SGNOG
Hackathon 2019

What is a Hackathon?

- Group of people (mostly strangers) collaborating and working intensive together to fix some problem(s), pain points.
    - New ideas, features
    - Manual & repetitive tasks
- Coding, scripting , programming, software development
- Creating solution (prototype) in the end

What actually happened?

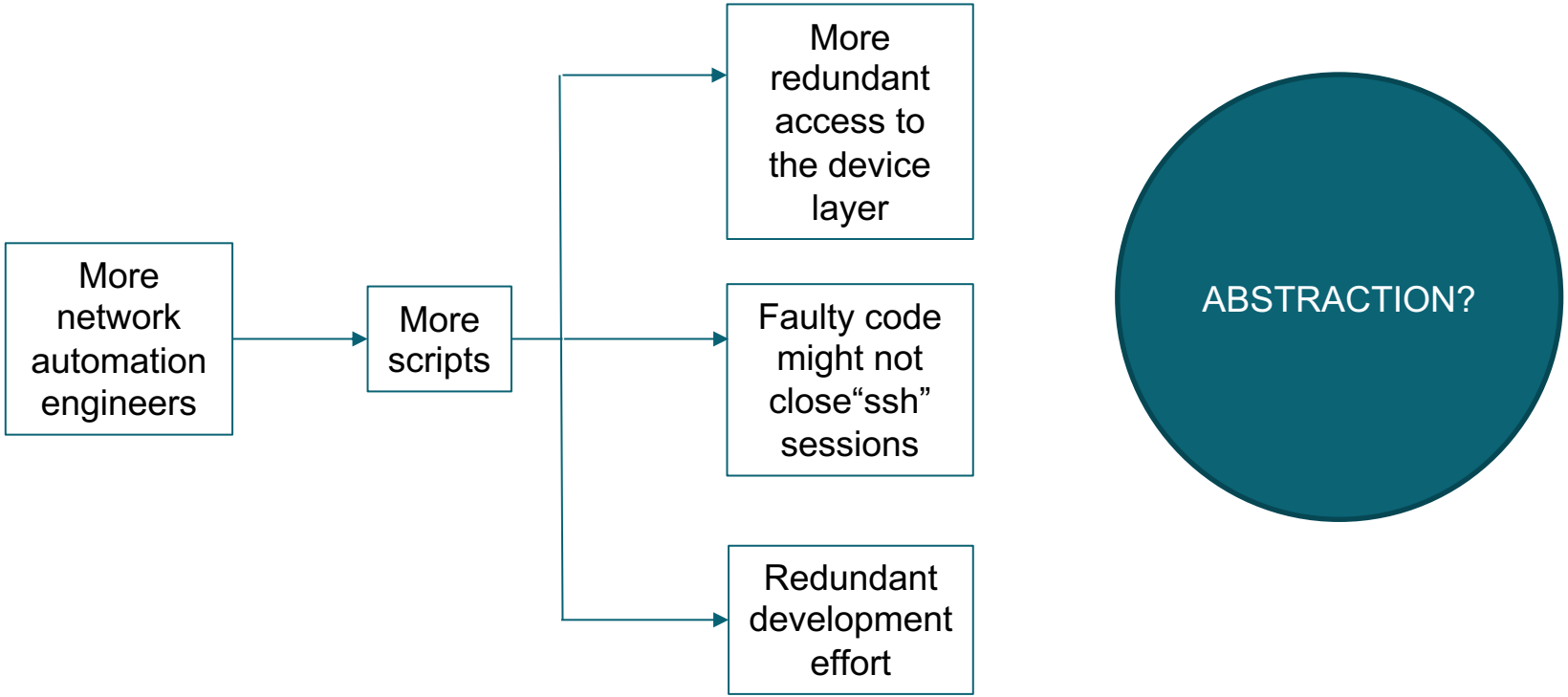
- Coding Workshop
- Lab work (setting up shell environment, created some virtual routers)
- Hack starts!
- while (hack)

```
{
        eat();
        code();
}
```

What they did ?

- Hear from them!
- 2 examples of the hack ideas to be shared

Problem Statement- StateDb



Solution: Network State DB

```
{
    "host1": {
        "bgp": {
            "enabled":bool
            "bgp": [peer1,peer2, peer3,...],
            peer1: {"local_asn": local_as, "remote_asn":neigh_as, "state": state_prfxrcd},
            peer2: {"local_asn": local_as, "remote_asn":neigh_as, "state": state_prfxrcd},
            peer_count: int
            prfx_count: int
            not_est_count: int
        },
        "interface": {
            interfaces: [interface1, interface2, ...]
            interface1 : { "desc": description, "status": link_status }
            interface2 ...
        },
        "lldp": {
            enabled: bool
            lldp: [local_if1, local_if2, ...]
            local_if1: {"nbr_name": neighbor, "nbr_if": neighbor_interface }
            local_if2:...
        },
        "ntp": {
            "enabled": bool
            ntp:[server_name1,server_name2,..]
            status: Bool
            reference: ref_server_name
        },

    }

}
```

StateDB

Flask
Netmiko
SNMP
Redis
REST

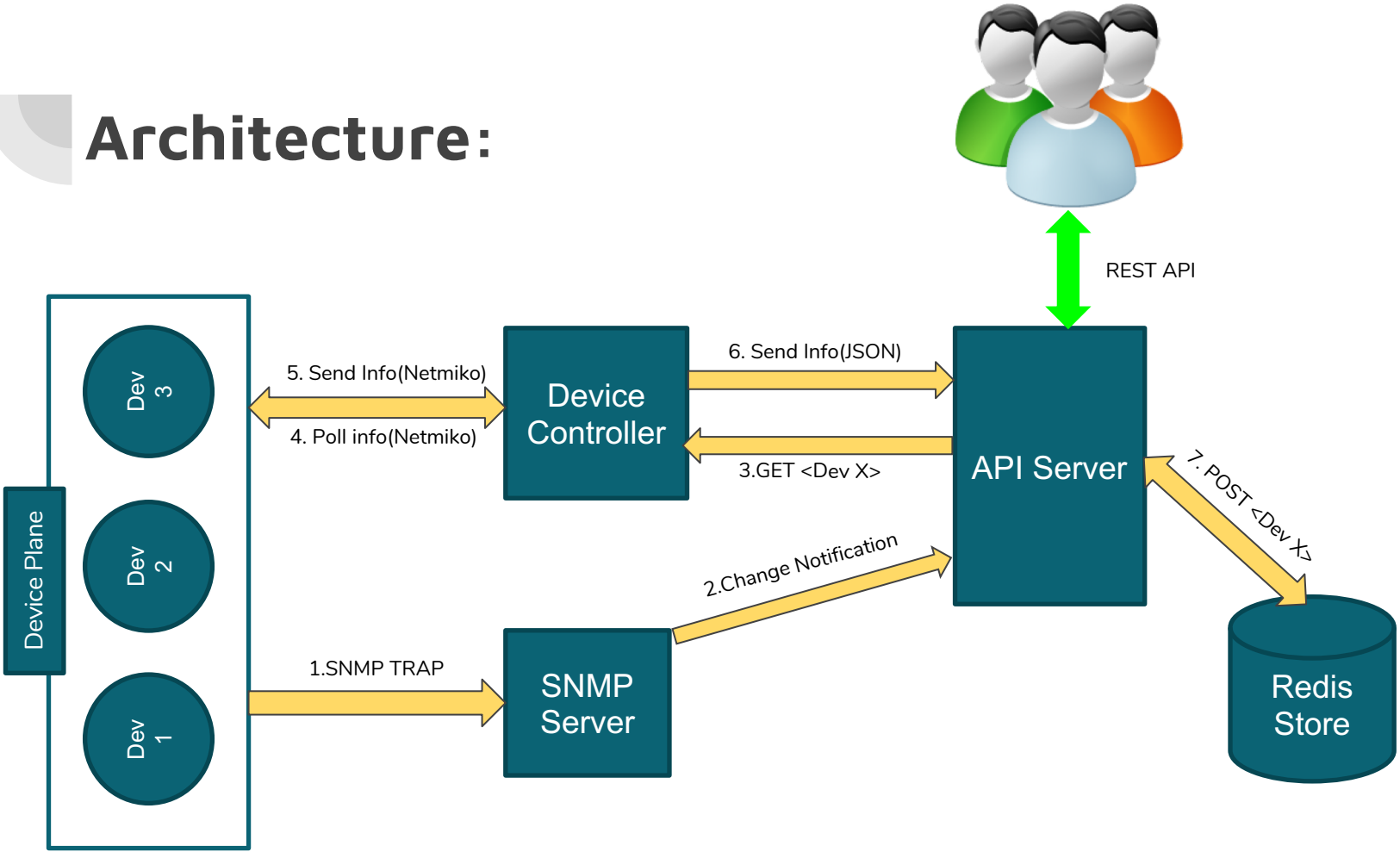
```
GET /v1/device/host1
```

Device Space

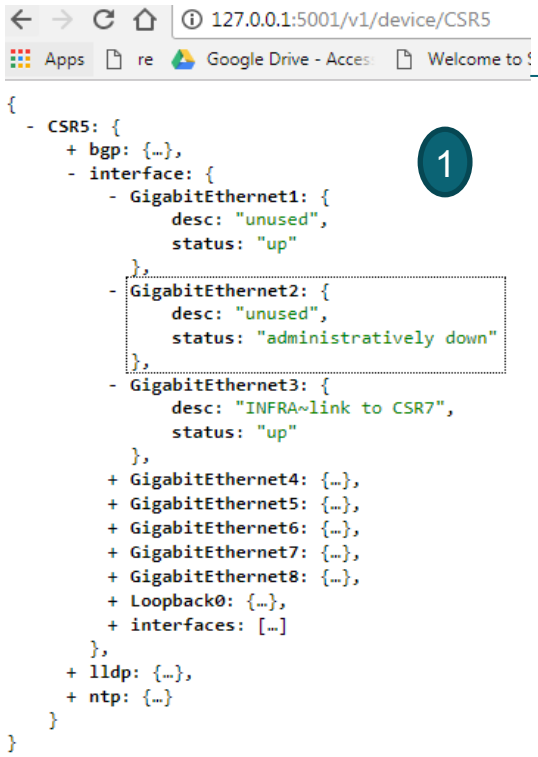
Code Space

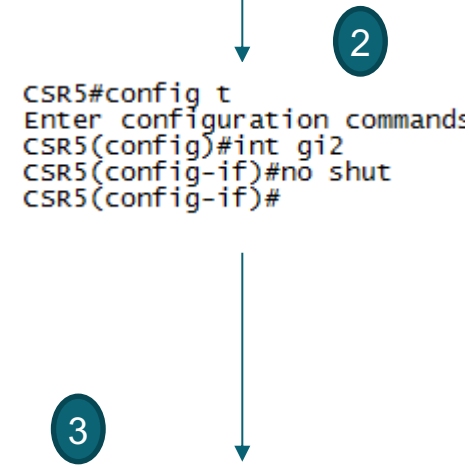
User Space

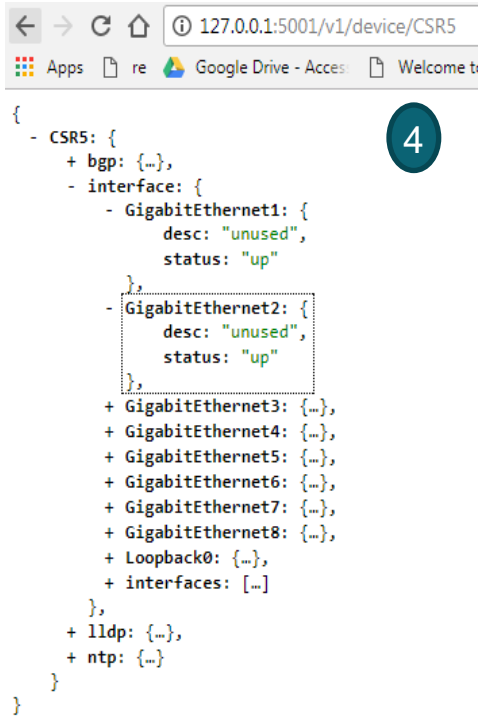
Architecture:



Demo:


127.0.0.1:5001/v1/device/CSR5
Apps
re
Google Drive - Acces
Welcome to
{
- CSR5: {
+ bgp: {…},
- interface: {
- GigabitEthernet1: {
desc: "unused",
status: "up"
},
- GigabitEthernet2: {
desc: "unused",
status: "administratively down"
},
- GigabitEthernet3: {
desc: "INFRA~link to CSR7",
status: "up"
},
+ GigabitEthernet4: {…},
+ GigabitEthernet5: {…},
+ GigabitEthernet6: {…},
+ GigabitEthernet7: {…},
+ GigabitEthernet8: {…},
+ Loopback0: {…},
+ interfaces: […]
},
+ lldp: {…},
+ ntp: {…}
}
}
1

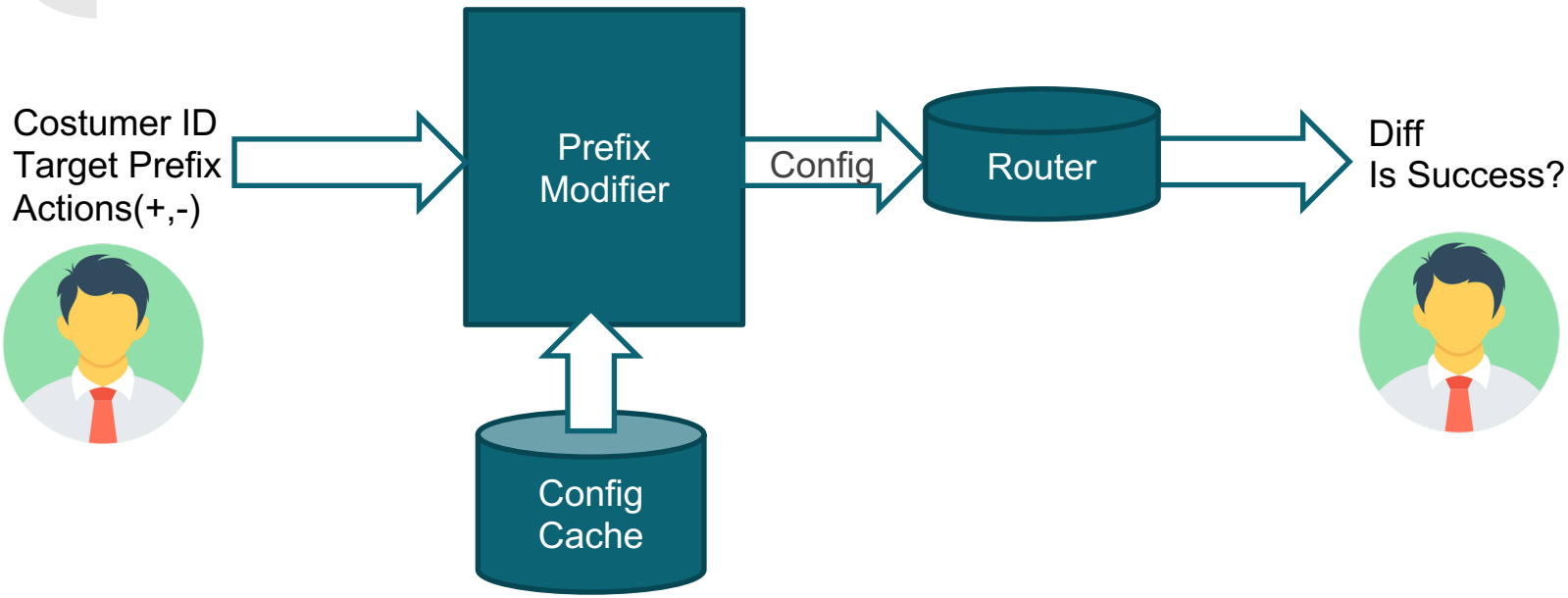

2
CSR5#config t
Enter configuration commands
CSR5(config)#int gi2
CSR5(config-if)#no shut
CSR5(config-if)#
3
INFO:werkzeug:172.16.14.3 - - [11/Jul/2019 08:47:38] "GET /favicon.ico HTTP/1.1" 404 -
INFO:__main__:Change Detected : Received statedb update request for : CSR5
INFO:__main__:Updating Statedb for CSR5
INFO:werkzeug:172.16.14.3 - - [11/Jul/2019 08:54:11] "POST /v1/device/update/ HTTP/1.1" 200 -


127.0.0.1:5001/v1/device/CSR5
Apps
re
Google Drive - Acces
Welcome t
{
- CSR5: {
+ bgp: {…},
- interface: {
- GigabitEthernet1: {
desc: "unused",
status: "up"
},
- GigabitEthernet2: {
desc: "unused",
status: "up"
},
+ GigabitEthernet3: {…},
+ GigabitEthernet4: {…},
+ GigabitEthernet5: {…},
+ GigabitEthernet6: {…},
+ GigabitEthernet7: {…},
+ GigabitEthernet8: {…},
+ Loopback0: {…},
+ interfaces: […]
},
+ lldp: {…},
+ ntp: {…}
}
}
4

Problem Statement- BGP Prefix-Set Modification via Python Script

- Customer's request to update their prefix-set is being done manually through these tedious, repetitive tasks:
  - Given the customer Service ID, look for the valid BGP neighbor
  - From the BGP configuration, look for applied route policy name
  - With the route policy name, derive the prefix-set name
  - Update the policy to add/remove the prefix
  - Validate the change
- Manual configuration of these BGP policies are more prone to human-error

Solution:



Packages Used: netmiko, ciscoconfparse, ipaddress

Source: https://github.com/shhackathon/sgnog19

Demo

```
RP/0/0/CPU0:xrv#show rpl prefix-set Service01_prefix
Thu Jul 11 05:05:41.585 UTC
prefix-set Service01_prefix
  1.1.1.4/22,
  1.1.1.2/24,
  1.1.1.49/24,
  1.1.1.50/24,
  1.1.1.3/22,
  104.2.0.0/16,
  102.3.0.0/16,
  11.1.10.3/22,
  192.168.0.0/24,
  10.0.0.0/24
end-set
!
RP/0/0/CPU0:xrv#
```

1. Prefix-set config before change

```
(sgnog-venv) daniel@ubuntu:~/sgnog19$ python main.py Service01 add 15.24.3.0/24
Service01
Service01 / Service01_prefix / prefix-set Service01_prefix
  1.1.1.4/22,
  1.1.1.2/24,
  1.1.1.49/24,
  1.1.1.50/24,
  1.1.1.3/22,
  104.2.0.0/16,
  102.3.0.0/16,
  11.1.10.3/22,
  192.168.0.0/24,
  10.0.0.0/24
 end set

edit prefix-set Service01_prefix inline add "15.24.3.0/24"

Thu Jul 11 05:00:41.405 UTC
[OK]
Proceed with commmit (yes/no)? [yes]:

Parsing.
226 bytes parsed in 1 sec (221)bytes/sec
Committing.
Prepared commit in 0 sec
.
1 items committed in 2 sec (0)items/sec
Updating.
Updated Commit database in 1 sec
DONE - Prefix 15.24.3.0/24 has been added to Service01_prefix successfully
Diff from Old to New set()
Diff from New to Old {'15.24.3.0/24'}
(sgnog-venv) daniel@ubuntu:~/sgnog19$
```

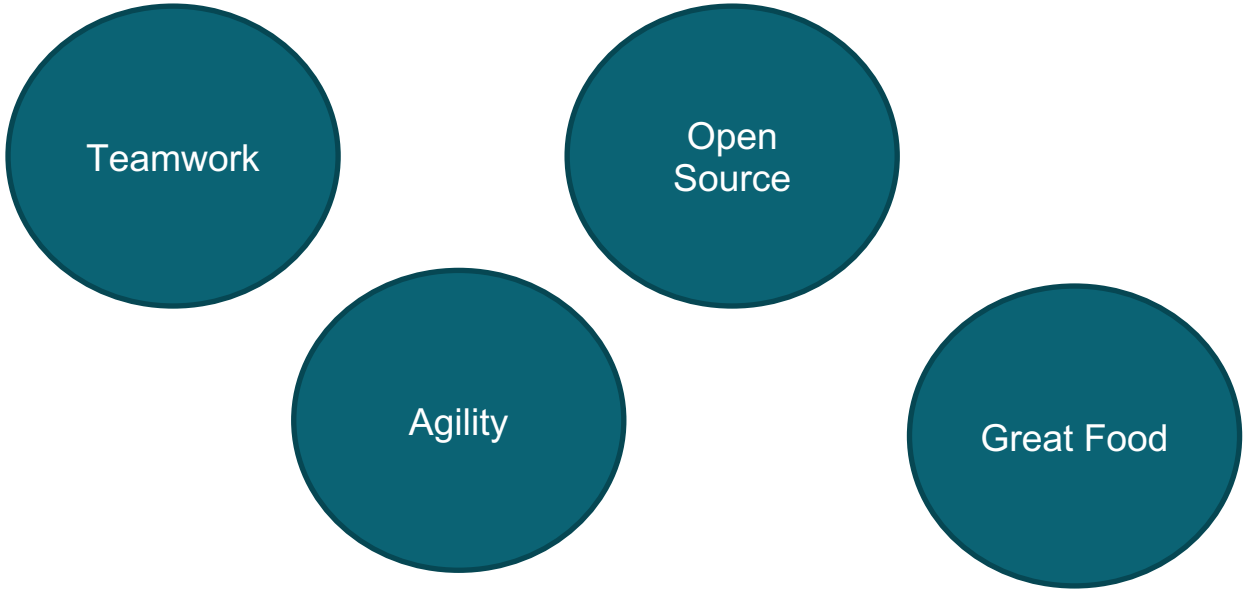
2. Script Output

```
DONE - Prefix 15.24.3.0/24 has been added to Service01_prefix successfully
Diff from Old to New set()
Diff from New to Old {'15.24.3.0/24'}
(sgnog-venv) daniel@ubuntu:~/sgnog19$
```

3.Prefix-set config after change:

```
RP/0/0/CPU0:xrv#show rpl prefix-set Service01_prefix
Thu Jul 11 05:09:25.889 UTC
prefix-set Service01_prefix
  1.1.1.4/22,
  1.1.1.2/24,
  1.1.1.49/24,
  1.1.1.50/24,
  1.1.1.3/22,
  104.2.0.0/16,
  102.3.0.0/16,
  11.1.10.3/22,
  192.168.0.0/24,
  10.0.0.0/24,
  15.24.3.0/24
end-set
!
RP/0/0/CPU0:xrv#
```

Key Takeaways










mini strawberry
cheesecake
- vegetarian

Questions, Suggestions?

#Fin.