

>>> network `.toCode()`

NANOOG 81

Automation without Config Deployment

Introduction

Ken Celenza

- Managing Director at Network to Code
- Traditional network engineer by day, coder by night
- Converted full time network Automator in 2016
- 20 years in the industry, primarily supporting enterprises



McKinsey
& Company



*Automation is the art of deploying configurations...
right?*

The Common Approach

Let's Automate the Fun Part!

Let's face it, the configuration is the fun part... why is it the first thing we try to automate?

- Well, I configure ["IPSec tunnels", "firewall rules", "switchports"] that must be where my time is spent.
- Let's automate the configuration deployment of the tedious tasks.
 - Develop *conf_ipsec_tunnel.py*, *deploy_fw_rule.yml*, etc.
- The configuration required only requires a few variables.

Why isn't Anyone Using my Automation?

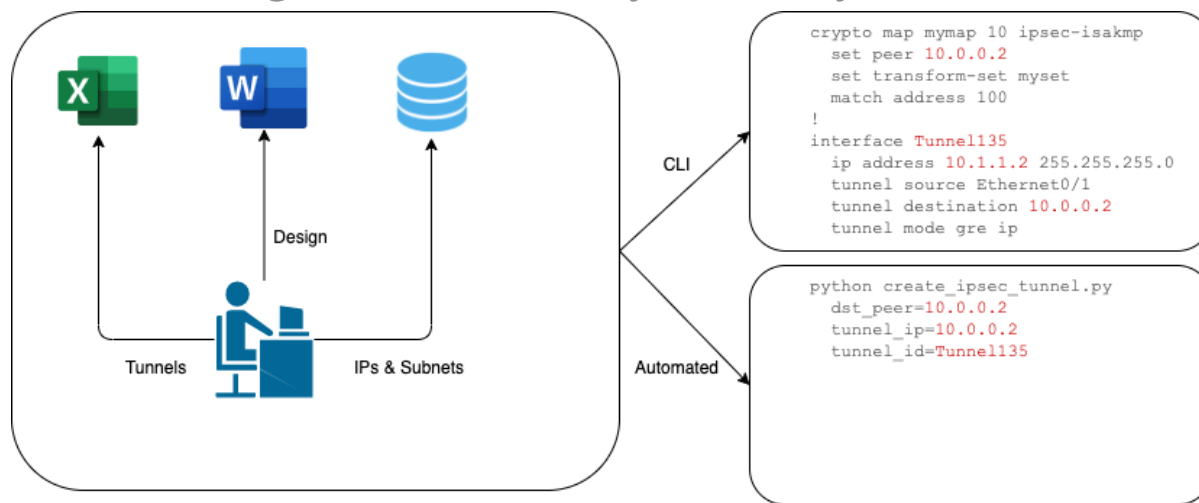
- Change window was an hour, and now “I am done a few minutes early.”
- “My change window is so short, I need to make sure everything works in time”
- “If I’m going to make the change, I want to know what configuration is going to be deployed”
- “Automation can’t be run unattended, I still need to verify everything myself”

What are the Issues?

Source of Truth

Time is spent curating the “correct data” and configuration.

- Data is kept transactionally, and not via the SoT.
- Results in re-doing the same analysis every time there is a change.

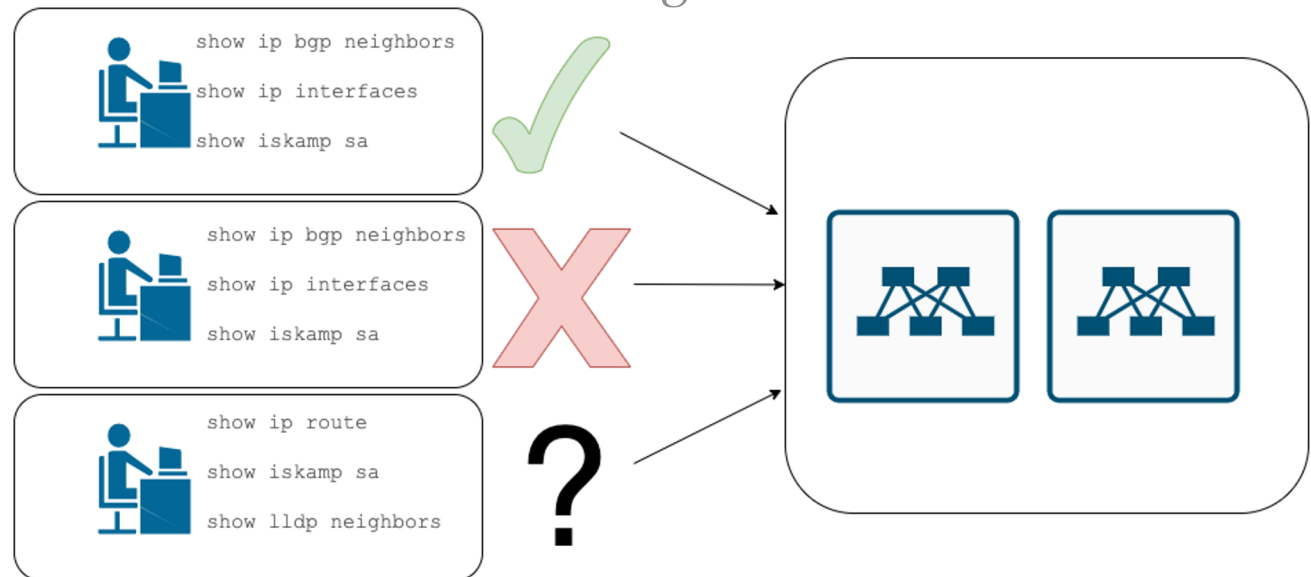


What are the Issues?

Verification

Time is spent on verifying the network is “healthy”

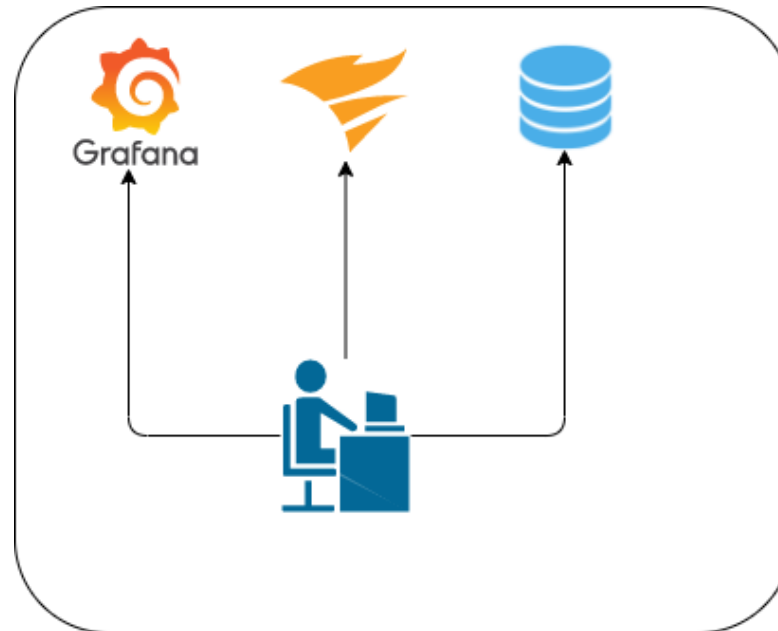
- This takes experience and institutional knowledge to know what that means



What are the Issues?

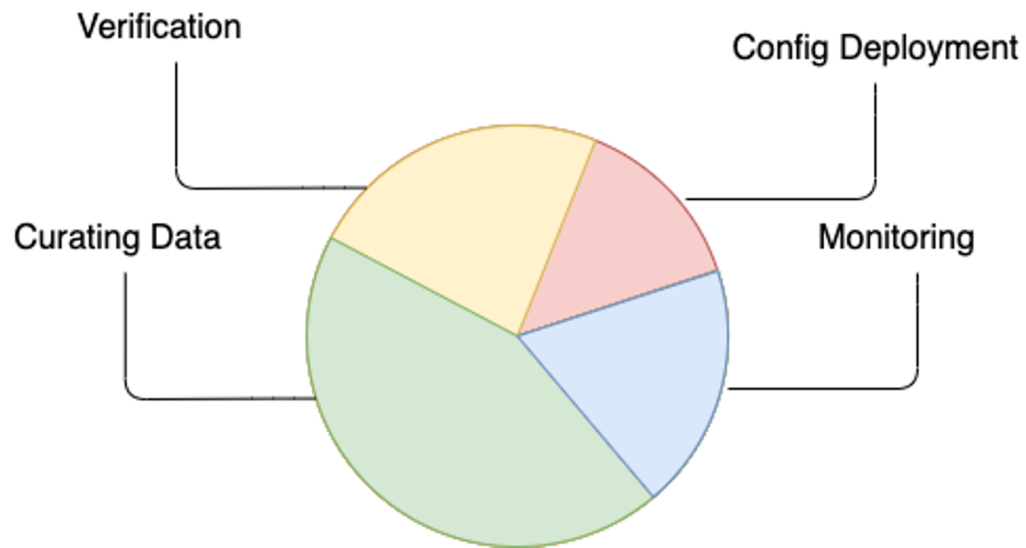
Network Management

Need to add to all other monitoring and inventory systems.



What are the Issues?

The actual configuration doesn't take long to deploy.



Workflow Analysis

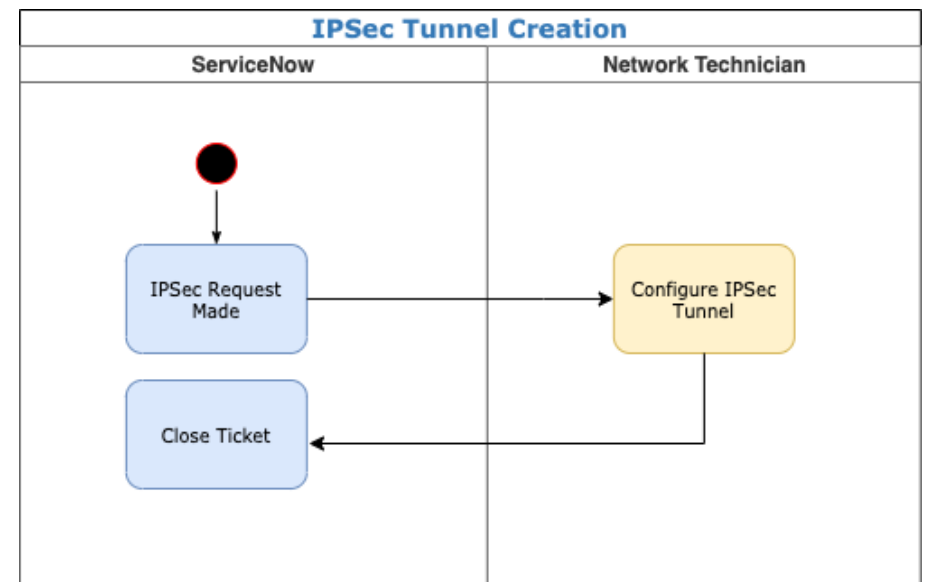
Why is it Important?

- Most networking groups are not actually aware of their own workflows.
- What should be tracked?
 - Number of times a type request happens
 - Amount of engineering time (hours worked)
 - Amount of time from request to completion
 - Opportunity Cost
- Should ask yourself, “how would I explain process to new engineer on the team”

First Take

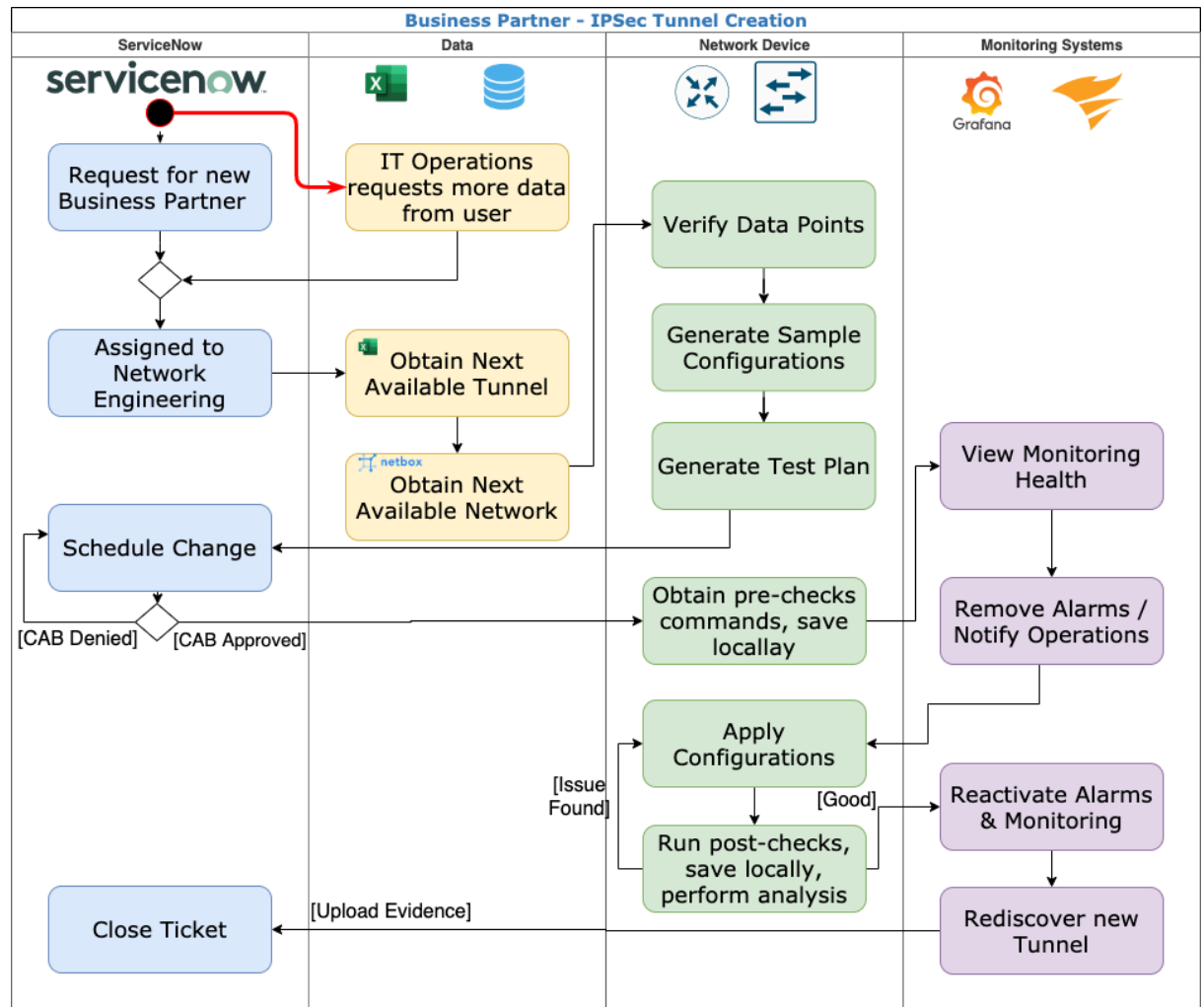
Tips:

- Do not discount the work being done.
- Take a system view of the workflow
- Consider all groups and approvals
- Consider **all tasks!**
- ... Don't do this ->



Second Take

Looking Better... still...

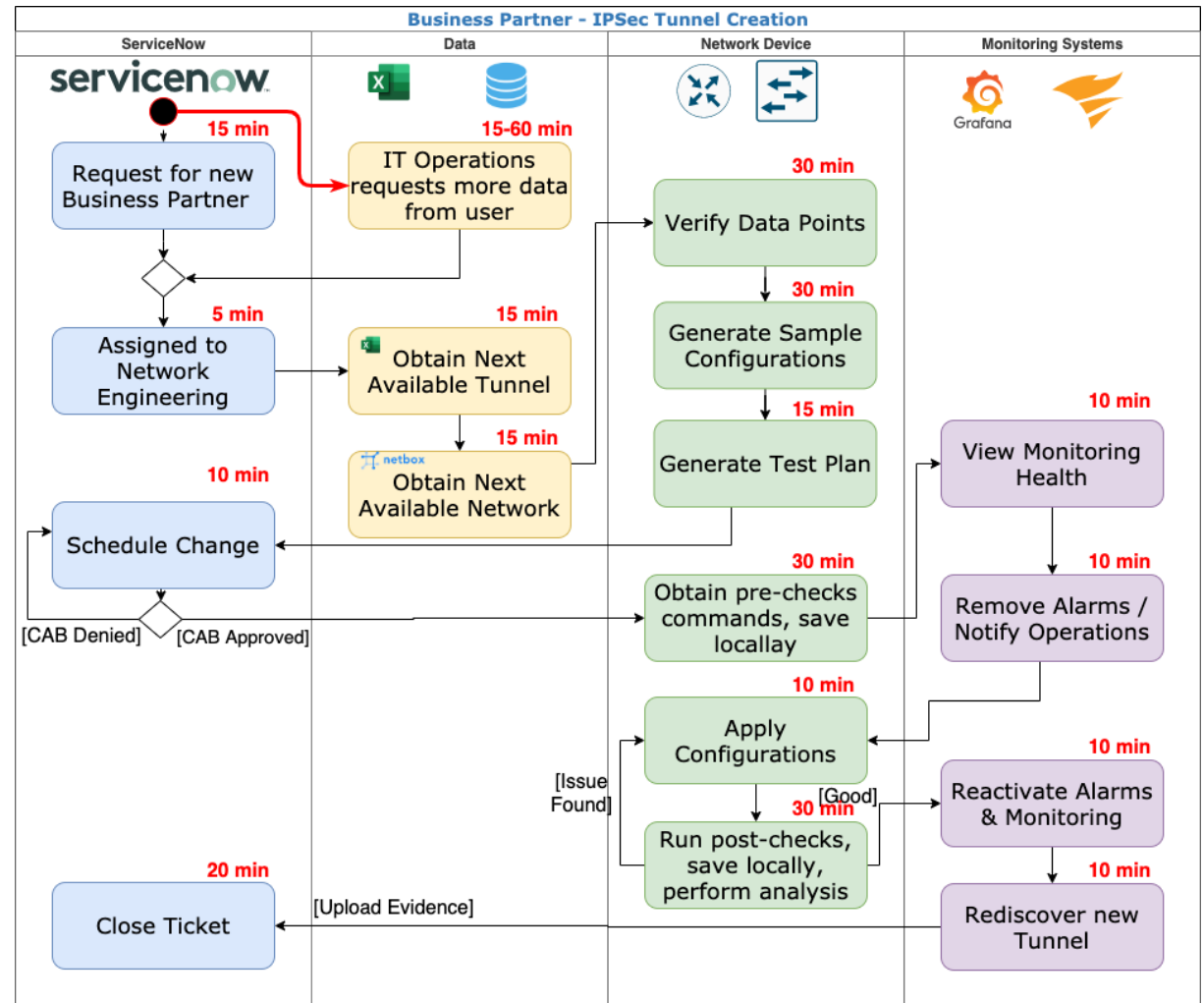


Time Required

Now we can see where the time is spent.

What does the data tell us?

Deploying configuration, has low ROI



Data Curation

Getting Better Data

- Limit free-form fields from requestors (work with ServiceNow developers as an example)
- Move tunnel assignments to programmatic accessible attribute
 - Database, NetBox, Git, gsheets, etc.
- Develop automation to update the Source of Truth for next available tunnels and subnets
- Develop automation to verify resources are free, by checking actual devices
- Develop automation to create configuration snippets and test plan

Generating Configuration

Populating a proper Source of Truth is the **cornerstone to automation**



```
interfaces:  
- name: Tunnel135  
  ip: 10.1.1.2/24  
  dst: 10.0.0.2
```

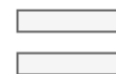
```
{% for interface in interfaces %}  
interface {{ interface['name'] }}  
  ip address {{ interface ['ip'] }}  
  tunnel source Ethernet0/1  
  tunnel destination {{ interface['dst'] }}  
  tunnel mode gre ip  
{% endfor %}
```

```
interface Tunnel135  
  ip address 10.1.1.2/24  
  tunnel source Ethernet0/1  
  tunnel destination 10.0.0.2  
  tunnel mode gre ip
```

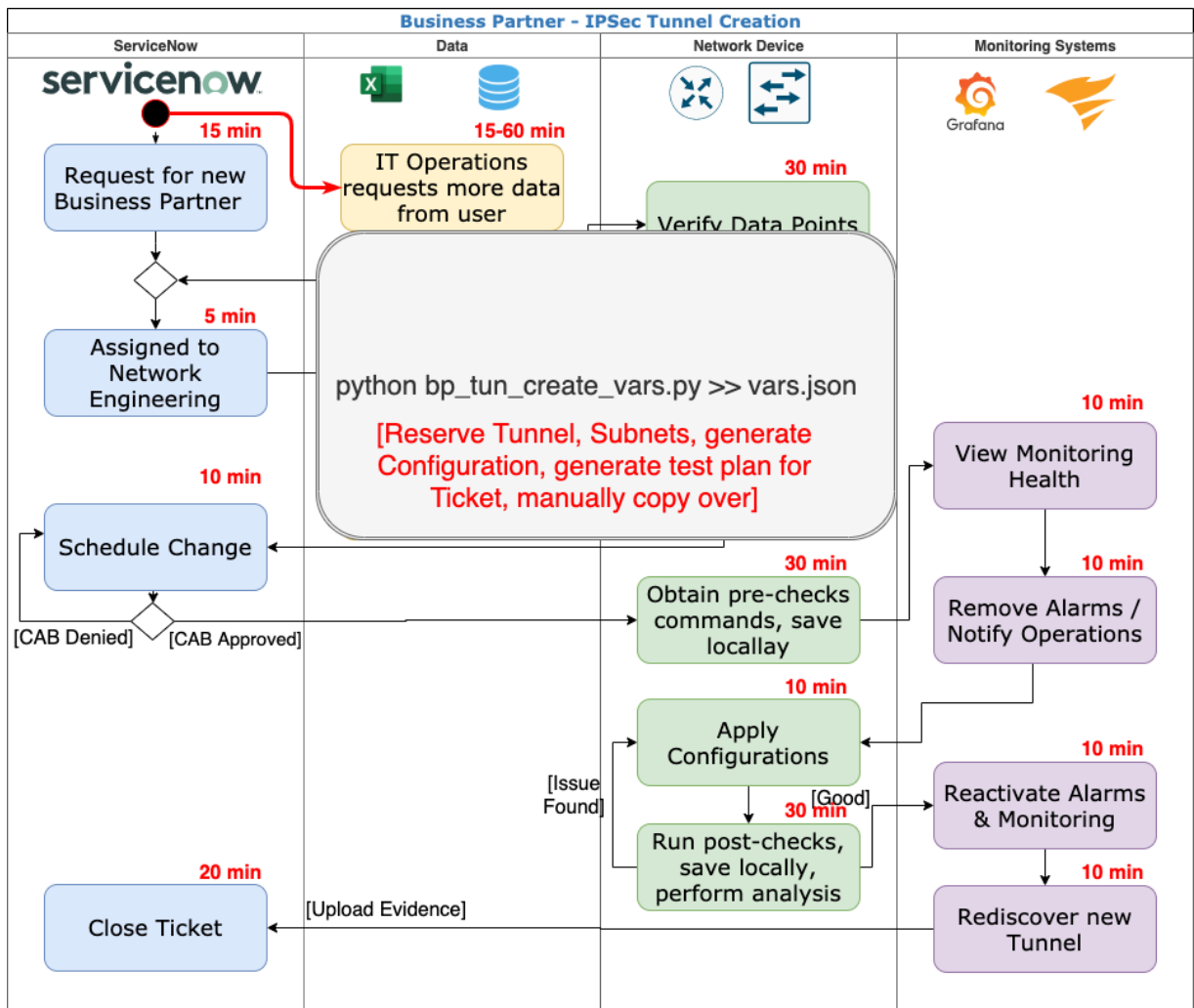
Data
Device Details



Templates
OS Implementation Details



Final Configuration



Verification

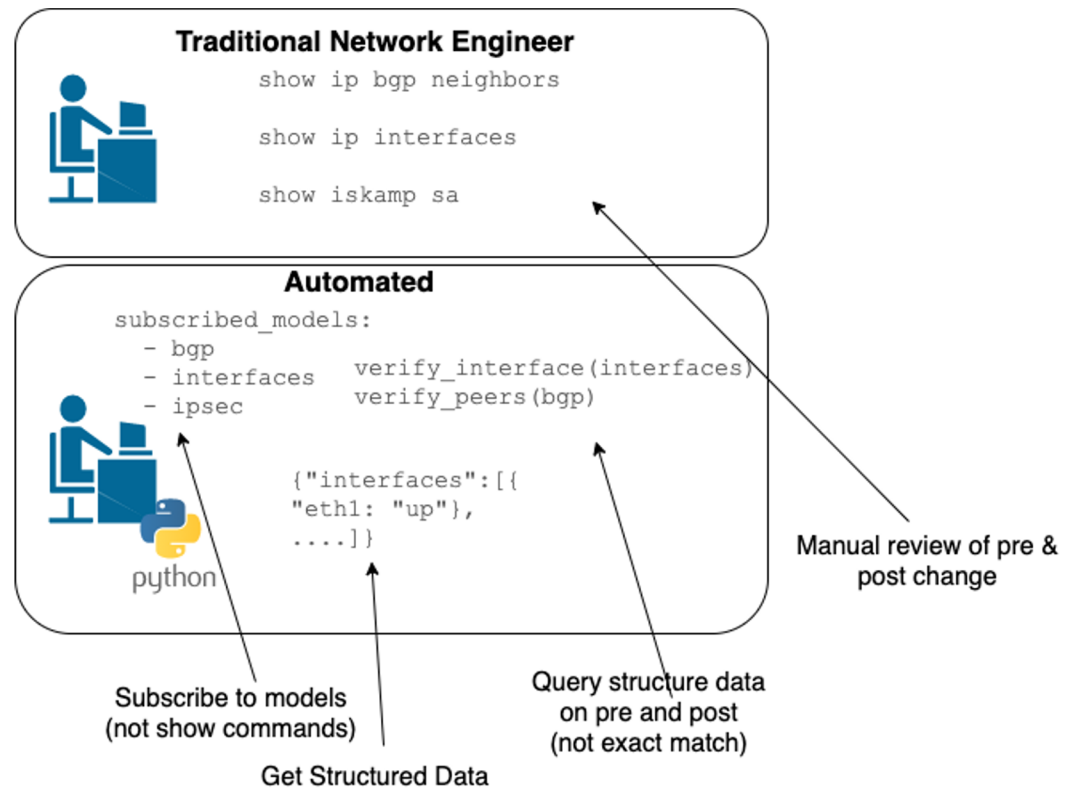
Pre & Post Checks

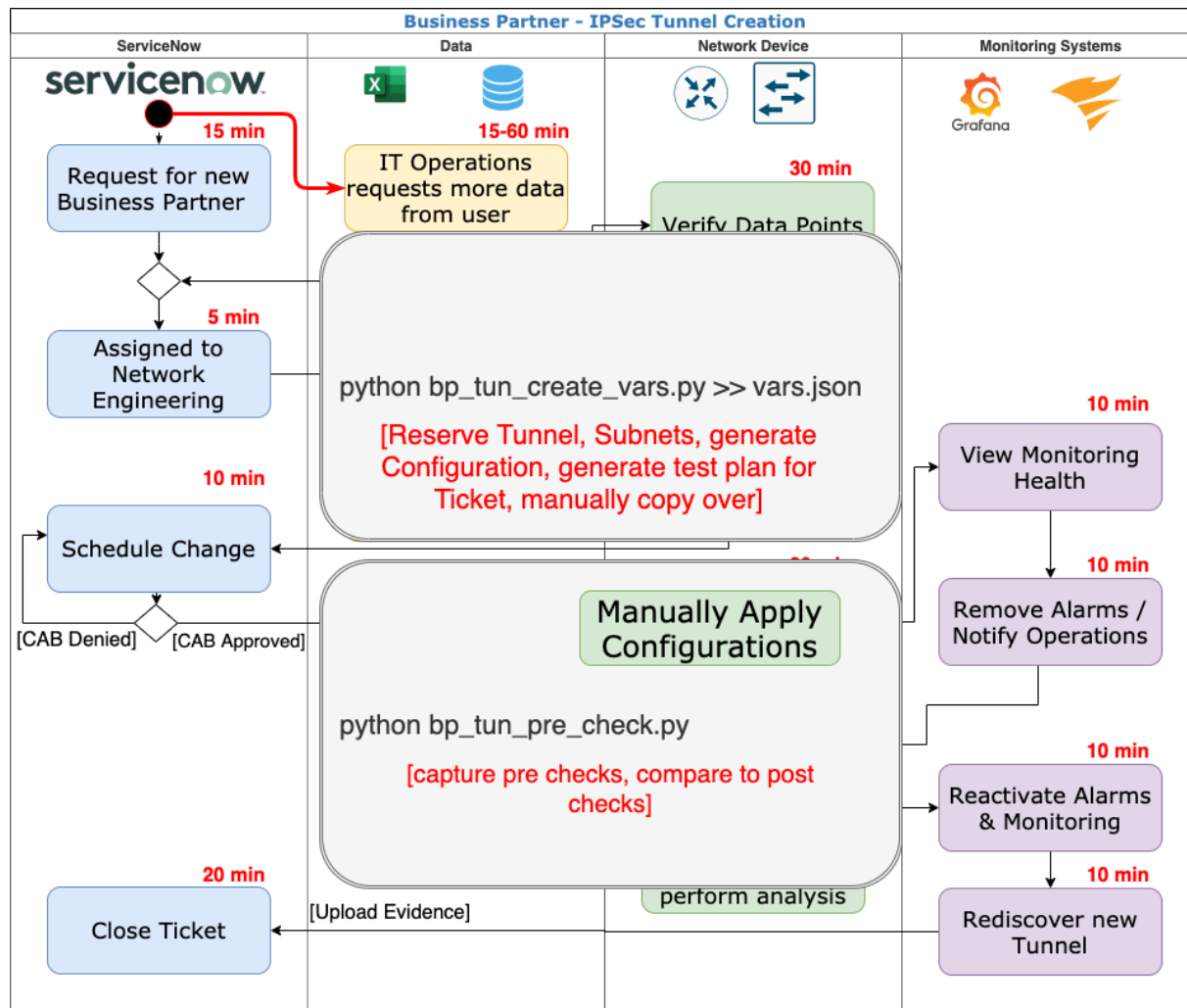
- Each engineer has their own tests
- There is no standardized definition of healthy
- There is no baseline for operational data (non-snmp, e.g. optic-levels)
- Data intends to change, 100 tunnels before change, 101 after
- Raw text is too large to compare
 - Timers and counters make it impossible to use diff
- There still needs to be evidence for change control

Pre & Post Checks

Rethink how checks are done

- Build queries against structured data
- Compare to “healthy” not just the change
 - Run all checks every time
- Remove manual diff review

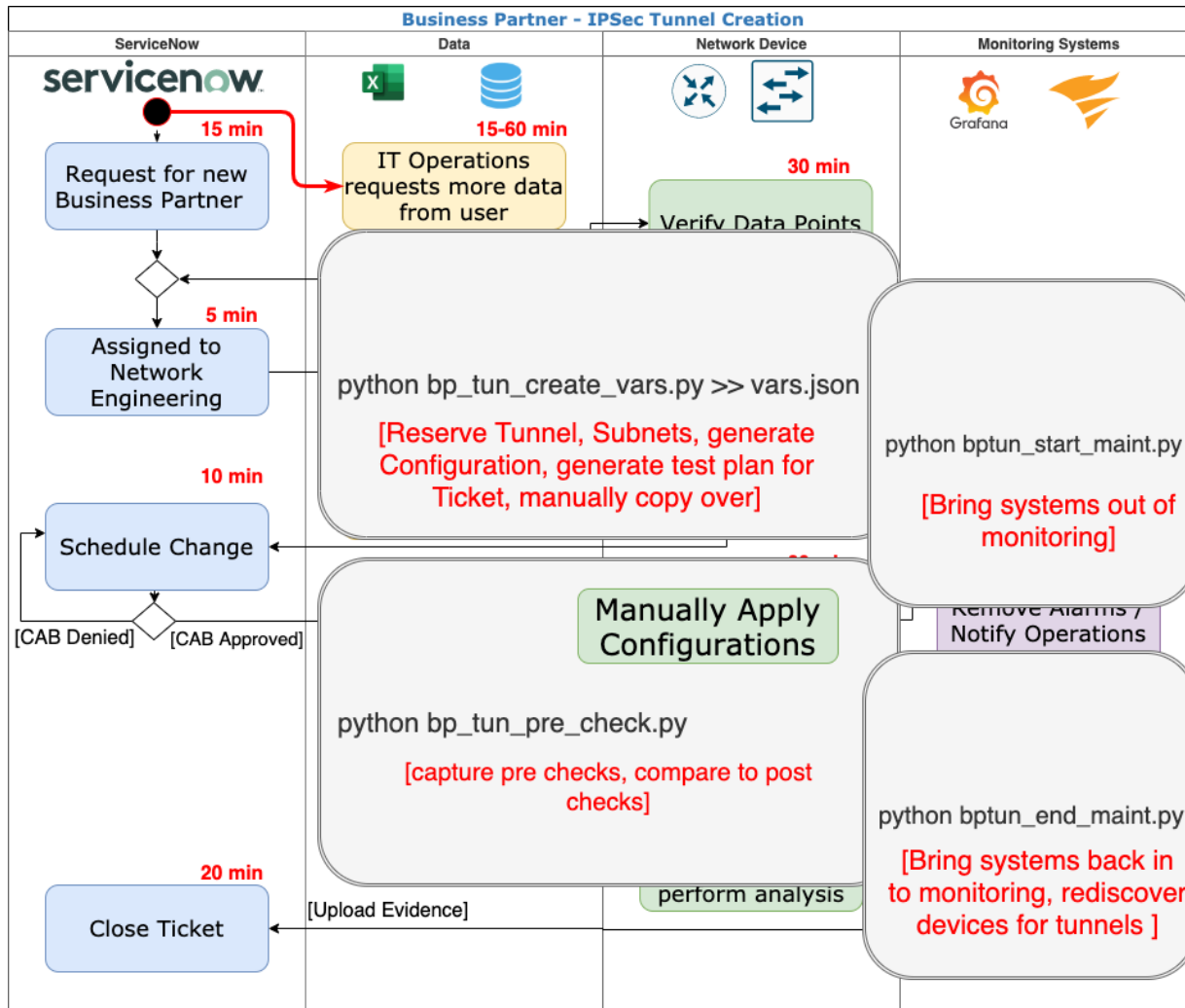




Network Management

Network Management

- Maintaining systems is difficult and tedious
- Large amount of false positives removes trust in monitoring
 - How many NOC's are filled with dozens of unanswered alarms?



Final Thoughts

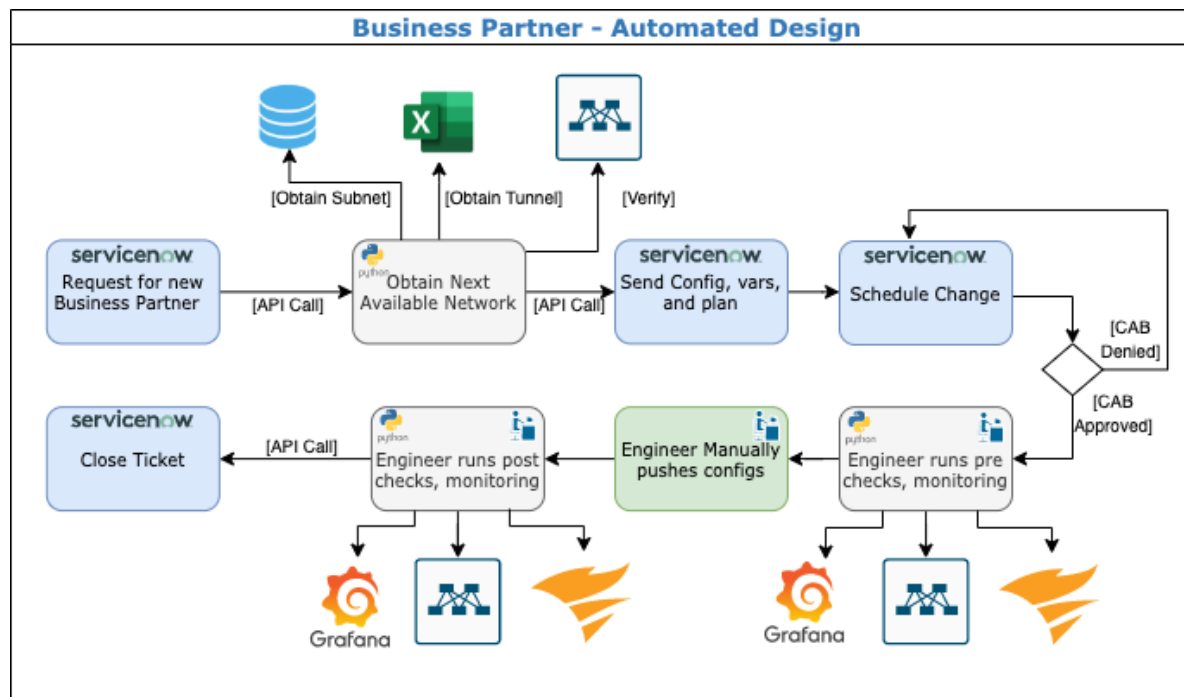
Automation without Config Deployment

What are the Benefits?

- Network Engineer is in control of what commands they send
 - This allows them to trust automation, engineers need to see the configs that are being sent
 - Automation **cannot be blamed** for issues
- Concentrate on tasks that take the most time
- Quicker to get into production and easier adoption
- Helps to build out a Source of Truth
- Allows automation to be introduced with less pressure

Final Design

Processed from 4-5 engineers hours to **less than 1!**



```
>>> network .toCode()
```

Thanks