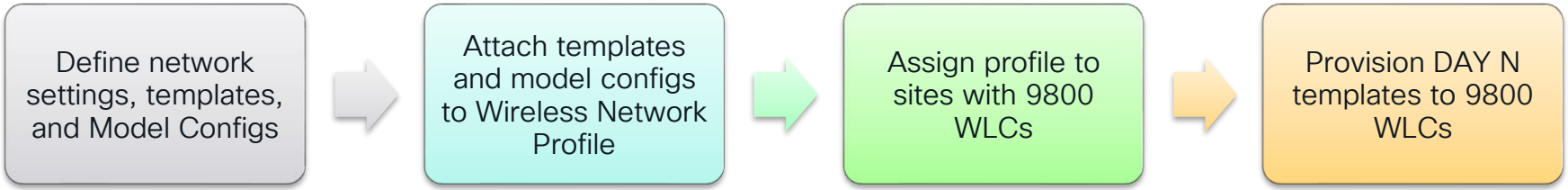
Buy

Configuration standardization with Catalyst Center

9800 Wireless LAN Controller (WLC) – Day 0 config via PnP

Configuration standardization with Catalyst Center

9800 Wireless LAN Controllers – Day N config



We were able to configure 100s of WLCs in a matter of minutes

Catalyst Center Configuration Management

The How

- Automate config standardization at scale
  - Onboard and provision Day 0 configs via Plug and Play
  - Configure Intent, and provision Day N Templates via Network Profile

- Increase trust in our network configuration
  - Config Backup and Config Drift
  - Config Compliance*

*In Progress

Config Compliance with Catalyst Center

- **Config Backups**

  What is configured on the device?

- **Config Drift**

  What configuration changed on the device?

- **Config Compliance***

  Does my device have desired configuration?

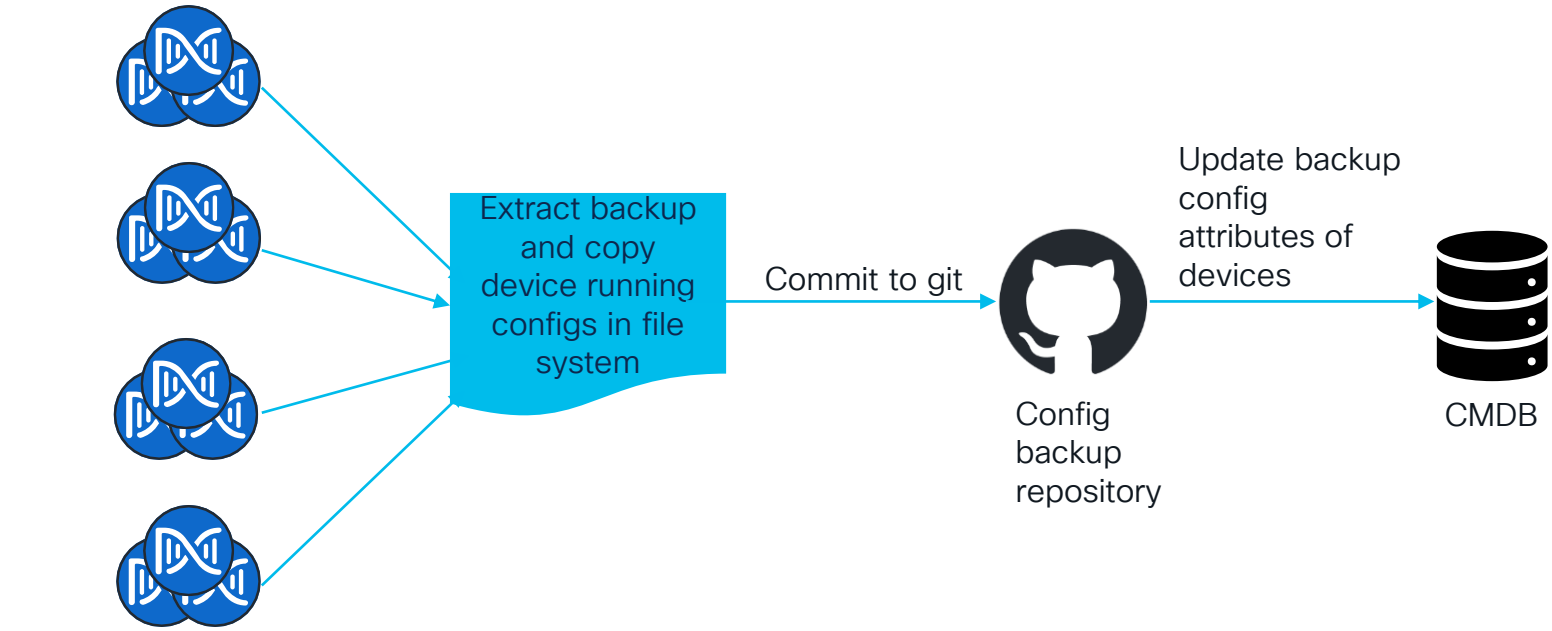
*In Progress

Extending Catalyst Center Configuration Capabilities

- Python-based and API driven extension of Catalyst Center features
- GitOps approach to "democratize" access to device configs and history
- Longer data retention of config change history
- Correlation with CMDB/ITSM*

*CMDB – Configuration Management Database
* ITSM – IT Service Management

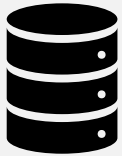
Extending Catalyst Center Config Backup Capability



Extending Catalyst Center Config Backup Capability

From Insights to Actions

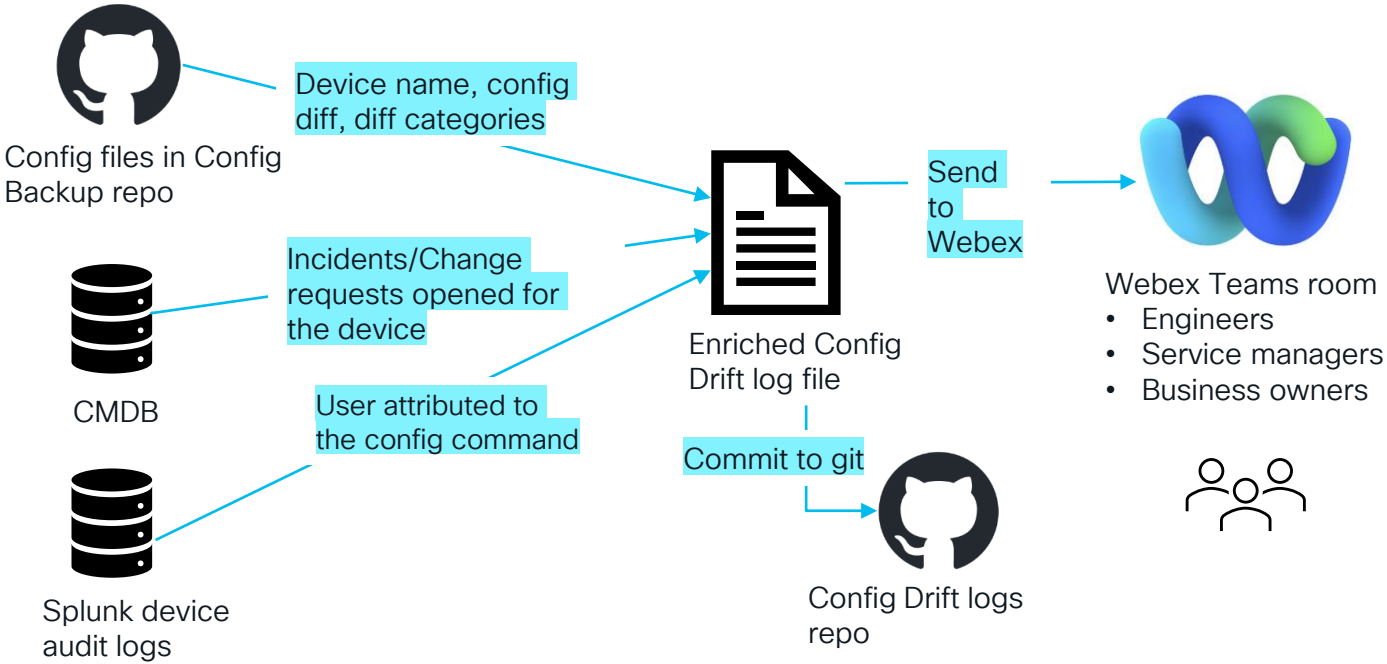
```
ci_name : <switch1>
class : IP Switch
last_backup : 2024-05-31
backup_url : <link_to_config_backup_git_repo>
backup_error_msg*:
service_offering : <workplace_network>
```



If old backup timestamp, open incident with the service offering for resolution

*If success, Backup error message is empty; If failure, has the error encountered while attempting to backup

Extending Catalyst Center Config Drift Capability



Extending Catalyst Center Config Drift Capability

Example – Controller conflict

```
08/28/2022
<wlc1>.cisco.com

+config logging syslog level 6
+config logging syslog level informational
-config logging syslog level 2
-config logging syslog level critical
```

Added by Catalyst Center

Removed by Catalyst Center

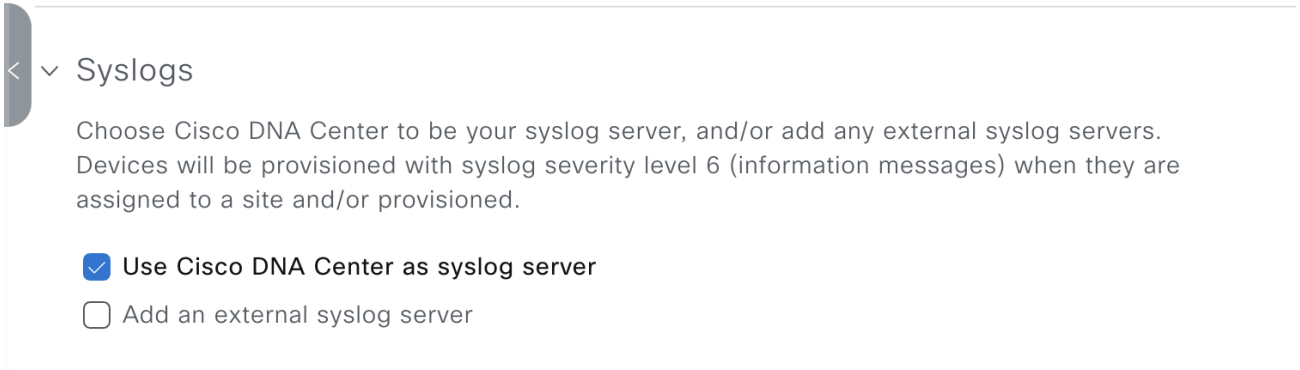
```
08/29/2022
<wlc1>.cisco.com

+config logging syslog level 2
+config logging syslog level critical
-config logging syslog level 6
-config logging syslog level informational
```

Added by legacy config mgmt. tool

Removed by legacy config mgmt. tool

Extending Catalyst Center Config Drift Capability

Example – Controller conflict



Extending Catalyst Center Config Drift Capability

Example – Configuring port security on security cameras

```
<switch1>.cisco.com

Change Category: ['Security camera']

interface <intf_name>
+switchport port-security mac-address sticky <mac_address>
```

```
grep -r "+ switchport port-security mac-address" . | wc -l

23274
```

We automated deployment of MAB for security cameras

Config Compliance with Catalyst Center

- **Config Backups**

  What is configured on the device?

  Is the backup latest?

- **Config Drift**

  What configuration changed on the device?

  Why did the configuration change?

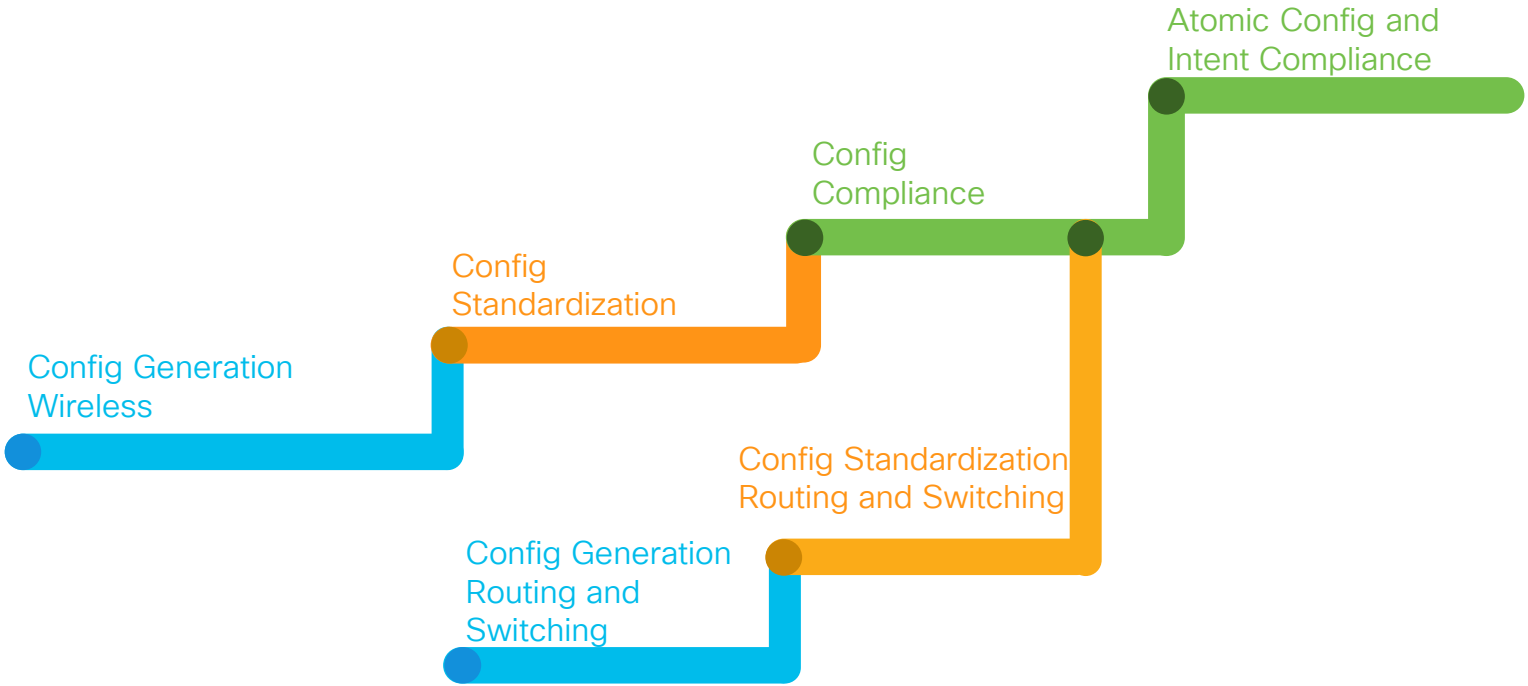
- **Config Compliance***

  Does my device have desired configuration?

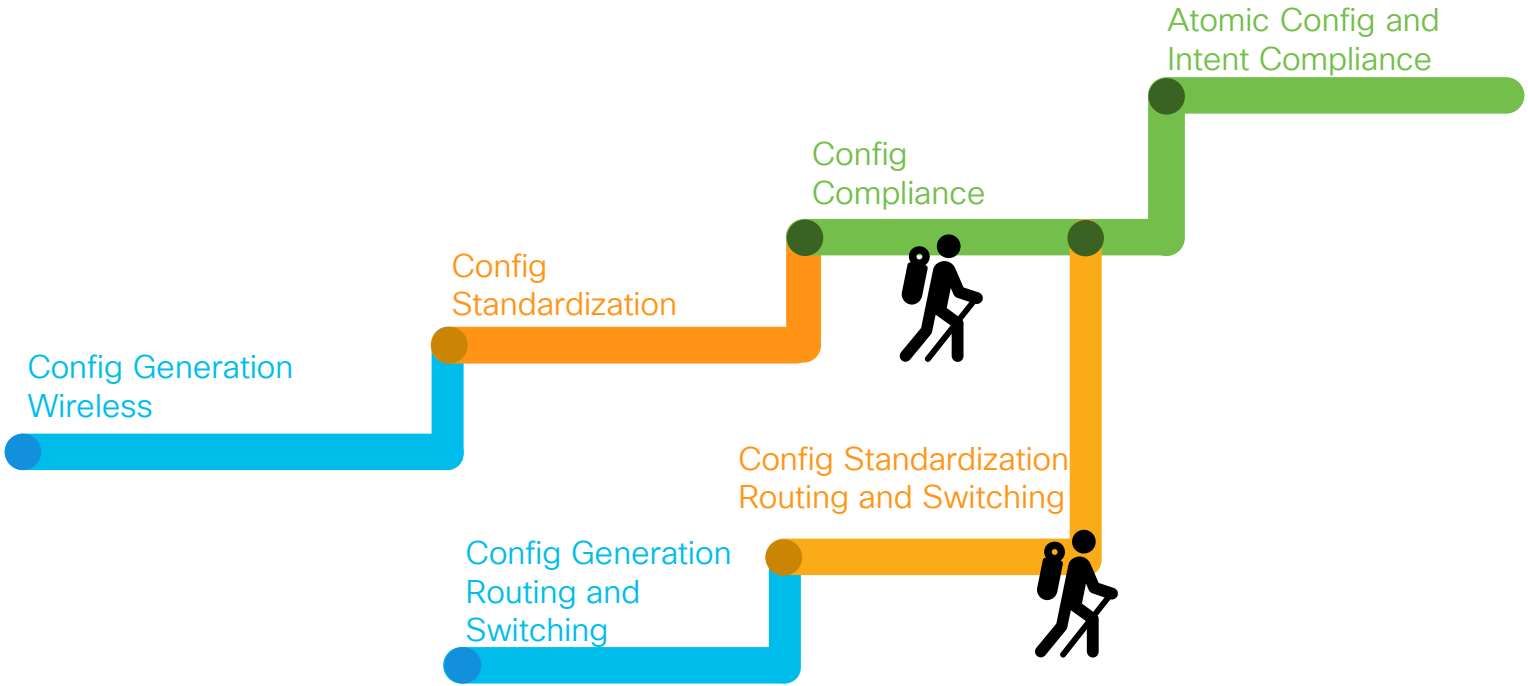
  How do I automate compliance?

*In Progress

Cisco IT Configuration Journey with Catalyst Center



Cisco IT Configuration Journey with Catalyst Center



Configuration Management

Before and After Catalyst Center

Before Catalyst Center

- ☑ Painful

  Manual configuration via device CLI or text-based templates

- ☑ Time-consuming

  Configuration took time; troubleshooting took time

- ☑ Repetitive

  Spray the same configs on boxes all the time

- ☑ (Boring)

After Catalyst Center

- ☑ Simple and Programmable

  Use Velocity or Jinja for seamless configurations

- ☑ Speed and Scale

  100 WLCs configured in less than an hour

- ☑ Event-driven automation*

  Provision configs “only” in the event of an unexpected change

- ☑ Reduced toil

*In Progress

Conclusion

- Cisco IT has a network growing both in scale and in scope.
- Catalyst Center has capabilities to seamlessly automate device software and configuration management at scale.
- This gives us time to work on new value-added initiatives such as:
  - Atomic SWIM
  - Extended Config Archive and Config Drift
- This sets us up for future goals such as :
  - More and frequent device software upgrades using Atomic SWIM
  - Automatic Atomic (Config and Intent) Compliance
  - And … stress-free weekends!

“Keep Calm and Automate Everything”

Continue Your Education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer 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

Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to **win 1 of 5 full conference passes** to Cisco Live 2025.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn **exclusive prizes!**

Complete your surveys in the **Cisco Live mobile app.**

The bridge to possible

Thank you